INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Project-Team Sardes

System Architecture for Reflective Distributed Computing Environments

Grenoble - Rhône-Alpes

Theme : Distributed Systems and Services

Activity Report

2009

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 Mines]
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, HdR]
Noël De Palma [Associate Professor, Institut National Polytechnique de Grenoble, HdR]
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
Lionel Debroux
Fabien Mottet

PhD Student

Jean Arnaud [government grant]
Benoît Claudel [government grant]
Stéphane Fontaine [government grant]
Fabien Gaud [government grant]
Michaël Lienhardt [government grant]
Sergueï Lenglet [government grant]
Willy Malvaud [Inria grant (contract)]
Claudio Mezzina [Inria grant (contract), from Oct. 2008]
Alessio Pace [Inria grant, from Jan. 2009]
Sylvain Sicard [Inria grant (contract)]
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-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 covers: (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 an ADL or programming language 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** Workk in this area focuses on the exploitation and development of discrete and continuous control techniques for the construction of adaptive component-based system. Application domains considered for this theme are, respectively, embedded systems and performance management for application server clusters.
- **Virtual machine for component systems** Work in this area focuses on the development of a component-based virtual machine for embedded systems, with two main goals: (1) to develop an extended instruction set for component support, including support for dynamic configuration, orthogonal component persistence, and isolation; (2) to develop a native implementation of the virtual machine, on resource-constrained hardware.

3. Scientific Foundations

3.1. 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 its 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 relate to the component-based software engineering [111], and software architecture [109] 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 [55]. Several component models and associated architecture description languages have been devised over the past fifteen years: see e.g. [88] for an analysis of recent component models, and [93], [61] for surveys of architecture description languages.

To natively support configurability and adaptability in systems, SARDES component technology also draws from ideas in reflective languages [82], and reflective middleware [86], [59], [67]. Reflection can be used both to increase the separation of concerns in a system architecture, as pioneered by aspect-oriented programming [83], 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 [89], [50], 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 [95], or the analysis of object-orientation [51]. Relevant developments for SARDES endeavours include behavioral theory and coinductive proof techniques [105], [103], process calculi with localities [63], [65], [68], and higher-order variants of the π-calculus [104], [76].

3.3. Open programming

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

The rigorous study of programming features relate to the study of programming language constructs and semantics [97], [113], 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 [107], Alice [101], [102], ArchJava [52], Classages [91], Erlang [54], Oz [113], and Scala [98].
- For their security and failure management features: Acute, E [94], Erlang and Oz [66].
- For their support for concurrent and distributed execution, Acute, Alice, JoCaml [71], E, Erlang, Klaim [58], and Oz.

3.4. 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 [110] has a long history, with systems such as Oberon [114], SPIN [57] or JX [73]. More recent and relevant developments for SARDES are:

- The developments around the Singularity project at Microsoft Research [70], [79], which illustrates the use of language-based software isolation for building a secure operating system kernel.
- The seL4 project [74], [84], 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 [60] and Barrelfish [56].
- The development of efficient run-time for event-based programming on multicore systems such as libasync [115], [87].

3.5. System management and control

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

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

The above motivates a new approach, in which a significant part of management-related functions is performed automatically, with minimal human intervention. This is the goal of the so-called *autonomic computing* movement [80]. Several research projects [49] are active in this area. [81], [78] 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 [69], and techniques for constructing control algorithms in the decision stage of administration feedback loops, including discrete control techniques [64], and continuous ones [75].

Management and control functions built by SARDES require also the development of distributed algorithms [92], [112] at different scales, from algorithms for multiprocessor architectures [77] to algorithms for dynamic peer-to-peer computing systems [53], [100]. Of particular relevance in the latter contexts are epidemic protocols such as gossip protocols [108] because of their natural resilience to node dynamicity or *churn*, an inherent scalability.

4. Application Domains

4.1. Application Domains

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

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, which is accessible at http://www.ow2.org/.

We list below software which has been under development by SARDES in 2009. We do not list software produced by the SARDES project in the past years which has not been further developed by SARDES in 2009.

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

Contact : 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 themes tackled by SARDES: languages and foundations (6.2), system support (6.3), management and control (6.4), virtual machine (6.5).

