INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Project-Team Sardes

System Architecture for Reflective Distributed Computing Environments

Grenoble - Rhône-Alpes

THEME COM

Activity Report

2008

Table of contents

SARDES is a project team of INRIA Grenoble-Rhône-Alpes and a research team of LIG (Grenoble Informatics Laboratory), a joint research unit (UMR 5217) of Centre National de la Recherche Scientifique (CNRS), Institut National Polytechnique de Grenoble (INPG) and université Joseph Fourier (UJF).

1. Team

Research Scientist

Jean-Bernard Stefani [Team leader, research director, Ingénieur Général des Télécommunications]
Alan Schmitt [research scientist]
Damien Pous [research scientist, Centre National de la Recherche Scientifique, from Oct. 2008]
Vivien Quéma [research scientist, Centre National de la Recherche Scientifique]

Faculty Member

Sara Bouchenak [associate professor, université Joseph Fourier]
Fabienne Boyer [associate professor, université Joseph Fourier]
Noël De Palma [associate professor, Institut National Polytechnique de Grenoble]
Olivier Gruber [professor, université Joseph Fourier, HdR]
Sacha Krakowiak [professor emeritus, université Joseph Fourier, HdR]
Renaud Lachaize [associate professor, université Joseph Fourier]
Jacques Mossière [professor, Institut National Polytechnique de Grenoble, HdR]

Technical Staff

Nikos Parlavantzas
Valerio Schiavoni
Alessio Pace
Lionel Debroux
Fabien Mottet

PhD Student

Jean Arnaud [government grant]
Benoît Claudel [government grant]
Stéphane Fontaine [government grant]
Fabien Gaud [government grant]
Jakub Kornaś [Inria grant (contract), till Oct. 2008]
Michaël Lienhardt [government grant]
Sergueï Lenglet [government grant]
Willy Malvaud [Inria grant (contract)]
Claudio Mezzina [Inria grant (contract), from Oct. 2008]
Juraj Polakovic [France Telecom RD, till Jun. 2008]
Jérémy Philippe [Inria grant (contract)]
Sylvain Sicard [Inria grant (contract)]
Christophe Taton [government grant, till November 2008]
Thomas Braibant [government grant, from Oct. 2008]
Ludovic Demontes [government grant, from Oct. 2008]
Sylvain Genevès [government grant, from Oct. 2008]

Administrative Assistant

Diane Courtiol

2. Overall Objectives

2.1. Overall objectives

The overall goal of the SARDES project is to develop concepts, languages, software tools and infrastructure (operating systems, middleware) to build adaptable, dependable and manageable distributed systems.

To contribute to the above goal, the project 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 infrastructure required to build scalable, self-configuring distibuted systems.

In line with these objectives, the project conducts research in several areas:

- *Component models and foundations*. Component-based software engineering and software architecture are now well established fields of study. Their semantical foundations, however, are still not firmly understood, in particular when dealing with dynamic and heterogeneous architectures. We develop new reflective component models and study their formal semantics, mostly through the definition of new process calculi (to capture the operational essence of component models), the study of their behavioral theory, and the development of associated proof techniques. Previous works in this area included the definition of the Fractal component model [3], of the M-calculus [10], and of the Kell calculus [11].
- *Programming languages*. In order to simplify the work of programmers, we design languages and tools, such as type systems, adapted to component-based programming and the specific problems we want to address. We develop languages for component-based programming, bidirectional data manipulation, and type systems for correct component assemblages. Previous works in this area included the development of logical methods and algorithms for the static analysis of XPath [6], and the definition of combinators for bidirectional tree transformations [5].
- *Distributed algorithms*. Many algorithms that are considered optimal in theory fail to live up to their potential in practice. Our goal is to help closing this gap between theory and practice by proposing complexity models closer to reality and by revisiting classical abstractions considering these new models. We are currently applying this approach to the design of algorithms for total order broadcast, eventual consistency, distributed shared memories, and publish/subscribe systems over P2P networks. Previous works in this area include the development of a high-throughput total order braodcast for cluster systems [8], and a high-throughput distributed atomic storage algorithm [7].
- *Operating system and virtual machine technology*. Adaptability and dependability properties require basic support from low-level softare infrastructure, notably operating system and virtual machine layers. We develop component-based operating system and virtual machine technology enabling provable component isolation and safe dynamic updates, both at the application level and at the infrastructure level. Previous works in this area include the development of the Think framework [4], and the development of a flexible code generation tool chain for heterogeneous architecture descriptions [9],
- *Autonomous system management*. Autonomic computing aims at providing systems and applications with self-management capabilities, including self-configuration (automatic organization according to a specified policy), self-optimization (continuous performance management), self-healing (automatic failure management), and self-protection (taking preventive measures and defending against malicious attacks). We pursue a control-based and architecture-based approach to autonomic computing. Previous results included the development of the Jade framework [1] and initial results on self-optimization [2].

3. Scientific Foundations

3.1. Introduction

Our approach to the construction of adaptable, dependable and manageable distributed systems is *architecture-based*. "Architecture-based" means that a system software architecture (and associated component-base structure) plays a pivotal role for enabling adaptations and management operations, which are seen primarily as actions inducing architectural changes in a system.

Our approach to system construction relates to, and builds on results in several research fields in computer science, mostly: component-based software engineering, software architecture, reflective systems and languages, programming languages, concurrency theory, type systems, distributed algorithms, operating systems, distributed systems and middleware, autonomic computing. In what follows, we (briefly) discuss some of these connections, mention relevant litterature, and point at open issues that are relevant to SARDES' work.

3.2. Components and semantics

The primary foundations of the software component technology developed by SARDES stem from component-based software engineering [72], and software architecture [71]. Both can be traced back to an early vision of systematically produced software [65], and are primarily concerned with the construction of software systems by composition or assembly of software components, with well-defined interfaces and explicit dependencies. 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 [49]. Several component models and associated architecture description languages have been devised over the past fifteen years (see e.g. [64] for an analysis of recent component models, [47] for a recent comparison of software component models based on a common design problem, and [66], [51] for surveys of architecture description languages).

