



IN PARTNERSHIP WITH:
CNRS

**Institut polytechnique de
Grenoble**

**Université Joseph Fourier
(Grenoble)**

Activity Report 2012

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

| | |
|---|-----------|
| 1. Members | 1 |
| 2. Overall Objectives | 1 |
| 3. Scientific Foundations | 2 |
| 3.1. Components and semantics | 2 |
| 3.2. Open programming | 3 |
| 3.3. Software infrastructure | 3 |
| 3.4. System management and control | 3 |
| 4. Software | 4 |
| 4.1. AAC_tactics | 4 |
| 4.2. ATBR | 4 |
| 4.3. MoKa | 5 |
| 4.4. ConSer | 5 |
| 4.5. e-Caching | 6 |
| 4.6. MRB | 6 |
| 4.7. BZR | 6 |
| 5. New Results | 7 |
| 5.1. Languages and Foundations: Process algebra | 7 |
| 5.2. Control for adaptive systems: Discrete control for adaptive and reconfigurable systems | 7 |
| 5.3. System support: System support for multicore machines | 8 |
| 5.4. System support: Performance and dependability benchmarking | 8 |
| 6. Bilateral Contracts and Grants with Industry | 9 |
| 7. Partnerships and Cooperations | 9 |
| 7.1. National Initiatives | 9 |
| 7.1.1. Automatique pour l'informatique autonome (CNRS PEPS) | 9 |
| 7.1.2. SocEDA (ANR Arpege project) | 9 |
| 7.1.3. PiCoq (ANR project) | 9 |
| 7.1.4. Project MyCloud (ANR project) | 10 |
| 7.1.5. Famous (ANR project) | 10 |
| 7.1.6. REVER (ANR project) | 11 |
| 7.1.7. CtrlGreen (ANR project) | 11 |
| 7.2. European Initiatives | 11 |
| 8. Dissemination | 12 |
| 8.1. Scientific Animation | 12 |
| 8.2. Teaching - Supervision - Juries | 12 |
| 9. Bibliography | 13 |

Project-Team SARDES

Keywords: Distributed System, Cloud Computing, Fault Tolerance, Formal Methods, Component Programming

Creation of the Project-Team: January 01, 2002 .

1. Members

Research Scientists

Jean-Bernard Stefani [Team Leader, Senior Researcher, Ingénieur Général des Mines]
Eric Rutten [Inria Junior Researcher, HdR]
Damien Pous [CNRS Junior Researcher]
Vivien Quéma [Professor, INPG, HdR]

Faculty Members

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]

PhD Students

Xin An [Inria, Famous contract]
Pierre-Louis Aublin [government grant]
Gautier Berthou [Inria, PLAY project]
Xavier Etchevers [Orange Labs]
Baptiste Lepers [CNRS, SocEDA project]
Quentin Sabah [ST Microelectronics]
Amit Sangroya [Inria, MyCloud project]
Soguy Mak-Kare Gueye [Inria, CtrlGreen project]

Post-Doctoral Fellows

Damian Serrano-Garcia [Inria, MyCloud project]
Barbara Petit [Inria, REVER project]

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 (multi-core 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** Work 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 [92], and software architecture [90] 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 [41]. Several component models and associated architecture description languages have been devised over the past fifteen years: see e.g. [71] for an analysis of recent component models, and [75], [47] for surveys of architecture description languages.

To natively support configurability and adaptability in systems, SARDES component technology also draws from ideas in reflective languages [66], and reflective middleware [69], [45], [52]. Reflection can be used both to increase the separation of concerns in a system architecture, as pioneered by aspect-oriented programming [67], 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 [72], [36], 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 [78], or the analysis of object-orientation [37]. Relevant developments for SARDES endeavours include behavioral theory and coinductive proof techniques [87], [85], process calculi with localities [48], [50], [53], and higher-order variants of the π -calculus [86], [60].