6.2. Languages and foundations

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

6.2.1. Process algebra

The goal of this work is to study process algebraic foundations for component-based programming. Because of the inherently higher-order character of dynamic configuration operations (modelled e.g. by the passivation construct of the Kell calculus [106]), we are led to study new techniques for proving program equivalence in higher-order calculi, to develop new forms of bisimulation for characterizing barbed congruence, the natural form of program equivalence in process calculi [96], and to study the expressivity of different constructs in higher-order calculi.

We have refined results obtained last year on the definition of bisimulations for higher-order calculi with passivation. The study of normal bisimulations for such calculi was published at FOSSACS [29]. We have shown that even in a simple calculus with passivation, there is no direct equivalent to the notion of normal bisimulation developed by Sangiorgi for the higher-order π-calculus [104]. The definition of a new semantics for higher-order process calculi, called complementary semantics, resulting in the first characterization of weak contextual congruence for calculi with passivation, was presented at CONCUR [28]. The PhD dissertation of Sergueiï Lenglet, to be defended in January 2010, shows how to apply these techniques to the Kell calculus.

In parallel, we have continued the study of the expressivity of higher-order calculi. In particular, during the six month visit of Jorge A. Perez (a PhD student of Davide Sangiorgi), we have shown that it is impossible to find a compositional encoding of the bi-adic higher-order pi-calculus in the monadic higher-order pi calculus. We have also started to work on such expressiveness results in presence of passivation, and will continue to do so with Cinzia di Giusto who is starting an 18 months postdoc in January 2010.

Together with Daniel Hirschkoff from the Plume team at ENS Lyon, we have studied bisimilarity in a fragment of CCS that contains only prefix, parallel composition, synchronisation and a limited form of replication [48]. The characterisation is not an axiomatisation, but is instead presented as a rewriting system. Our method allows us to derive a new congruence result in the π-calculus: congruence holds in the sub-calculus that does not include restriction nor sum, and features a limited form of replication. This work has been submitted for publication.

6.2.2. Proof assistant

As part of a general work towards proof-assistant-based tools for verifying distributed systems and distributed abstract machines, we have implemented and proved correct a decision procedure for the equational theory of Kleene algebra [22]. This required a formalisation of finite automata theory algorithms, and raised several issues about writing efficient code in Coq. This work also raised an original question about the ability to add and remove types to various non commutative algebraic structures [40]. This work is the subject of the PhD thesis of Thomas Braibant.

6.2.3. Type systems

A new type system for the assemblage of communicating components has been defined, using a slightly simplified semantics for the routing of messages. This small modification allows us to have a decidable type inference, which is a strong improvement on our previous type system [90]. We have implemented the type inference algorithm and tested it with Fractal [62] and Click [85] configurations. This work has been presented at FORTE/FMOODS [30], and is part of the PhD thesis of Michaël Lienhardt.

In the setting of our collaboration with Nabil Layaïda and Pierre Genevès of the WAM project team, we have extended our previous work on an efficient static analysis tool for XML paths and types [72] to now include counting queries. This work has been submitted for publication.

6.3. System support

Participants: Renaud Lachaize, Vivien Quéma, Alan Schmitt, Jean-Bernard Stefani, Fabien Gaud, Sylvain Geneves, Willy Malvaud, Alessio Pace, Quentin Sabah, Valerio Schiavoni.

6.3.1. Support for event-based programming in multicore systems

Many high-performance communicating systems are designed using the event-driven paradigm. As multicore platforms are now pervasive, it becomes crucial for such systems to take advantage of the available hardware parallelism. Event-coloring is a promising approach in this regard. First, it allows programmers to simply and progressively inject support for the safe, parallel execution of multiple event handlers through the use of annotations. Second, it relies on a workstealing algorithm to dynamically balance the execution of event handlers on the available cores. We have studied the impact of the workstealing algorithm on the overall system performance. We have showed that the only existing workstealing algorithm designed for event-coloring runtimes is not always efficient: for instance, it causes a 33% performance degradation on a Web server. We have introduced several enhancements to improve the workstealing behavior. An evaluation using both microbenchmarks and real applications, a Web server and the Secure File Server (SFS), shows that our system consistently outperforms a state-of-the-art runtime (Libasync-smp), with or without workstealing. In particular, our new workstealing improves performance by up to +25% compared to Libasync-smp without workstealing and by up to +73% compared to the Libasync-smp workstealing algorithm, in the Web server case. This work is described in the research report [47].