To natively support configurability and adaptability in systems, SARDES' component technology also draws from ideas in reflective languages [61], and reflective middleware [63], [50] (a reflective system is one that maintains a manipulable causally connected representation of itself). Reflection can be used both to increase the separation of concerns in a system architecture, as pioneered by aspect-oriented programming [62], and to provide systematic means for modifying a system implementation.

The semantical foundations of component-based and reflective systems are not firmly established, however. Despite much work on formal foundations for component-based systems (see e.g. [45], [46] for surveys of relevant approaches), 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 SARDES' component-based technology, we have adopted a kernel language approach, where we try to model relevant constructs and capabilities in a small language or 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 [67], or the analysis of object-orientation [48]. The developments around the π-calculus, including behavioral theory and coinductive proof techniques [70], process calculi with localities [52], [53], [54], and higher-order versions [69], [57], provide the background for SARDES' work on process calculi for component-based programming.

3.3. Autonomous system management

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 administration as a *control* activity, involving an event-reaction loop: the administration system detects events that may alter the ability of the administered 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, reconfiguration, resource management, performance management, and fault management.

Up to now, administration tasks have mainly been performed by persons. 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 [68] 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 remarks have motivated 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 [59]. Several research projects [44] are active in this area. [60], [58] 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 (see e.g. [55] for an assessment), and techniques for constructing control algorithms in the decision stage of administration feedback loops (see e.g. [56] for a survey of control-theoretic approaches).

4. Application Domains

4.1. Application Domains

Keywords: *electronic commerce*, *embedded systems*, *multimedia*, *power supply*, *systems administration*, *telecommunications*.

SARDES develops generic tools for distributed applications, in the form of languages, middleware, system kernels, and information servers. These tools are useful in application domains that have one or more of the following properties.

- Need for dynamic adaptation of infrastructures or applications;
- Use of high performance information servers and clusters.
- Deployment and management of information servers and clusters.
- Cooperation using shared distributed information;
- Mobility of users, information and services;

Applications are important for a project like SARDES, in which experimental aspects play a significant part. They provide testbeds to evaluate prospective designs, and they help us establish links with industrial partners, allowing us to transfer results and to identify relevant research problems.

In recent years, SARDES has been active in the following application areas:

1. Electronic commerce: flexible access to remote services by mobile users, efficient transaction management.
2. Embedded computing: development of custom made kernels for specific applications (robotics, real time), dynamically reconfigurable kernels;
3. Multimedia applications: dynamic adaptation of a videoconferencing system for use by mobile clients;
4. Power supply : administration and monitoring of power supply networked equipment, e.g. uninterrupted power supply units.
5. Telecommunications : administration of large scale networks, servers and caches for the Web, management of configurable added value services;

5. Software

5.1. Introduction

Keywords: *J2EE*, *autonomic computing*, *benchmark*, *cluster*, *clustered databases*, *communication middleware*, *publish-suscribe*, *transactions*.

Software development is an important aspect of the activity of SARDES. This software serves as a testbed to apply, validate and evaluate the methods and tools developed in the project.

Software developed in SARDES, or to which SARDES heavily contributed, is available in the OW2 open source code base (see 8.2), which is accessible at http://www.ow2.org/.

We list below software which has been under development by SARDES in 2008. We do not list software produced by the SARDES project in the past years which has not been further developed by SARDES in 2008.

5.2. DepOz, a Framework for Building Self-deployable Components

Contact : Christophe Taton, Jean-Bernard Stefani

DEPOZ is a framework developed with the Oz/Mozart programming language. It supports the construction of complex distributed software architectures, the programming of complex deployment processes, and the construction of self-configurable components. DEPOZ comprises two libraries, FRUCTOZ and LACTOZ. FRUCTOZ implements the Fractal model in Oz, and provides support for the contruction of complex deployment processes. LACTOZ provides support for navigating, querying and monitoring a distributed dynamic architecture.

5.3. Dream, a Framework for Building Asynchronous Middleware

Contact : see OW2 page.

DREAM is a component-based framework dedicated to the construction of communication middleware. It provides a component library and a set of tools to build, configure and deploy middleware implementing various communication paradigms: group communications, message passing, event-reaction, publish-subscribe, etc. DREAM builds upon the FRACTAL component framework, which provides support for hierarchical and dynamic composition.

DREAM is an OW2 project distributed under an LGPL license.

See http://dream.objectweb.org

5.4. Jade, a Framework for Building Autonomic Management Systems

Contact : see OW2 page.

JADE is a framework for the construction of autonomic systems using the FRACTAL reflective component model. The controlled system is described in terms of an assembly of components equipped with elementary management capabilities. This description, in turn, is the base of the feedback control loops that implement various self-management functions. Legacy applications are managed by wrapping them into components. Since Jade is itself developed using the component model, the autonomic functions also apply to Jade.

JADE is available under a Cecill-C open source licence, as part of the JASMINe OW2 open source project. See http://forge.objectweb.org/projects/jasmine.

5.5. Fractal/Cecilia: Fractal component-based programming in C

Contact : see OW2 page.

FRACTAL is a component-based framework developed in cooperation by SARDES and FTR&D. Cecilia merges the original FRACTAL ADL compiler and the Think ADL compiler into an implementation of FRACTAL in C. Cecilia also extends the original FRACTAL ADL compiler into a retargettable compiler for heterogeneous architecture descriptions.

Cecila is available under an LGPL licence as part of the FRACTAL OW2 open source project. See http://forge.objectweb.org/projects/fractal.

6. New Results

6.1. Introduction

In this section, we present new results in the main areas covered by SARDES: Component Models and Frameworks (6.2), Programming Languages (6.3), Distributed Algorithms (6.4), Operating System and Virtual Machine Technology (6.5), and Autonomous System Management (6.6).

6.2. Component Models and Foundations

Participants: Damien Pous, Alan Schmitt, Jean-Bernard Stefani, Thomas Braibant, Sergueï Lenglet, Michaël Lienhardt, Claudio Mezzina.