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 [83] — 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 [79], [94], 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 [88], Alice [83], [84], ArchJava [38], Classages [73], Erlang [40], Oz [94], and Scala [80].
- For their security and failure management features: Acute, E [77], Erlang and Oz [51].
- For their support for concurrent and distributed execution, Acute, Alice, JoCaml [56], E, Erlang, Klaim [44], 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 [91] has a long history, with systems such as Oberon [95], SPIN [43] or JX [57]. More recent and relevant developments for SARDES are:

- The developments around the Singularity project at Microsoft Research [55], [63], which illustrates the use of language-based software isolation for building a secure operating system kernel.
- The seL4 project [58], [68], 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 [46] and Barrelfish [42].
- The development of efficient run-time for event-based programming on multicore systems such as libasync [96], [70].

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 [81] 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 [64]. Several research projects [35] are active in this area. [65], [62] 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 [54], and techniques for constructing control algorithms in the decision stage of administration feedback loops, including discrete control techniques [49], and continuous ones [59].

Management and control functions built by SARDES require also the development of distributed algorithms [74], [93] at different scales, from algorithms for multiprocessor architectures [61] to algorithms for cloud computing [76] and for dynamic peer-to-peer computing systems [39], [82]. Of particular relevance in the latter contexts are epidemic protocols such as gossip protocols [89] 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. MOKA is 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 admission 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 $AM-C$, $PM-C$, $AA-PM-C$ and $PA-AM-C$. 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: Damian Serrano-Garcia, 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-Garcia, 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://bzx.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, Jean-Bernard Stefani, Barbara Petit.

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 and Rever projects.

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 primitives for controlling reversibility in a higher-order variant of the π -calculus. We have shown that the combination of a basic notion of message alternative coupled with a rollback primitive that respects causal consistency provides enough expressive power to encode various rollback policies. We have also started to study the expressive power of these primitives with respect to transactional constructs. In particular, we have shown that our primitives allow for a faithful encoding of a notion of communicating transaction proposed by Hennessy et al, while avoiding spurious rollbacks which mar Hennessy's approach. This work has been submitted for publication. A digest of our main ideas on controlling reversibility has appeared in [25].

We have also started a study on the cost of making a concurrent programming language reversible. More specifically, we have started from an abstract machine for a fragment of the Oz programming language and made it reversible. We have shown that the overhead of the reversible machine with respect to the original one in terms of space is at most linear in the number of execution steps, and that this bound is tight since some programs cannot be made reversible without storing a commensurate amount of information. This work has been published in [26].

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

Participants: Eric Rutten, Noël De Palma, Olivier Gruber, Fabienne Boyer, Xin An, Soguy Mak-Kare Gueye.

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. 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.

This year, we have started investigating the application of discrete controller synthesis to various problems in computer systems management and administration. The increasing complexity of computer systems has led to the automation of administration functions, in the form of autonomic managers. One important aspect requiring such management is the issue of energy consumption of computing systems, in the perspective of green computing. As these managers address each a specific aspect, there is a need for using several managers to cover all the domains of administration. However, coordinating them is necessary for proper and effective global administration. Such coordination is a problem of synchronization and logical control of administration operations that can be applied by autonomous managers on the managed system at a given time in response to events observed on the state of this system. We therefore propose to investigate the use of reactive models with events and states, and discrete control techniques to solve this problem. In [20], [21], [31], [30], we illustrate this approach by integrating a controller obtained by synchronous programming, based on Discrete Controller Synthesis, in an autonomic system administration infrastructure. The role of this controller is to orchestrate the execution of reconfiguration operations of all administration policies to satisfy properties of logical consistency. We have applied this approach to coordinate energy-aware managers for self-optimization, self-regulation of processor frequency and self-repair.

5.3. System support: System support for multicore machines

Participants: Vivien Quéma, Renaud Lachaize, Baptiste Lepers.

Multicore machines with Non-Uniform Memory Accesses (NUMA) are becoming commodity platforms. Efficiently exploiting their resources remains an open research problem. This line of work investigates system support to tackle various issues related to efficient resource management and programming support.