6.3.2. Gossip-based protocols

Gossip-based information dissemination protocols are considered easy to deploy, scalable and resilient to network dynamics. They are also considered highly ßexible, namely tunable at will to increase their robustness and adapt to churn. So far however, they have mainly been evaluated through simulation, very often assuming ideal settings. In collaboration with Rachid Guerroui's group at EPFL and the INRIA ASAP team, we have developed variants of gossip protocols better suited to real-world settings. We have obtained results in two directions:

- Heterogeneity: we have proposed a new gossip protocol, called HEAP, 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. This work has been presented at the DSN and Middleware conferences [26], [25].
- NAT resilience: gossip peer sampling protocols typically ignore the fact that in a real wide-area setting a significant proportion of peers sit beind Network Address Translation (NAT) devices, preenting direct communication without specific NAT-traversal mechanisms such as hole-punching. 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. This work has been presented at the ICDCS conference [27].

6.4. Management and control

Participants: Sara Bouchenak, Fabienne Boyer, Noël De Palma, Olivier Gruber, Eric Rutten, Jean-Bernard Stefani, Gwenael Delaval, Soufyane Aboubekr, Jean Arnaud, Loris Bouzonnet, Jérémy Philippe.

6.4.1. Self-tuning for Internet Services

This work aims at building SLA (Service Level Agreement) management and self-tuning functions for Web application servers. In particular, we address the issue of providing guarantees on both service performance and service availability, two criteria of service quality that are usually handled separetely, and can be seen as antagonistic (high service availability is usually obtained at the expense of performance, and good performance has usually an impact on availability). We have developed two sets of techniques to achieve self-tuning with guarantees on service level objectives (SLOs) such as maximum response times and maximum abandon rates.

The first set of techniques applies fluid models from control theory to modeling, capacity planning, system control and provisioning of server systems. Our main contributions are:

1. a novel model that reflects the non-linear behavior of server systems,
2. a dynamically and automatically evolving model state based on online monitoring in order to reflect service workload variation,
3. novel control laws that take into account both performance and availability SLOs to dynamically and automatically control the system and provision its resources accordingly.

These techniques were proposed in the form of theoretical model and control laws, implemented in a Java-based prototype called *ConSer*, and successfully evaluated in a real warehouse online service running on a database system [31], [32], [33], [34]. This work is conducted in the context of a collaboration between SARDES and the NeCS INRIA Project-Team. This work is the subject of the PhD thesis of Luc Malrait, who is co-advised by Nicolas Marchand (NeCS) and Sara Bouchenak (SARDES).

The second set of techniques makes use of queuing theory to model, provision and plan the capacity of Internet services deployed on clusters of computers, as is usually the case of e-commerce services. Our main contributions consist of:

1. an extended queuing model that takes into account the distribution and parallelism of cluster-based distributed systems, and allows to predict system performance and availability,
2. a novel approach for dynamically and automatically configuring model state, which reflects workload changes and does not require system administrators to perform offline calibration of the model, a technically tricky phase usually necessary prior to the use of these types models,
3. a novel control algorithm that takes into account both performance and availability SLOs while minimizing system costs; it applies dynamic and automatic configuration and provisioning of cluster-based systems with necessary and sufficient resources that guarantee target service performance and availability.

These techniques were proposed in the form of a theoretical extension of the well-known MVA queuing model, the specification of a capacity planning and provisioning algorithm, the design of an online distributed monitoring mechanism of cluster-based systems, and the implementation of a Java-based software prototype called *MoKa* running in realistic distributed Web applications running on Web servers and database servers [21]. This work is the subject of the Ph.D. thesis of J. Arnaud.

6.4.2. Self-repair for cluster systems

We have completed the implementation and evaluation of a self-repair architecture for cluster systems, built with the JADE framework. Self-repair is achieved through a combination of component-based design, reßection and active replica- tion of the management subsystem. The self-repair architecture has been applied to a JEE Web application server cluster, and an NFS file server cluster. Our evaluations show that the JADE framework and the self-repair mechanisms add negligible overhead to the operation of a managed system, and that our self-repair architecture achieves short MTTR (Mean Time To Repair) even with a simple repair policy consisting of reconfiguring the managed system so as to maintain the same Web application server architecture in the running cluster. The self-repair architecture is described in detail in Sylvain Sicard's PhD thesis [16], and in the book chapter [44].