6.2.1. Program equivalences in higher-order calculi

Much progress has been accomplished in 2008 in the study of higher-order calculi as models of component-based programs. During Alan Schmitt's sabbatical leave at University of Bologna, the collaboration with Davide Sangiorgi and several of his students resulted in the design of a core higher-order calculus that is rich enough to be Turing complete, yet fundamental enough to have a decidable notion of program equivalence. This work was presented at the LICS international conference [24].

In parallel, we have continued to investigate the impact of the addition of a passivation operator, required to faithfully model live-cycle properties of components, to the proof of program equivalence. We have showed that the interaction of passivation and restriction prevents the efficient testing of program equivalence [35], and very recently discovered a proof technique, called complementary bisimulation, to characterize program equivalence [34] for higher-order process calculi with passivation and restriction. Both of these results are currently submitted for publication.

6.2.2. Fractal specification

We have also developed a formal specification of the Fractal component model using the Alloy specification language [37]. The specification covers all the elements of the informal Fractal specification, it lifts all the ambiguities of the informal specification, and it provides a truly programming-language independent specification of the Fractal model in first-order relational logic. This work constitute a basis for ongoing developments towards a formalization of the Fractal architecture description language.

6.3. Programming Languages

Participants: Alan Schmitt, Jean-Bernard Stefani, Michaël Lienhardt, Claudio Mezzina.

Work has continued on the definition of a type system for the assemblage of communicating components. A first type system was published this year [25]. The paper proves that the type inference problem for this type system is undecidable, and provides a semi-algorithm for type inference. In parallel, we have developed a second, more practical type system with decidable type inference. A first prototype implementation has been developed and has been integrated with the Fractal Toolchain. This implementation has enabled us to check the use of our type analyser on large Dream configurations and to show that the type system can be used with other component models, such as Click. This work is currently submitted for publication.

6.4. Distributed Algorithms

Participants: Vivien Quéma, Alan Schmitt, Jean-Bernard Stefani, Willy Malvaud, Valerio Schiavoni, Alessio Pace.

6.4.1. Gossip-based information dissemination

Gossip-based information dissemination protocols are considered easy to deploy, scalable and resilient to network dynamics. Load-balancing is inherent in these protocols as the dissemination work is evenly spread among all nodes. Yet, large-scale distributed systems are usually heterogeneous with respect to network capabilities such as bandwidth. In practice, a blind load-balancing strategy might significantly hamper the performance of the gossip dissemination. In collaboration with Rachid Guerraoui's group at EPFL and the INRIA ASAP team, we have proposed a new protocol, called HEAP (HEterogeneity-Aware gossip Protocol), where nodes dynamically adapt their contribution to the gossip dissemination according to their bandwidth capabilities. Using a continuous, itself gossip-based, approximation of relative bandwidth capabilities, HEAP dynamically leverages the most capable nodes by increasing their fanout, while decreasing by the same proportion those of less capable nodes. HEAP preserves the simplicity and proactive (churn adaptation) nature of gossip, while significantly improving its effectiveness. HEAP has been extensively evaluated in the context of a video streaming application. HEAP significantly improves the perceived quality of the streaming over standard gossip protocols.

6.4.2. NAT-resilient gossip peer sampling

Gossip peer sampling protocols now represent a solid basis to build and maintain peer to peer (p2p) overlay networks. They typically provide peers with a random sample of the network and maintain connectivity in highly dynamic settings. They rely on the assumption that, at any time, each peer is able to establish a communication with any of the peers of the sample provided by the protocol. Yet, this ignores the fact that there is a significant proportion of peers that now sit behind NAT devices, preventing direct communication without specific mechanisms. This has been largely ignored so far in the community. Our experiments demonstrate that the presence of NATs, introducing some restrictions on the communication between peers, significantly hurts both the randomness of the provided samples and the connectivity of the p2p overlay network, in particular in the presence of high rate of peers arrivals, departures and failures. In collaboration with the INRIA ASAP team, we have proposed a NAT-resilient gossip peer sampling protocol, called Nylon, that accounts for the presence of NATs. Nylon is fully decentralized and spreads evenly the extra load caused by the presence of NATs, between peers. Nylon ensures that a peer can always establish a communication, and therefore initiates a gossip, with any peer in its sample. This is achieved through a simple, yet efficient mechanism, establishing a path of relays between peers. Our results show that the randomness of the generated samples is preserved, that the connectivity is not impacted even in the presence of high churn and a high ratio of peers sitting behind NAT devices.

6.4.3. Byzantine fault tolerant replication

State machine replication (SMR) is a software technique for tolerating failures using commodity hardware. The critical service to be made fault-tolerant is modeled by a state machine. Several, possibly different, copies of the state machine are then placed on different nodes. Clients of the service access the replicas through a SMR protocol which ensures that, despite concurrency and failures, replicas perform client requests in the same order. Two objectives underly the design and implementation of a SMR protocol: robustness and performance. Robustness conveys the ability to ensure availability (liveness) and one-copy semantics (safety) despite failures and asynchrony. Performance measures the time it takes to respond to a request (latency) and the number of requests that can be treated per time unit (throughput). The most robust protocols are those that tolerate (a) arbitrarily large periods of asynchrony, and (b) arbitrary (Byzantine) failures of any client as well as up to one-third of the replicas. The development of Byzantine fault-tolerant SMR protocols is notoriously difficult. In collaboration with Rachid Guerraoui's group at EPFL, we have proposed a generic abstraction to simplify this task. We view a BFT protocol as a, possibly dynamic, composition of instances of our abstraction, each instance developed and analyzed independently. To illustrate our approach, we have developed two new

BFT protocols. Among all protocols we know of, the first has the lowest latency in synchronous periods that are free from contention and failures; the second has the highest peak throughput in failure-free and synchronous periods.

6.5. Operating system and virtual machine technology

Participants: Fabienne Boyer, Noël De Palma, Olivier Gruber, Renaud Lachaize, Vivien Quéma, Jean-Bernard Stefani, Fabien Gaud.