One of the key concerns in efficiently exploiting multicore NUMA architectures is to limit as much as possible the number of remote memory accesses (i.e., main memory accesses performed from a core to a memory bank that is not directly attached to it). However, in many cases, existing profilers do not provide enough information to help programmers achieve this goal. We have developed MemProf [24], the first profiler that allows programmers to choose and implement efficient application-level optimizations for NUMA systems. MemProf achieves this goal by allowing programmers to (i) precisely understand which memory objects are accessed remotely in memory, and (ii) building temporal flows of interactions between threads and objects. We evaluated MemProf using four applications (FaceRec, Streamcluster, Psearchy, and Apache) on three different machines. In each case, we showed how MemProf helped us choose and implement efficient optimizations, unlike existing profilers. These optimizations provide significant performance gains on the studied applications (up to 161%), while requiring very lightweight modifications (10 lines of code or less).

State-machine replication is a well-known fault-tolerance technique. Unfortunately existing state-machine replication schemes do not scale well on multicore machines. In collaboration with U. Texas at Austin (L. Alvisi), we have developed a new state-machine replication scheme [23], that departs from the standard agree-execute architecture of existing schemes, in favor of a more optimistic, and less deterministic, execute-verify replication scheme, which yields much better scalability. We have evaluated Eve's throughput gain compared with traditional sequential execution approaches, as well as Eve's overheads compared to unreplicated multithreaded execution and to alternative replication approaches.

5.4. System support: Performance and dependability benchmarking

Participants: Amit Sangroya, Damian Serrano-Garcia, Sara Bouchenak [correspondant].

MapReduce is a popular programming model for distributed data processing. Extensive research has been conducted on the reliability of MapReduce, ranging from adaptive and on-demand fault-tolerance to new fault-tolerance models. However, realistic benchmarks are still missing to analyze and compare the effectiveness of these proposals. To date, most MapReduce fault-tolerance solutions have been evaluated using micro benchmarks in an ad-hoc and overly simplified setting, which may not be representative of real-world applications. To remedy this situation, we have developed MRBS, a comprehensive benchmark suite for evaluating the dependability of MapReduce systems. MRBS 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. online interactive applications. MRBS allows to inject various types of faults at different rates. It also considers different application workloads and data loads, and produces extensive reliability, availability and performance statistics. We have shown the use of MRBS with Hadoop clusters running on Amazon EC2, and on a private cloud [29], [28].

6. Bilateral Contracts and Grants with Industry

6.1. Bilateral Grants with Industry

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

7. Partnerships and Cooperations

7.1. National Initiatives

7.1.1. *Automatique pour l'informatique autonome (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/>

7.1.2. *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 eventdriven 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.

7.1.3. *PiCoq (ANR project)*

Participants: Damien Pous, Jean-Bernard Stefani.

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 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/>.

7.1.4. Project MyCloud (ANR project)

Participants: Amit Sangroya, Sara Bouchenak, Damian Serrano-Garcia.

The objective of the MyCloud project is to define and implement a novel cloud model: *SLAaaS* (*SLA aware Service*). 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 transparency 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.

7.1.5. 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.

7.1.6. *REVER (ANR project)*

Participants: Barbara Petit, Jean-Bernard Stefani.

The REVER project aims to develop semantically well-founded and composable abstractions for dependable distributed computing on the basis of a reversible programming model, where reversibility means the ability to undo any program execution and to revert it to a state consistent with the past execution. The critical assumption behind REVER is that by combining reversibility with notions of compensation and modularity, one can develop systematic and composable abstractions for dependable programming.

The REVER workprogramme is articulated around three major objectives:

- To investigate the semantics of reversible concurrent processes.
- To study the combination of reversibility with notions of compensation, isolation and modularity in a concurrent and distributed setting.
- To investigate how to support these features in a practical (typically, object-oriented and functional) programming language design.

The project partners are Inria (Sardes in Grenoble and Focus in Bologna), Université de Paris VII (PPS laboratory), and CEA (List laboratory).

The project runs from December 2011 to November 2015.

7.1.7. *CtrlGreen (ANR project)*

Participants: Fabienne Boyer, Noël De Palma, Eric Rutten, Soguy Mak-Kare Gueye.