6.4.3. Discrete control for adaptive and reconfigurable systems

Our approach is to apply control techniques based on the behavioral model of reactive automata and the algorithmic techniques of discrete controller synthesis. We consider the problem theoretically and also practically. We therefore adopt the synchronous approach to reactive systems: this way we can, on the one hand, rely on synchronous languages as modeling formalisms, which are equipped with compilers and executable code generators, and on the other hand, use an associated effective controller synthesis tool: Sigali. Both are integrated into a programming language, called BZR, and its compiler, as an extension of the Heptagon language [20], [46]. We thus have a complete tool-supported method from control modeling down to concrete execution, considering different execution models, and targetting either software or hardware.

This gives us a multi-tier approach to the model-based control of adaptive computing systems, where we can consider the general problem, and contribute on a more specialized topic: to model the discrete behaviours of the system to be regulated, and to obtain a correct controller by automated synthesis. The originality of our approach is in the emerging nature of this topic in computer science, which is appearing in soft-, hard- and middleware. It represents a new combination of existing approaches in:

- operating systems and middleware
- (discrete) control theory
- formal methods and tools, as a link between the two.

We explore control theory for computer science, as an original alernative to computer science for control (as more usually in embedded systems), and to classical discrete control theory (as more usually applied to manufacturing) [37].

We currently have the following activities and contacts on the different aspects of our approach:

- Control techniques are explored, in discrete controller synthesis (DCS) in cooperation with Herve Marchand (VerTecs, INRIA Rennes), and in optimal DCS on paths, in cooperation with VerTecs (INRIA Rennes), Pop Art (INRIA Grenoble) and INSA Lyon. Other DCS techniques for distributed control are considered woth contacts with GIPSA-Lab and ENSI Tunis.
- At the language-level, a modelling and controller generation language called BZR, which involves DCS in its compilation, is designed and develloped with VerTecs (INRIA Rennes); this is done in relation with other synchronous languages developed in the Synchronics Inria cooperative project. An ongoing activity is the language-level integration in a formal specification of Fractal in Alloy, and with the behavioral ADL in Fractal, in the framework of the Minalogic project MIND.
- At the execution level, modelling of mechanisms and execution of adaptation control is explored in terms of reactive models of adaptation, and linking the generated controller in the adaptative middleware, related to Fractal: in the framework of the Minalogic project MIND, we target the C implementation (related to Think/Cecilia) in cooperation with ST Microelectronics and Orange labs. We consider execution models with sequential code generation (C, Java, Caml), and begin to explore distributed control.
- We are exploring other target application domains, where we expect to find commonalities in the control problems, and variations in the definitions of configurations, and in the criteria motivating adaptation. We are linking BZR with Orccad, a programming environment for control systems, in cooperation with NeCS and SED at INRIA Grenoble. We are examining administration loops in a virtual machine, in cooperation with the Synergy group of Sardes. We are starting an activity concerning reconfigurable circuits, particularly FPGAs, in the framework of the ANR project FAMOUS, in cooperation with DaRT (INRIA Lille), and Lab-STICC in Lorient.

6.5. Virtual machine

Participants: Fabienne Boyer, Noël De Palma, Olivier Gruber, Jean-Bernard Stefani, Benoit Claudel, Ludovic Demontes, Stéphane Fontaine, Jérémy Philippe.

As part of a more global goal of constructing a reflective virtual machine for component systems, we have studied issues of memory safety and isolation between concurrent components. This work is reported in the PhD thesis of Benoît Claudel [13]. Our approach comprises:

- An asynchronous concurrent component model, where components logically share no information but are allowed to exchange arbitrary object graphs though messages.
- A modified Java-virtual machine that implements a proxy mechanism for transferrable objects and the necessary checks to keep track of object ownership during execution (a mutable object is owned by at most one component during execution).

This combination of features is original compared to the current state of the art exemplified e.g. by the Singularity operating system [70], where messages exchanged between isolation units (called processes in Singularity) are restricted to have a tree structure. A simpler and potentially more efficient virtual machine design for the same asynchronous model, which relies on simple read barriers and a combination of alias and control flow analysis, has also been devised and is currently being implemented.

7. Contracts and Grants with Industry

7.1. National Actions

7.1.1. ASR Network

SARDES is a member of the CNRS research network GDR 725 ASR "Architecture, Système et Réseau". See http://asr.cnrs.fr/.

7.1.2. 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 has run from January 2007 to December 2009.

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

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

7.1.4. Project MIND (ANR-Minalogic)

Participants: Eric Rutten, Jean-Bernard Stefani, Gwenael Delaval, Soufyane Aboubekr.