The work in this area within SARDES is currently undergoing a transition, with two notable directions: first, a new research direction on virtual machine support for component-based programming, and second an increased emphasis on operating system services for multiprocessor systems on chip (MPSoC) or multicore architectures.

The work on virtual machine support for component-based programming stems from the need to revisit the design of current object-oriented virtual machines such as the Java virtual machine or the Microsoft .Net Common Language Runtime in order to provide better support for components, including native support for software update and dynamic reconfiguration, memory protection and component isolation, as well as support for event-driven programming. This work has just begun and we will not report on it here.

6.5.1. EfÞcient multiprocessor support for event-driven programming

Event-driven programming is a popular approach for the development of robust applications such as network servers and clients. The strength of this model mainly lies in its expressivity (fine-grain management of concurrency including asynchronous network and disk I/O), portability, support for dynamic reconfiguration, and, in some cases, lower memory and better performance than other design strategies. However, a genuine event-based library cannot take advantage of multiprocessor platforms since it relies on a single thread executing the main processing loop. The main approach proposed to date for supporting multiprocessor execution within an event-based library consists in: (i) enforcing correctness by allowing programmers to incrementally inject support for safe parallel execution through annotations specifying events that can run concurrently and (ii) dynamically balancing the load on the multiple cores via workstealing. Our ongoing research work strives to improve both the programming interface and the performance of event-based libraries for multicore platforms. Most of our work so far has focused on the latter aspect. We propose novel event dispatching heuristics and design choices to improve the behavior of workstealing, taking into account parameters such as cache affinities of tasks and inherent synchronization costs. The first evaluations of our runtime on high performance network servers show that it achieves better throughput than existing solutions. Part of this work is reported in the Master thesis of Ludovic Demontes and Sylvain Genevs.

6.6. Autonomous System Management

Participants: Sara Bouchenak, Fabienne Boyer, Noël De Palma, Olivier Gruber, Jean-Bernard Stefani, Jean Arnaud, Jakub Kornaś, Jérémy Philippe, Sylvain Sicard, Christophe Taton.

We have continued the development of the JADE framework for autonomous distributed system management. We report below new results obtained this year.

6.6.1. Automated deployment and configuration

We have continued the development of facilities for the automated deployment and configuration of distributed software architectures. Two main results have been obtained this year: the development of new deployment and reconfiguration support in Jade, and the development of the DEPOZ framework for the construction of self-deployable and self-configurable components.

The new deployment and reconfiguration support in Jade introduces a new module and package system for Fractal components in Java, that overcomes limitations of the OSGI framework, previously used in Jade. The new module and package system allows a uniform and system-wide management of component executables, and supports dynamic component code updates. The system is described in detail in Jakub Kornaś' PhD thesis [12].

The DEPOZ framework supports the construction of complex distributed software architectures, the programming of complex deployment processes, and the construction of self-configurable components. It complements the previous work with the ability to program highly parameterized deployment workflows and to construct self-deployable and self-configurable distributed components. It is described in detail in Chrstophe Taton's PhD thesis [14].

6.6.2. *Autonomous Management of Performance and QoS*

The goal of self-optimization is to maintain optimal (or near-optimal) system performance and quality-of-service (QoS) in spite of wide variations of the load or of the amount of available resources. Performance may be measured by various criteria, such as average response time or average throughput for an Internet service, or bounded jitter for a video server, etc. QoS may reflect several service characteristics such as service availability which may be measured as service abandon rate in an Internet service.

We have proposed simple heuristics-based approaches to self-optimization of cluster-based Internet services through dynamic resource provisioning. These approaches were successfully applied to replicated database systems through dynamic resource (un-)provisioning upon database load variation. Dynamic resource provisioning was applied in conjunction with replication protocols and group communication protocols as part of the GORDA European project [39], [40], [43]. The proposed approaches were also applied to messaging systems in the context of the JORAM open-source Java message-oriented middleware hosted by the OW2 consortium. We have used the Jade framework to build autonomic capabilities on top of the JORAM middleware, and described how to (i) dynamically adapt the load distribution among the servers (load-balancing aspect) and (ii) dynamically adapt the replication level (provisioning aspect) [30], [32], [14]. Finally, cluster-based multi-tier enterprise applications were also used as a testbed of the proposed dynamic resource provisioning policies of self-optimization. Among the reasearch issues tackled in this work, we can cite system oscillation due to potential concurrent reconfigurations on the distributed multi-tier system. We proposed system oscillation prevention techniques that follows a software architecture-based approach [38], [14].

The above-mentioned work mainly proposes heuristics-based techniques that provide a best-effort behavior and aims at keeping the managed system near-optimal. We have also proposed new techniques that guarantee system optimality with strict guarantees on service level objectives (SLOs) such as maximum latency and maximum abandon rate for Internet services. The proposed solutions make use of queuing theory to modeling and capacity planning of cluster-based multi-tier enterprise systems [33]. We also cooperate with the NeCS INRIA research group to apply control theory to modeling and capacity planning of database and server systems [36]. In the context of this cooperation, the PhD thesis of L. Malrait is co-advised by Nicolas Marchand (NeCS) and Sara Bouchenak. A patent proposal on this work is ongoing.

7. Contracts and Grants with Industry

7.1. Collaboration with France-Telecom Orange Labs

Participants: Jean-Bernard Stefani, Vivien Quéma, Jakub Kornaś, Nikos Parlavantzas.

SARDES maintains an active collaboration with France-Telecom Orange Labs. This collaboration concerns the following aspects:

- Further developments of the FRACTAL component model.
- The development of the JADE framework, and its application to large scale distributed environments, see 8.1.2,8.1.3.

7.2. National Actions

7.2.1. *ASR Network*

SARDES is a member of the CNRS research network GDR 725 ASR “Architecture, Systme et Rseau”. See http://asr.cnrs.fr/.

7.2.2. Project SelfWare (ANR)

Participants: Noël De Palma, Fabienne Boyer, Jérémy Philippe, Sylvain Sicard.