The goal of the CtrlGreen project is to develop the control techniques and software infrastructure required to build energy-efficient data centers. Because resource management must meet performance, dependability and scalability objectives and as well as service level agreements, energy-efficiency must be considered as a multi-criteria control problem. CtrlGreen aims to develop an autonomic system approach, where multiple control loops may coexist and coordinate. Specifically, the work will proceed along four directions:

- The study of reactive control techniques including synchronous languages and discrete controller synthesis to program, verify and synthesize coordinating controllers.
- The development of a controllable platform that can provide system level support for the deployment and integration of the required controllers.
- The study of several green data center scenarios that involve the coordination between several controllers at different levels (hardware, operating system and middleware) and targetting different objectives (performance, availability, energy efficiency, etc).
- Experiments with an industrial data center to evaluate CtrlGreen techniques in a real world environment, with multiple running applications.

The project partners include Eolas, Inria Rennes, INPT/IRIT Toulouse, LIG (Sardes) and ScalAgent. The project runs from January 2012 to December 2014.

7.2. European Initiatives

7.2.1. *FP7 Projects*

7.2.1.1. *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: EBM WebSourcing (Fr), Inria (OASIS and SARDES) (Fr), France Telecom (Fr), ICCS (Gr), Ecole des Mines Albi (Fr), CIM (Serbia).

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 should enable the 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 will comprise in particular:

- A 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.
- A 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.

8. Dissemination

8.1. Scientific Animation

S. Bouchenak: Co-General Chair of the CFSE/Renpar/SympA conferences in 2013.

F. Boyer: Member of the Programme Committee of ICAS 2012.

N. De Palma: Chair of the GCM (Green and Cloud Management) 2012 workshop.

Vivien Quema: Organizer of the Winter School on Hot Topics in Distributed Systems (each year since 2008).

E. Rutten: Member of the Programme Committee of APRES (Adaptive and Reconfigurable Embedded Systems workshop) 2012. Member of the IFAC Technical Committee 1.3 on Discrete Event and Hybrid Systems.

J.B. Stefani: Chairman of IFIP WG6.1 on Architectures and Protocols for Distributed Systems. Chairman of the Steering Committee of the FMOODS/FORTE conference series. Member of the Programme Committee of ACM SAC 2012. Member of the editorial board of the journal Annals of Telecommunications. Member of the Technology Council of the company STMicroelectronics.

8.2. Teaching - Supervision - Juries

8.2.1. Teaching

V. Quéma is a full-time professor at Grenoble INP.

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 assistant-professor at Grenoble University.

Sara Bouchenak is a full-time assistant-professor at Grenoble University.

Fabienne Boyer is a full-time assistant-professor at Grenoble University.

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

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", D. SANGIORGI, J. RUTTEN (editors), CambridgeTracts in Theoretical Computer Science, Cambridge University Press, 2012, chap. 6, n^o 52.

- [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] T. BRAIBANT. *Algèbres de Kleene, réécriture modulo AC et circuits en coq*, Université de Grenoble, February 2012, <http://hal.inria.fr/tel-00683661>.
- [14] C. A. MEZZINA. *Réversibilité dans le pi calcul d'ordre supérieur*, Université de Grenoble and Università di Bologna, February 2012, <http://hal.inria.fr/tel-00683964>.

Articles in International Peer-Reviewed Journals

- [15] T. BRAIBANT, D. POUS. *Deciding Kleene Algebras in Coq*, in "Logical Methods in Computer Science", 2012, vol. 8, n^o 1.
- [16] N. DE PALMA, D. HAGIMONT, F. BOYER, L. BROTO. *Self-Protection in a Clustered Distributed System*, in "IEEE Transactions on Parallel Distributed Systems", 2012, vol. 23, n^o 2.
- [17] D. POUS. *Untyping Typed Algebras and Colouring Cyclic Linear Logic*, in "Logical Methods in Computer Science", 2012, vol. 8, n^o 2.
- [18] L. SEINTURIER, P. MERLE, R. ROUYOY, D. ROMERO, V. SCHIAVONI, J.-B. STEFANI. *A Component-Based Middleware Platform for Reconfigurable Service-Oriented Architectures*, in "Software: Practice and Experience", May 2012, vol. 42, n^o 5, p. 559-583 [DOI : 10.1002/SPE.1077], <http://hal.inria.fr/inria-00567442>.