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

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

The project partners include STMicroelectronics, CEA, INRIA (Adam and Sardes prokect teams), Schneider. The project runs from October 2008 to October 2010.

8. Other Grants and Activities

8.1. EC-funded projects

8.1.1. 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 has run from June 1, 2006 to September 30, 2009.

8.1.2. 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 has run from January 2006 to May 2009.

8.1.3. IST Project OMP

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

Participants : Renaud Lachaize, Vivien Quéma, 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 has run from January 2008 to December 2009.

8.2. OW2 Consortium

Participants: Jean-Bernard Stefani, Valerio Schiavoni.

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. 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. This collaboration has been supported in 2009 by the Bacon “Equipe associée”. Contact persons in SARDES: D. Pous, 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 was a member of the Program Committee of the ACES-MB 2009 (Model Based Architecting and Construction of Embedded Systems) and CBHPC 2009 (Component-Based High-Performance Computing) workshops.
- J.B. Stefani was the PC Chair and organizer of the workshop "Architeture and Languages for Self-Managed Systems", held in conjunction with the SASO 2009 conference.
- J.B. Stefani is a member of the OW2 Technical Council, and a member of the Technology Council of STMicroelectronics.
- Eric Rutten was a member of the Program Committee of the Workshop on Adaptive and Reconfigurable Embedded Systems (APRES) 2009.
- Eric Rutten and Sara Bouchenak are the chairs of the 5th International Workshop on Feedback Control Implementation and Design in Computing Systems and Networks (FeBID 2010), to be held in conjunction with the EuroSys 2010 conference.
- A. Schmitt was a member of the program committee of the Types Workshop 2009.
- A. Schmitt was the General Chair and Program Committee chair of the Journées Francophones des Langages Applicatifs (JFLA 2009).
- Vivien Quéma organized the 2nd Winter School on Hot Topics in Distributed Computing, co-sponsored by INRIA and Eurosys.
- S. Bouchenak is an officer of ACM SIGOPS France.
- S. Bouchenak is a member of CES INRIA - Grenoble (Comité des Emplois Scientifiques).
- S. Bouchenak was a member of CSE at University of Grenoble I, and Institut Polytechnique de Grenoble (Commission de Spécialistes d'Etablissement).
- S. Bouchenak was a member of the program committees of 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^{o} 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^{o} 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] F. BOYER. *Gestion de l'adaptabilité dans les applications réparties*, Université Joseph Fourier, Grenoble, December 2009, Habilitation Ã Diriger des Recherches.

[13] B. CLAUDEL. *Mécanismes logiciels de protection mémoire*, Institut Polytechnique de Grenoble, Grenoble France, December 2009, Ph. D. Thesis.

[14] N. DE PALMA. *De l'Administration vers l'Administration Autonome*, Institut National Polytechnique de Grenoble, June 2009, Habilitation Ã Diriger des Recherches.

[15] J. PHILIPPE. *Auto-Adaptation du Niveau de Service dans les Systèmes Distribués*, Institut Polytechnique de Grenoble, Grenoble France, January 2009, Ph. D. Thesis.

[16] S. SICARD. *Vers l'autoréparation dans les systèmes répartis*, Université Joseph Fourier, Grenoble, Grenoble France, March 2009, Ph. D. Thesis.

Articles in International Peer-Reviewed Journal

[17] A. GAMATIÉ, E. RUTTEN, H. YU, P. BOULET, JEAN-LUC. DEKEYSER. *Model-driven Engineering and Formal Validation of High-Performance Embedded Systems*, in "SCPE electronic journal (Scalable Computing: Practice and Experience)", vol. 10, n^o^ 2, June 2009, http://www.scpe.org/?a=qs&limit=6&offset=0&search=Model-driven+Engineering.

[18] A. GIRAULT, E. RUTTEN. *Automating the Addition of Fault Tolerance with Discrete Controller Synthesis*, in "Int. journal on Formal Methods in System Design", vol. 35, n^o^ 2, October 2009.

[19] C. TATON, N. DE PALMA, S. BOUCHENAK, D. HAGIMONT. *Improving the performance of JMS-based applications*, in "International Journal of Autonomic Computing (IJAC)", vol. 1, n^o^ 1, 2009.

International Peer-Reviewed Conference/Proceedings

[20] S. ABOUBEKR, G. DELAVAL, E. RUTTEN. *A Programming Language for Adaptation Control: Case Study*, in "2nd Workshop on Adaptive and Reconfigurable Embedded Systems, APRES 2009 (in conjunction with ESWeek 2009)", 2009.