The goal of project Selfware is to provide a framework for autonomous management of large scale distributed applications including legacy software systems. The expected results are reducing operator mistakes (a major cause of failure for large Internet services), saving hardware ressources, and minimizing administrator efforts. The main challenge is to be able to manage any arbitrary complex legacy system with minimum effort. This framework will be evaluated for two actual middleware infrastructures: (i) a clustered J2EE application Server (JONAS) and (ii) a Message Oriented Middleware (JORAM). Experimental scenarios will illustrate self-optimizing and self-healing.

The project partners are INRIA Grenoble-Rhône-Alpes; France Telecom Orange Labs; Ecole des Mines de Nantes; Bull; IRIT/ENSEEIHT, Toulouse.

The project runs from June 2006 to December 2008.

7.2.3. Project Modyfiable (ANR)

Participants: Alan Schmitt, Jean-Bernard Stefani, Michaël Lienhardt, Sergueï Lenglet.

The goal of Modyfiable is to study the formal basis of "dynamic modularity", i.e. aspects related to dynamic creation, deployment, linking, and update of modules and components, considered as first-class programming abstractions. In contrast to classical modularity, which only deals with modular program construction, dynamic modularity is concerned with the manipulation and control of modules and components, understood as units of modularity and deployment, during execution.

The project partners are: INRIA Grenoble-Rhône-Alpes (SARDES project team); École Normale Supérieure de Lyon (Plume team).

The project runs from January 2006 to December 2008.

7.2.4. Project ScorWare (ANR)

Participants: Vivien Quéma, Renaud Lachaize, Valerio Schiavoni.

The SCOrWare project aims at providing an open source implementation of the recent Service Component Architecture (SCA) specifications defined by the Open SOA collaboration, an industrial consortium in the domain of software engineering, including BEA Systems, IBM Corporation, IONA Technologies, Oracle, Red Hat, Rogue Wave Software, Siemens, Sun Microsystems, and Sybase. Briefly, SCA defines a new architecture and programming model for Service Oriented Applications (SOA) based on the component paradigm and supporting several service description languages like WSDL and Java interfaces, several programming languages such as Java, C++, and BPEL, several communication protocols between applications such as SOAP, CORBA, Java RMI, and JMS.

The project aims to build a configurable and manageable SCA platform built upon the Fractal component model, with advanced functionalities such as dynamic reconfiguration of SCA component assemblies, a binding factory supporting different communication protocols between SCA components, a transaction service, a semantic trading service of SCA components, a deployment engine of autonomous SCA architecture and graphical administration consoles. The project partners are INRIA; Artenum; EBM WebSourcing; Edifixio; INT, Evry; IRIT, Toulouse; Obeo; OpenWide.

The project runs from January 2007 to December 2008.

See http://www.scorware.org

7.2.5. Project Flex-e-Ware (ANR)

Participants: Vivien Quéma, Renaud Lachaize, Alessio Pace.

The Flex-e-Ware project aims at defining software development tools and mechanisms dedicated to flexible, reconfigurable embedded systems. It aims to develop a common platform for the development of component-based embedded systems, based on the integration of the FRACTAL and Lightweight CCM component models.

The project partners are INRIA (Adam and Sardes project teams); CEA List; ENST, Paris; LIP6, Paris; France Telecom; Schneider Electric; STMicroelectronics; Teamlog; Trialog; Thales.

The project runs from January 2007 to December 2009.

See http://www.flex-eware.org/

7.2.6. *Project Aravis (ANR-Minalogic)*

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

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 STMicrolectronics, CEA-LETI, TIMA and INRIA (Necs and Sardes project teams). The project runs from October 2007 to September 2010.

8. Other Grants and Activities

8.1. EC-funded projects

8.1.1. *IST Project Gorda*

Participants: Sara Bouchenak, Christophe Taton.

Gorda (Open Replication of Databases) is an IST STREPS project, which aims at (i) promoting the interoperability of databases and replication protocols by defining generic architecture and interfaces that can be standardized; (ii) providing general purpose and widely-applicable database systems; and (iii) providing uniform techniques and tools for managing secure and heterogeneous replicated database systems. The project partners are Universidade do Minho (R. Oliveira), Università della Svizzera Italiana (F. Pedone), Universidade de Lisboa (L. Rodrigues), INRIA (SARDES project), Emic Networks (Finland) and MySQL AB (Sweden). The main contribution of SARDES is the C-JDBC technology, and the development of frameworks and tools for its use.

The project runs from October 2004 to April 2008.

See http://gorda.di.uminho.pt

8.1.2. *IST Project SelfMan*

Participants: Jean-Bernard Stefani, Alan Schmitt, Vivien Quéma, Willy Malvaud.

“Abnormal” events such as software updates, faults, threats, and performance hotspots become normal and even frequent occurrences in large distributed systems. The goal of SelfMan, an IST STREPS project, is to handle these events automatically by making the systems self managing: the systems will reconfigure themselves to handle changes in their environment or requirements without human intervention but according to high-level management policies. The focus is on four axes of self management, namely self configuration, self healing, self tuning, and self protection. The project partners are Université catholique de Louvain, Belgium; Royal Institute of Technology (Kungliga Tekniska Högskolan), Sweden; Institut National de Recherche en Informatique et Automatique (INRIA), France; France Telecom R& D, France; Konrad-Zuse-Zentrum für Informationstechnik Berlin, Germany; E-Plus Mobilfunk GmbH & Co. KG, Germany; and National University of Singapore, Singapore.

The project runs from June 1, 2006 to May 31, 2009.

8.1.3. IST Project Grid4All

Participants: Noël De Palma, Jean-Bernard Stefani, Nikos Parlavantzas.

Grid4All aims at enabling domestic users, non-profit organisations such as schools, and small enterprises, to share their resources and to access massive Grid resources when needed, envisioning a future in which access to resources is democratised, readily available, cooperative, and inexpensive. Examples include home users of an image editing application, school projects like volcanic eruption simulations, or small businesses doing data mining. Cooperation examples include joint homework between pupils, and international collaboration such as the edition of a multilingual newsletter between schools from different countries.