International Conferences with Proceedings

- [19] X. AN, A. GAMATIÉ, E. RUTTEN. *CLASSY: a Clock Analysis System for Rapid Prototyping of Embedded Applications on MPSoCs*, in "15th International Workshop on Software and Compilers for Embedded Systems Map2MPSoC/SCOPEs 2012", 2012.
- [20] F. BOYER, G. DELAVAL, N. DE PALMA, O. GRUBER, E. RUTTEN. *Discrete supervisory control application to computing systems administration*, in "14th IFAC Symposium on Information Control Problems in Manufacturing, INCOM'12", 2012.
- [21] S. M.-K. GUEYE, N. DE PALMA, E. RUTTEN. *Coordinating energy-aware administration loops using discrete control*, in "8th Int. Conf. on Autonomic and Autonomous Systems (ICAS)", 2012.
- [22] S. GUILLET, F. DE LAMOTTE, N. LE GRIGUER, E. RUTTEN, G. GOGNIAT, J.-P. DIGUET. *Designing formal reconfiguration control using UML/MARTE*, in "7th International Workshop on Reconfigurable and Communication Centric Systems-on-Chip, ReCoSoC 2012", 2012.
- [23] M. KAPTRITSOS, Y. WANG, V. QUÉMA, A. CLEMENT, L. ALVISI, M. DAHLIN. *Eve: Execute-Verify Replication for Multi-Core Servers*, in "10th USENIX Symposium on Operating Systems Design and Implementation (OSDI)", 2012.

- [24] R. LACHAIZE, B. LEPEERS, V. QUÉMA. *MemProf: a Memory Profiler for NUMA Multicore Systems*, in "USENIX Annual Technical Conference 2012 (USENIX '12)", USENIX, 2012.
- [25] I. LANESE, C. A. MEZZINA, J.-B. STEFANI. *Controlled Reversibility and Compensations*, in "4th Int. Workshop on Reversible Computation (RC 2012)", Lecture Notes in Computer Science, Springer, 2012, vol. 7581.
- [26] M. LIENHARDT, I. LANESE, C. A. MEZZINA, J.-B. STEFANI. *A Reversible Abstract Machine and Its Space Overhead*, in "14th IFIP Int. Conf. FMOODS and 32nd IFIP Int. Conf. FORTE (FMOODS/FORTE)", Lecture Notes in Computer Science, Springer, 2012, vol. 7273.
- [27] G. SALAÜN, X. ETCHEVERS, N. DE PALMA, F. BOYER, T. COUPAYE. *Verification of a self-configuration protocol for distributed applications in the cloud*, in "ACM Symposium on Applied Computing (SAC)", 2012.
- [28] A. SANGROYA, D. SERRANO, S. BOUCHENAK. *Benchmarking Dependability of MapReduce Systems*, in "31st IEEE Int. Symposium on Reliable Distributed Systems (SRDS)", IEEE, 2012.
- [29] A. SANGROYA, D. SERRANO, S. BOUCHENAK. *MRBS: Towards Dependability Benchmarking for Hadoop MapReduce*, in "Euro-Par 2012: Parallel Processing Workshops", Lecture Notes in Computer Science, Springer, 2012, vol. 7640.

Conferences without Proceedings

- [30] SOGUY MAK-KARE. GUEYE, E. RUTTEN, A. TCHANA. *Discrete control for the coordination of administration loops*, in "3rd International Workshop on Green and Cloud Management (GCM'2012)", 2012.
- [31] S. M.-K. GUEYE, N. DE PALMA, E. RUTTEN. *Discrete control for the coordination of administration loops*, in "7th International Workshop on Feedback Computing", 2012.
- [32] S. GUILLET, F. DE LAMOTTE, N. LE GRIGUER, E. RUTTEN, J.-P. DIGUET, G. GOGNIAT. *Modeling and Synthesis of a dynamic and Partial reconfiguration Controller*, in "22nd International Conference on Field Programmable Logic and Applications (Poster Session), FPL 2012", 2012.