[21] J. ARNAUD, S. BOUCHENAK. *Modélisation et planification de capacités dans les systèmes multi-étagés*, in "9eme Conférence Internationale sur Les Nouvelles Technologies de la Répartition (NOTERE 2009)", 2009.

[22] T. BRAIBANT, D. POUS. *A Tactic for Deciding Kleene Algebras*, in "Proceedings of the 1st Coq Workshop", August 2009, http://hal.archives-ouvertes.fr/hal-00383070/fr/.

[23] B. CLAUDEL, G. HUARD, O. RICHARD. *TakTuk, Adaptive Deployment of Remote Executions*, in "Proceedings of the International Symposium on High Performance Distributed Computing (HPDC)", jun 2009.

[24] N. DE PALMA, N. PARLAVANZAS, K. POPOV, P. BRAND, V. VLASSOV. *Tools for Autonomic Computing*, in "5th International Conference on Autonomic and Autonomous Systems (ICAS 2009)", IEEE Computer Society, 2009.

[25] D. FREY, R. GUERRAOUI, A.-M. KERMARREC, B. KOLDEHOFE, M. MOGENSEN, M. MONOD, V. QUÉMA. *Heterogeneous Gossip*, in "10th ACM/IFIP/USENIX International Middleware Conference (Middleware)", 2009.

[26] D. FREY, R. GUERRAOUI, A.-M. KERMARREC, M. MONOD, V. QUÉMA. *Stretching Gossip with Live Streaming*, in "39th IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)", 2009.

[27] A.-M. KERMARREC, A. PACE, V. QUÉMA, V. SCHIAVONI. *NAT-resilient Gossip Peer Sampling*, in "29th International Conference on Distributed Computing Systems (ICDCS)", 2009.

[28] S. LENGLET, A. SCHMITT, J.-B. STEFANI. *Howe's method for calculi with passivation*, in "20th International Conference on Concurrency Theory (CONCUR 2009)", Lecture Notes in Computer Science, Springer, 2009.

[29] S. LENGLET, A. SCHMITT, J.-B. STEFANI. *Normal bisimulations in process calculi with passivation*, in "12th International Conference on Foundations of Software Science and Computational Structures (FOSSACS 2009)", Lecture Notes in Computer Science, vol. 5504, Springer, 2009.

[30] M. LIENHARDT, C. A. MEZZINA, A. SCHMITT, J.-B. STEFANI. *Typing Component-Based Communication Systems*, in "11th Formal Methods for Open Object-Based Distributed Systems (FMOODS) and 29th Formal Techniques for Networked and Distributed Systems (FORTE)", Lecture Notes in Computer Science, Springer, 2009.

[31] 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.

[32] L. MALRAIT, S. BOUCHENAK, N. MARCHAND. *Modeling and control of server systems: application to database systems*, in "10th European Control Conference (ECC 09)", 2009.

[33] L. MALRAIT, N. MARCHAND, S. BOUCHENAK. *Average Delay Guaranty in Server Systems Using Admission Control*, in "8th IFAC Workshop on Time-Delay Systems", September 2009.

[34] L. MALRAIT, N. MARCHAND, S. BOUCHENAK. *Modélisation et contrôle d'un serveur*, in "French Chapter of the ACM-SIGOPS, 7ème Conférence Française sur les Systèmes d'Exploitation (CFSE 2009)", September 2009.

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

[36] J. PHILIPPE, N. DE PALMA, O. GRUBER, F. BOYER. *Self-Adapting Service Level in Java Enterprise Edition*, in "10th ACM/IFIP/USENIX International Middleware Conference (Middleware)", 2009.

[37] E. RUTTEN. *Supervisory control of adaptive and reconfigurable computing systems*, in "13th IFAC Symposium on Information Control Problems in Manufacturing, INCOM'09", 2009.

[38] L. SEINTURIER, P. MERLE, D. FOURNIER, N. DOLET, V. SCHIAVONI, J.-B. STEFANI. *Reconfigurable SCA Applications with the FraSCAti Platform*, in "6th IEEE International Conference on Services Computing (SCC)", IEEE, 2009.

[39] H. YU, G. DELAVAL, A. GAMATIÉ, E. RUTTEN. *A Case Study on Controller Synthesis for Data-Intensive Embedded Systems*, in "6th IEEE International Conference on Embedded Software and Systems, ICESS'09", 2009.