Grid4All goals entails a system pooling large amounts of cheap resources; a dynamic system satisfying spikes of demand; using self-management techniques to scale and adapt to changes in environment; supporting isolated, secure, dynamic, geographically distributed user groups and using secure peer-to-peer techniques to federate large numbers of small-scale resources into large-scale Grids. The technical issues addressed are aspects of security, support for multiple administrative and management authorities, self-management by combining the strong points of structured overlay P2P networks and that of component models, on-demand resource allocation, heterogeneity, and fault tolerance.

The partners of the project are France Telecom (FT R&D); Institut National de Recherche en Informatique et en Automatique (INRIA); The Royal Institute of Technology (KTH); Swedish Institute of Computer Science (SICS); Institute of Communication and Computer Systems (ICCS); University of Piraeus Research Center (UPRC); Universitat Politecnica de Catalunya (UPC); Rededia S.L. (REDEDIA);

The project runs from January 2006 to December 2008.

8.1.4. IST Project OMP

Participants: Renaud Lachaize, Vivien Quéma, Jean-Bernard Stefani.

Participants : Renaud Lachaize, Vivien Quma, Jean-Bernard Stefani

OMP (Open Media Platform) is an IST STREP project, which aims at defining an open service infrastructure for media-rich end-user devices (such as smartphones) to address software productivity and optimal service delivery challenges. The OMP approach combines 1) component based programming to dynamically manage and assemble media services, 2) efficient compilation flows to bring software portability with limited performance penalties and 3) standard API improvements to keep the execution of media components under QoS control, including the power management aspect. The project partners are STMicroelectronics (Italy), NXP (Belgium/The Netherlands), Incoras (Ireland), Movial (Finland), Fraunhofer Heinrich Hertz Institute (Germany), Politecnico di Milano (Italy), INRIA (Alchemy and Sardes projects). The project runs from January 2008 to December 2009.

8.1.5. CoreGrid Network of Excellence (IST-004265)

SARDES is a member of CoreGrid, the European Research Network on Foundations, Software Infrastructures and Applications for large scale distributed, GRID and Peer-to-Peer Technologies, launched in 2004 for four years. Its objective is to coordinate European efforts in the area of Grid and Peer-to-Peer technologies. It gathers forty-two institutions. SARDES is contributing in the areas of programming models and software architecture (FRACTAL has been adopted as a basis for the component-based programming model of CoreGrid).

See http://www.coregrid.net/

8.2. OW2 Consortium

Participants: Jean-Bernard Stefani, Vivien Quéma, Fabienne Boyer, Noël De Palma, Renaud Lachaize, Valerio Schiavoni, Alessio Pace.

OW2 is an open-source software consortium that has been created as the successor of the ObjectWeb consortium put in place in 2002 by Bull, France Telecom and INRIA. Its goal is the development of open-source distributed middleware.

SARDES contributes to OW2 through its technical involvement in the development of software components and frameworks (e.g. FRACTAL, DREAM, THINK, JADE) and through participation in the management structures of the consortium (J.-B. Stefani, member of the Technical Council).

See http://www.ow2.org/

8.3. International Bilateral Collaborations

8.3.1. Europe

SARDES maintains collaboration with several research groups in Europe:

- École Polytechnique Fédérale de Lausanne: Distributed Programming Laboratory (Prof. Rachid Guerraoui). Distributed algorithms. Contact persons in SARDES: V. Quéma.
- Università di Roma, La Sapienza, Dipartimento di Informatica e Sistemistica (Prof. Roberto Baldoni). Collaboration on distributed algorithms for large scale systems (post-doctoral stay of Vivien Quéma, 2005-2006). Contact person in SARDES: V. Quéma.
- Université Catholique de Louvain (Belgium): *Department of Computing Science and Engineering* (Prof. Peter Van Roy). Collaboration on component-based distributed programming, autonomic systems. Contact person in SARDES: J.B. Stefani.
- Swedish Institute of Computer Science (SICS, Stockholm): *Computer System Laboratory* (Prof. Seif Haridi). Collaboration on large scale distributed systems management, component-based distributed programming (9-month visit of Noel De Palma in 2008). Contact persons in SARDES: J.B. Stefani, N. De Palma.
- Università di Bologna (Italy): *Department of Computer Science* (Prof. Davide Sangiorgi). Collaboration on distributed process calculi, type systems for components (one-year visit of Alan Schmitt, 2007-2008). Contact persons in SARDES: A. Schmitt, J.B. Stefani.

8.3.2. U.S.A.

Collaboration is under way with the University of Pennsylvania, Philadelphia (Prof. Benjamin C. Pierce), on the Harmony universal synchronizer and bi-directional languages: distributed algorithms, programming language design, type systems (visits, shared software and joint publications). Contact person in SARDES: A. Schmitt.

9. Dissemination

9.1. Community Service

- J.-B. Stefani is a member of the editorial board of the journal *Annals of Telecommunications*.
- J.B. Stefani is a member of the Steering Committee of the ACM/IFIP Middleware conference series, and of the program committees of ACES-MB 2008, CHPBC 2008, DOA 2008, FORTE 2008, Middleware 2008, WADS 2008.
- J.B. Stefani is a member of the OW2 Technical Council, and a member of the Technology Council of STMicroelectronics.
- A. Schmitt is a member of the program committee of the International Conference on Functional Programming (ICFP 2008).

- A. Schmitt is the chair of the Journes Francophones des Langages Applicatifs (JFLA 2009), both as the program committee chair and as local organizer.
- A. Schmitt is the local organizer for the Ocaml Meeting 2009, which will take place in February 2009.
- Vivien Quéma is in the organization committee of the second International Conference on Distributed Event-Based Systems (DEBS 2008).
- Vivien Quéma organized the first Winter School on Hot Topics in Distributed Computing. The school was co-sponsored by INRIA and Eurosys.
- S. Bouchenak is an officer of ACM SIGOPS France.
- S. Bouchenak is a member of CES INRIA - Grenoble (Comit des Emplois Scientifiques).
- S. Bouchenak was a member of CSE at University of Grenoble I, and Institut Polytechnique de Grenoble (Commission de Spcialistes d'Etablissement).
- S. Bouchenak is a member of the program committees of CFSE 2008 and CFSE 2009.