Scientific Books (or Scientific Book chapters)

- [33] J. ARNAUD, S. BOUCHENAK. *Performance, Availability and Cost of Self-Adaptive Internet Services*, in "Performance and Dependability in Service Computing", V. CARDELLINI, E. CASALICCHIO, K. BRANCO, J. ESTRELLA, F. MONACO (editors), IGI Global, 2012, chap. 10.
- [34] D. POUS, D. SANGIORGI. *Enhancements of the coinductive proof method*, in "Advanced Topics in Bisimulation and Coinduction", D. SANGIORGI, J. RUTTEN (editors), CambridgeTracts in Theoretical Computer Science, Cambridge University Press, 2012, chap. 6, n^o 52.

References in notes

- [35] *Research Projects in Autonomic Computing*, 2003, IBM Research.
- [36] Z. LIU, H. JIFENG (editors). *Mathematical Frameworks for Component Software - Models for Analysis and Synthesis*, World Scientific, 2006.

-
- [37] M. ABADI, L. CARDELLI. *A theory of objects*, Springer, 1996.
- [38] J. ALDRICH, C. CHAMBERS, D. NOTKIN. *Architectural Reasoning in ArchJava*, in "Proceedings 16th European Conference on Object-Oriented Programming (ECOOP)", 2002.
- [39] S. ANDROUTSELLIS-THEOTOKIS, D. SPINELLIS. *A survey of peer-to-peer content distribution technologies*, in "ACM Comput. Surv.", 2004, vol. 36, n^o 4.
- [40] J. ARMSTRONG, M. WILLIAMS, C. WIKSTROM, R. VIRIDING. *Concurrent Programming in Erlang*, Prentice Hall, 1996.
- [41] L. BASS, P. CLEMENTS, R. KAZMAN. *Software Architecture in Practice*, SEI Series in Software Engineering, 2nd edition, Addison-Wesley, 2003.
- [42] 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.
- [43] B. BERSHAD, S. SAVAGE, P. PRZEMYSŁAW, 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.
- [44] 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.
- [45] 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.
- [46] 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. *Core: an operating system for many cores*, in "8th Usenix Symposium on Operating System Design and Implementation (OSDI '08)", 2008.
- [47] 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.
- [48] 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.
- [49] C. G. CASSANDRAS, S. LAFORTUNE. *Introduction to Discrete Event Systems*, 2nd edition, Springer, 2008.
- [50] G. CASTAGNA, J. VITEK, F. Z. NARDELLI. *The Seal Calculus*, in "Inf. Comput.", 2005, vol. 201, n^o 1.

-
- [51] R. COLLET. *The Limits of Network Transparency in a Distributed Programming Language*, Université Catholique de Louvain, Belgium, 2007.
- [52] G. COULSON, G. S. BLAIR, P. GRACE, F. TAĪANI, A. JOOLIA, K. LEE, J. UHEYAMA, T. SIVAHARAN. *A generic component model for building systems software*, in "ACM Trans. Computer Systems", 2008, vol. 26, n^o 1.
- [53] 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.
- [54] A. DEARLE. *Software Deployment, Past, Present and Future*, in "Future of Software Engineering (FOSE '07)", IEEE, 2007.
- [55] 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.
- [56] 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.
- [57] M. GOLM, M. FELSER, C. WAWERSICH, J. KLEINÖDER. *The JX Operating System*, in "USENIX Annual Technical Conference, General Track", USENIX, 2002.
- [58] G. HEISER, K. ELPHINSTONE, I. KUZ, G. KLEIN, S. PETERS. *Towards Trustworthy Computing Systems: Taking Microkernels to the Next Level*, in "Operating Systems Review", 2007, vol. 41, n^o 4.
- [59] J. HELLERSTEIN, Y. DAO, S. PAREKH, D. TILBURY. *Feedback Control of Computing Systems*, Wiley-Interscience, 2004.
- [60] M. HENNESSY, J. RATHKE, N. YOSHIDA. *SafeDpi: a language for controlling mobile code*, in "Acta Informatica", 2005, vol. 42, n^o 4-5.
- [61] M. HERLIHY, N. SHAVIT. *The Art of Multiprocessor Programming*, Morgan Kaufmann, 2008.
- [62] 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.
- [63] G. C. HUNT, J. R. LARUS. *Singularity: rethinking the software stack*, in "Operating Systems Review", 2007, vol. 41, n^o 2.
- [64] 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.
- [65] 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>.
- [66] G. KICZALES, J. DES RIVIÈRES, D. BOBROW. *The Art of the Metaobject Protocol*, MIT Press, 1991.