Workshops without Proceedings

[40] D. POUS. *Untyping Typed Algebraic Structures*, in "Proceedings of the CAM-CAD Workshop", October 2009, http://hal.archives-ouvertes.fr/hal-00421158/fr/.

Scientific Books (or Scientific Book chapters)

[41] A. Girault, E. Rutten (editors). *Workshop on Model-driven High-level Programming of Embedded Systems, SLA++P'08*, Electronic Notes in Theoretical Computer Science (ENTCS), vol. 238, n° 1, Elsevier Science, june 2009.

[42] F. Maraninchi, M. Mendler, M. Pouzet, A. Girault, E. Rutten (editors). *Model-driven High-level Programming of Embedded Systems. Selected papers from Sla++p'07 and Sla++p'08*, Special issue of the Eurasip Journal on Embedded Systems, April 2009.

[43] E. Niel, L. Pietrac, J. Muller, E. Rutten (editors). *Interactions of discrete-event automation and computer sciences*, Special section of the Journal of Universal Computer Science, July 2009.

[44] F. Boyer, N. De Palma, O. Gruber, S. Sicard, J.-B. Stefani. *A self-repair architecture for cluster systems*, in "Architecting Dependable Systems 6", R. de Lemos, J.-C. Fabre, C. Gacek, F. Gadducci, M. H. ter Beek (editors), Lecture Notes in Computer Science, vol. 5835, Springer, 2009.

Books or Proceedings Editing

[45] G. S. Blair, T. Coupaye, J.-B. Stefani (editors). *Component-based architecture: the Fractal initiative*, Annales des Télécommunications, vol. 64, n° 1-2, 2009.

Research Reports

[46] Gwenaël. Delaval, Hervé. Marchand, E. Rutten. *BZR Contracts for Modular Discrete Controller Synthesis*, n° 7111, INRIA, November 2009, http://hal.inria.fr/inria-00436560/, Rapport de Recherche.

[47] F. Gaud, S. Geneves, R. Lachaize, B. Lepers, F. Mottet, G. Muller, V. Quéma. *Mely: Efficient Workstealing for Multicore Event-Driven Systems*, n° RR-7169, INRIA, 2010, http://hal.inria.fr/inria-00449530/fr/, Research Report.

Other Publications

[48] D. Hirschkoff, D. Pous. *On characterising strong bisimilarity in a fragment of CCS with replication*, 2009, http://hal.archives-ouvertes.fr/hal-00329468/fr/, (Available as a HAL report).

References in notes

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

[50] Z. Liu, H. Jifeng (editors). *Mathematical Frameworks for Component Software - Models for Analysis and Synthesis*, World Scientific, 2006.

[51] M. Abadi, L. Cardelli. *A theory of objects*, Springer, 1996.

[52] J. Aldrich, C. Chambers, D. Notkin. *Architectural Reasoning in ArchJava*, in "Proceedings 16th European Conference on Object-Oriented Programming (ECOOP)", 2002.

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

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

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

[56] 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.

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

[58] 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, n^{o} 2874, Springer-Verlag, 2003.

[59] 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.

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

[61] 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.

[62] 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", vol. 36, n^{o} 11-12, 2006.

[63] 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.

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

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

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

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

[68] 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.

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

[70] 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.

[71] 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, vol. 2638, 2003.

[72] P. GENEVÈS, N. LAYAIDA, A. SCHMITT. *Efficient static analysis of XML paths and types*, in "Proceedings of the ACM SIGPLAN 2007 Conference on Programming Language Design and Implementation (PLDI)", ACM, 2007.

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

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

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

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

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

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

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

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

[81] 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.

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

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

[84] 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.

[85] E. KOHLER, R. MORRIS, B. CHEN, J. JANNOTTI, M. F. KAASHOEK. *The Click Modular Router*, in "ACM Trans. Comput. Syst.", vol. 18, n^{o} 3, 2000.

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

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

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

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

[90] 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.

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

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

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

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

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

[96] R. MILNER, D. SANGIORGI. *Barbed Bisimulation*, in "Automata, Languages and Programming, 19th International Colloquium, ICALP92", Lecture Notes in Computer Science, vol. 623, Springer, 1992.

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

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

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

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

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

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

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

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

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

[106] 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.

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

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

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

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

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

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

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

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

[115] 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.