9.2. University Teaching

S. Bouchenak, F. Boyer, N. De Palma, O. Gruber, R. Lachaize, J. Mossière, and V. Quéma have taught several operating systems and distributed systems courses at the M.S. and M.Eng. levels, both at Institut National Polytechnique de Grenoble and at université Joseph Fourier, Grenoble, France. Most of our Ph.D. students contributed to these courses as teaching assistants.

9.3. Participation in Seminars, Workshops, Conferences

Several members of SARDES attended various scientific conferences and workshops. See the relevant section of the Bibliography for details. Most of the publications of the project are available on line from the SARDES web site:

http://sardes.inrialpes.fr/

10. Bibliography

Major publications by the team in recent years

[1] S. BOUCHENAK, F. BOYER, S. KRAKOWIAK, D. HAGIMONT, A. MOS, N. DE PALMA, V. QUÉMA, J.-B. STEFANI. *Architecture-Based Autonomous Repair Management: An Application to J2EE Clusters*, in "24th IEEE Symposium on Reliable Distributed Systems (SRDS 2005)", IEEE Computer Society, 2005.

[2] S. BOUCHENAK, N. DE PALMA, D. HAGIMONT, C. TATON. *Autonomic Management of Clustered Applications*, in "IEEE International Conference on Cluster Computing", 2006.

[3] 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)", vol. 36, n° 11-12, 2006.

[4] J.-PH. FASSINO, J.-B. STEFANI, J. LAWALL, G. MULLER. *THINK: A Software Framework for Component-based Operating System Kernels*, in "Proceedings of Usenix Annual Technical Conference, Monterey (USA)", June 10th-15th 2002.

[5] 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.", vol. 29, n° 3, 2007.

[6] P. GENEVÈS, N. LAYAIDA, 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.

[7] R. GUERRAOUI, D. KOSTIC, R. R. LEVY, V. QUÉMA. *A High Throughput Atomic Storage Algorithm*, in "27th IEEE International Conference on Distributed Computing Systems (ICDCS 2007)", IEEE Computer Society, 2007.

[8] R. GUERRAOUI, R. R. LEVY, B. POCHON, V. QUÉMA. *High Throughput Total Order Broadcast for Cluster Environments*, in "International Conference on Dependable Systems and Networks (DSN 2006)", IEEE Computer Society, 2006.

[9] 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.

[10] A. SCHMITT, J.-B. STEFANI. *The M-calculus: A Higher-Order Distributed Process Calculus*, in "In Proceedings of the 30th Annual ACM Symposium on Principles of Programming Languages (POPL'03)", 2003.

[11] A. SCHMITT, J.-B. STEFANI. *The Kell Calculus: A Family of Higher-Order Distributed Process Calculi*, in "Global Computing", Lecture Notes in Computer Science, vol. 3267, Springer, 2005.

Year Publications

Doctoral Dissertations and Habilitation Theses

[12] J. KORNAS. *Contributions to software deployment in a component-based reflexive architecture*, Ph. D. Thesis, Université Joseph Fourier, Grenoble France, October 2008.

[13] J. POLAKOVIC. *Architecture logicielle et outils pour systèmes d'exploitation reconfigurables*, Ph. D. Thesis, Institut Polytechnique de Grenoble, Grenoble France, June 2008.

[14] C. TATON. *Vers l'auto-optimisation dans les systèmes autonomes*, Ph. D. Thesis, Institut Polytechnique de Grenoble, Grenoble France, November 2008.

Articles in International Peer-Reviewed Journal

[15] N. DE PALMA, S. BOUCHENAK, F. BOYER, D. HAGIMONT, S. SICARD, C. TATON. *Jade : Un Environnementd'Administration Autonome*, in "Techniques et Sciences Informatiques", to appear, 2008.

[16] J. POLAKOVIC, J.-B. STEFANI. *Architecting reconfigurable component-based operating systems*, in "Journal of Systems Architecture", vol. 54, n° 6, 2008.

[17] C. TATON, N. DE PALMA, S. BOUCHENAK, D. HAGIMONT. *Improving the Performance of JMS-Based Applications*, in "International Journal of Autonomic Computing", To appear, 2009.

[18] C. TATON, N. DE PALMA, J. PHILIPPE, S. BOUCHENAK. *Self-Optimization of Clustered Message-Oriented Middleware*, in "International Journal of Autonomic Computing", to appear, 2008.

International Peer-Reviewed Conference/Proceedings

[19] J. ARNAUD, S. BOUCHENAK. *Gestion de Ressources dans les Services Internet*, in "6e Conférence Française des Systèmes d'Exploitation (CFSE-6)", 2008.

[20] A. BOHANNON, J. N. FOSTER, B. C. PIERCE, A. PILKIEWICZ, A. SCHMITT. *Boomerang: resourceful lenses for string data*, in "35th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, (POPL)", 2008.

[21] L. BROTO, D. HAGIMONT, P. STOLF, N. DE PALMA, S. TEMATE. *Autonomic management policy specification in Tune*, in "ACM Symposium on Applied Computing (SAC)", 2008.

[22] N. DE PALMA. *Spécification de Politiques d'AdministrationAutonome avec Tune*, in "6e Conférence Française des Systèmes d'Exploitation (CFSE-6)", 2008.

[23] N. DE PALMA, K. POPOV, N. PARLAVANTZAS. *Tools for Architecture-Based Autonomic System*, in "5th International Conference on Autonomic and Autonomous Systems (ICAS 2009)", To appear, 2009.

[24] I. LANESE, J. A. PÉREZ, D. SANGIORGI, A. SCHMITT. *On the Expressiveness and Decidability of Higher-Order Process Calculi*, in "23rd Annual IEEE Symposium on Logic in Computer Science (LICS 2008)", 2008.