-
- [67] G. KICZALES. *Aspect-Oriented Programming*, in "ACM Computing Surveys", 1996, vol. 28, n^o 4, 154, <http://doi.acm.org/10.1145/242224.242420>.
- [68] 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.
- [69] 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.
- [70] M. N. KROHN, E. KOHLER, M. F. KAASHOEK. *Events Can Make Sense*, in "USENIX Annual Technical Conference", 2007.
- [71] K. LAU, Z. WANG. *Software Component Models*, in "IEEE Trans. Software Eng.", 2007, vol. 33, n^o 10.
- [72] G. LEAVENS, M. SITARAMAN. *Foundations of Component-Based Systems*, Cambridge University Press, 2000.
- [73] Y. LIU, S. SMITH. *Interaction-Based Programming with Classages*, in "Proceedings OOPSLA", 2005.
- [74] N. LYNCH. *Distributed Algorithms*, Morgan Kaufmann, 1996.
- [75] 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.
- [76] P. MELL, T. GRANCE. *The NIST Definition of Cloud Computing*, 2010, <http://csrc.nist.gov/groups/SNS/cloud-computing/>.
- [77] M. MILLER. *Robust Composition: Towards a Unified Approach to Access Control and Concurrency Control*, Johns Hopkins University, baltimor, Maryland, USA, 2006.
- [78] R. MILNER. *Communicating and mobile systems : the π -calculus*, Cambridge University Press, 1999.
- [79] J. MITCHELL. *Foundations for Programming Languages*, MIT Press, 1996.
- [80] M. ODERSKY, M. ZENGER. *Scalable component abstractions*, in "Proceedings OOPSLA", 2005.
- [81] 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.
- [82] J. RISSON, T. MOORS. *Survey of research towards robust peer-to-peer networks: Search methods*, in "Computer Networks", 2006, vol. 50, n^o 17.
- [83] 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.

-
- [84] A. ROSSBERG. *The Missing Link - Dynamic Components for ML*, in "11th International Conference on Functional Programming", ACM Press, 2006.
- [85] D. SANGIORGI. *On the origins of bisimulation and coinduction*, in "ACM Trans. Program. Lang. Syst.", 2009, vol. 31, n^o 4.
- [86] D. SANGIORGI. *Bisimulation for Higher-Order Process Calculi*, in "Information and Computation", 1996, vol. 131, n^o 2.
- [87] D. SANGIORGI, D. WALKER. *The π -calculus: A Theory of Mobile Processes*, Cambridge University Press, 2001.
- [88] P. SEWELL, J. LEIFER, K. WANSBROUGH, F. Z. NARDELLI, M. ALLEN-WILLIAMS, P. HABOUZIT, V. VAPEIADIS. *Acute: High-level programming language design for distributed computation*, in "Int. Conf. Functional Programming", 2005.
- [89] D. SHAH. *Gossip Algorithms*, in "Foundations and Trends in Networking", 2009, vol. 3, n^o 1.
- [90] M. SHAW, D. GARLAN. *Software Architecture: Perspectives on an Emerging Discipline*, Prentice-Hall, 1996.
- [91] A. SILBERSCHATZ, P. B. GALVIN, G. GAGNE. *Operating System Concepts*, 8th edition, Wiley, 2008.
- [92] C. SZYPERSKI. *Component Software, 2nd edition*, Addison-Wesley, 2002.
- [93] G. TEL. *Introduction to Distributed Algorithms*, 2nd edition, Cambridge University Press, 2001.
- [94] P. VAN ROY, S. HARIDI. *Concepts, Techniques and Models of Computer Programming*, MIT Press, 2004.
- [95] N. WIRTH. *Project Oberon: The Design of an Operating System and Compiler*, ACM Press, 1992.
- [96] 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.