[25] M. LIENHARDT, A. SCHMITT, J.-B. STEFANI. *Typing Communicating Component Assemblages*, in "Proceedings of the 7th International Conference on Generative Programming and Component Engineering (GPCE'08)", 2008.

[26] M. PASIN, S. FONTAINE, S. BOUCHENAK. *Failure Detection in Large-Scale Distributed Systems: A Survey*, in "6th IEEE Workshop on End-to-End Monitoring Techniques and Services (E2EMon 2008)", 2008.

[27] P. V. ROY, S. HARIDI, A. REINEFELD, J.-B. STEFANI, R. H. C. YAP, T. COUPAYE. *Self Management for Large-Scale Distributed Systems: An Overview of the SELFMAN Project*, in "Formal Methods for Components and Objects, 6th International Symposium (FMCO 2007)", Lecture Notes in Computer Science, vol. 5382, Springer, 2008.

[28] 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.

[29] M. TOURE, G. BERHE, P. STOLF, L. BROTO, N. DE PALMA, D. HAGIMONT. *Autonomic Management for Grid Applications*, in "16th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP 2008)", IEEE Computer Society, 2008.

Scientific Books (or Scientific Book chapters)

[30] S. BOUCHENAK, N. DE PALMA. *Message Queuing Systems*, in "Encyclopedia of Database Systems", Springer Verlag, 2008.

[31] S. KRAKOWIAK. *Systèmes d'exploitation : principes et fonctions*, in "Technologies logicielles", Techniques de l'Ingénieur, 2008.

[32] C. TATON, N. DE PALMA, S. BOUCHENAK. *Adaptive Middleware for Message Queuing Systems*, in "Encyclopedia of Database Systems", Springer Verlag, 2008.

Research Reports

[33] J. ARNAUD, S. BOUCHENAK. *MoKa: A System for Modeling and Capacity Planning of Multi-Tier Systems*, Technical report, n^{o} RR-6730, INRIA, 2008, http://hal.inria.fr/inria-00340772/fr/.

[34] S. LENGLET, A. SCHMITT, J.-B. STEFANI. *Howe's Method for Early Bisimilarities*, Research Report, n^{o} RR-6773, INRIA, 2008, http://hal.inria.fr/inria-00347137/fr/.

[35] S. LENGLET, A. SCHMITT, J.-B. STEFANI. *Normal bisimulations in process calculi with passivation*, Research Report, n^{o} RR-6664, INRIA, 2008, http://hal.inria.fr/inria-00330565/fr/.

[36] L. MALRAIT, S. BOUCHENAK, N. MARCHAND. *Modeling and control of server Systems*, (non-public report), Technical report, INRIA, 2008.

[37] P. MERLE, J.-B. STEFANI. *A formal specification of the Fractal model in Alloy*, Research Report, n^{o} RR-6721, INRIA, 2008, http://hal.inria.fr/inria-00338987/fr/.

[38] C. TATON, S. BOUCHENAK, N. DE PALMA, D. HAGIMONT. *Self-Optimization of Internet Services with Dynamic Resource Provisioning*, Technical report, n^{o} RR-6575, INRIA, 2008, http://hal.inria.fr/inria-00294071/fr/.

Other Publications

[39] A. CORREIRA, A. NUNES, A. SOUSA, E. CECCHET, J. PERIERA, L. SOARES, L. RODRIGUES, M. MATOS, N. CARVALHO, R. OLIVEIRA, R. VALÇA, S. BOUCHENAK, S. SICARD. *GORDA Deliverable D5.2 – Prototype of the Integrated System*, GORDA European Project, 2008, http://gorda.di.uminho.pt/.

[40] A. CORREIRA, C. TATON, F. MÉTRAL, J. PERIERA, L. SOARES, M. MATOS, N. CARVALHO, R. VALÇA, R. OLIVEIRA, S. BOUCHENAK, S. SICARD. *GORDA Deliverable D5.4 – Management Tool Set*, GORDA European Project, 2008, http://gorda.di.uminho.pt/.

[41] L. DEMONTES. *Observation auto-adaptative de systèmes événementiels structurés en composants*, Masters thesis, Université Joseph Fourier, Grenoble, France, 2008, http://sardes.inrialpes.fr/papers/files/08-Demontes-MasterThesis.pdf.

[42] S. GENEVÈS. *Répartition adaptative dâévénements en contexte multiprocesseur,*, Masters thesis, Université Joseph Fourier, Grenoble, France, 2008, http://sardes.inrialpes.fr/papers/files/08-Geneves-MasterThesis.pdf.

[43] J. PERIERA, L. RODRIGUES, N. CARVALHO, R. OLIVEIRA, F. PEDONE, E. CECCHET, S. BOUCHENAK. *GORDA Deliverable D7.1 – Deployment Plan*, GORDA European Project, 2008, http://gorda.di.uminho.pt/.

References in notes

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

[45] G. LEAVENS, M. SITARAMAN (editors). *Foundations of Component-Based Systems*, Cambridge University Press, 2000.

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

[47] A. RAUSCH, R. REUSSNER, R. MIRANDOLA, F. PLASIL (editors). *The Common Component Modeling Example – Comparing Software Component Models*, Lecture Notes in Computer Science, vol. 5153, Springer, 2008.

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

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

[50] G. 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.

[51] 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.

[52] 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.

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

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

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

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

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

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

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

[60] 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.

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

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

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

[64] K. LAU, Z. WANG. *Software Component Models*, in "IEEE Trans. Software Eng.", vol. 33, n° 10, 2007.

[65] M. MCILROY. *Mass produced software components*, in "Software Engineering, Report on a conference sponsored by the NATO Science Committee, Garmisch, Germany, 7th to 11th October 1968", P. NAUR, B. RANDELL (editors), 1968.

[66] 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.

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

[68] 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.

[69] D. SANGIORGI. *Bisimulation for higher-order process calculi*, in "Information and Computation,Vol. 131, No 2", 1996.

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

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

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