IN PARTNERSHIP WITH:
CNRS

Université Nantes

Ecole des Mines de Nantes

Activity Report 2013

Project-Team ASCOLA

Aspect and composition languages

IN COLLABORATION WITH: Laboratoire d'Informatique de Nantes Atlantique (LINA)

RESEARCH CENTER
Rennes - Bretagne-Atlantique

THEME
Distributed programming and Software engineering

Table of contents

Project-Team ASCOLA

Keywords: Software Composition, Aspect-oriented Programming, Programming Languages, Distributed Systems, Cloud Computing, Formal Methods, Energy Consumption, Security

Creation of the Project-Team: 2009 January 01.

1. Members

Research Scientists

Adrien Lèbre [Inria, Researcher, on leave from Mines Nantes since Sep 2013]
Nicolas Tabareau [Inria, Researcher]

Faculty Members

Mario Südholt [Team leader, Mines Nantes, Professor, HdR]
Pierre Cointe [Mines Nantes, Professor, HdR]
Rémi Douence [Mines Nantes, Associate Professor]
Hervé Grall [Mines Nantes, Associate Professor]
Thomas Ledoux [Mines Nantes, Associate Professor]
Jean-Marc Menaud [Mines Nantes, Associate Professor, HdR]
Julien Mercadal [Mines Nantes, Associate Professor, temporary position, until Aug 2013]
Jacques Noyé [Mines Nantes, Associate Professor]
Flavien Quesnel [Mines Nantes, Associate Professor, temporary position, since Oct 2013]
Jean-Claude Royer [Mines Nantes, Professor, HdR]

PhD Students

Gustavo Bervian Brand [ARMINES, until Dec 2013]
Paul Blouët [VU Brussel & Mines Nantes, co-supervision, since Nov 2013]
Frederico Alvares de Oliveira Junior [U. Nantes, until Apr 2013]
Akram Ajouli [Mines Nantes, until Sep 2013]
Diana Allam [Univ. Nantes, until Sep 2013]
Walid Benghabrit [ARMINES, since Oct 2013]
Ronan-Alexandre Cherrueau [ARMINES]
Frédéric Dumont [ARMINES, CIFRE]
Simon Dupont [Mines Nantes, CIFRE]
Ismael Figueroa [U. Chile, co-supervision]
Alexandre Garnier [ARMINES, since Oct 2013]
Sabbir Hasan [INSA Rennes, since Nov 2013]
Guilhem Jaber [ENS Cachan]
Guillaume Le Louët [ARMINES]
Florent Marchand de Kerchove de Denterghem [Mines Nantes]
Jonathan Pastor [Mines Nantes]
Charles Prud'Homme [Mines Nantes, co-supervision with the TASC team]
Kevin Quirin [ENS Cachan, since Jun 2013]
Jurgen Van Ham [TU Darmstadt & Mines Nantes, co-supervision]

Post-Doctoral Fellows

Thierry Bernard [ARMINES]
Omar Chebaro [Inria, until May 2013]
Takahiro Hirofuchi [Mines Nantes, since Jan 2013 until Dec 2013]
Yousri Kouki [ARMINES, since Dec 2013]
Frederico Alvares [Mines Nantes, until Oct 2013]
Hemant Kumar Mehta [Inria, since Feb 2013]

Mohamed Sellami [ARMINES, since Sep 2013]

Administrative Assistants

Cecile Derouet [Inria, part-time 33%]
Florence Rogues [Mines Nantes, part-time 25%]

2. Overall Objectives

2.1. Presentation

The ASCOLA project-team aims at developing and harnessing advanced application structuring mechanisms, and supporting the transformational development of correct large-scale applications and their evolution throughout their entire life cycle. We apply a language-based approach to achieve this goal, defining new languages in order to represent architectural as well as implementation level abstractions and exploit formal methods to ensure correctness.

Concretely, we investigate expressive aspect languages to modularize crosscutting concerns. Those languages enable sophisticated relationships between execution events to be formulated and manipulated directly at the language level. We study how to reconcile invasive accesses by aspects with strongly encapsulated software entities. Furthermore, we foster the transformational development of implementations from higher-level architectural software representations using domain-specific languages. Finally, we focus on abstractions and development methods for distributed and concurrent systems, in particular flexible, secure and energy-efficient applications and infrastructures for the Cloud.

Our results are subjected to validation in the context of four main application domains: enterprise information systems, service-oriented architectures, cloud applications, and pervasive systems.

2.2. Highlights of the Year

This year we have published two groups of major results. In the domain of efficient resource management in the Cloud, we have proposed an elastic consolidation service, new techniques for enforcing SLA guarantees, and new simulation methods for Cloud environments. These results have all been published in major conferences of the domain: IEEE Cloud, IEEE/ACM CCGrid and IEEE CloudCom, see Sec. 6.3 for more information.

The team has also presented several major results on the foundations of software composition. These include new programming language mechanisms for the correct definition of aspects and techniques for their semantic, in particular type-based, definition. These results have also been published through major dissemination channels: Elsevier's journal Science of Computer Programming and the international conference Modularity.aosd, see Sec. 6.2 for more information.

Moreover, Ismael Figueroa Palet, a PhD student co-supervised by Nicolas Tabareau and Prof. Éric Tanter from University of Chile, has finished first place at the ACM Student Research Competition at Modularity:aosd'13 with the work "Taming Aspects with Monads and Membranes."

Finally, Adrien Lèbre has been appointed in September on an Inria research position for three years.

3. Research Program

3.1. Overview

Since we mainly work on new software structuring concepts and programming language design, we first briefly introduce some basic notions and problems of software components (understood in a broad sense, i.e., including modules, objects, architecture description languages and services), aspects, and domain-specific languages. We conclude by presenting the main issues related to distribution and concurrency that are relevant to our work.

3.2. Software Composition

Modules and services. The idea that building *software components*, i.e., composable prefabricated and parameterized software parts, was key to create an effective software industry was realized very early [86]. At that time, the scope of a component was limited to a single procedure. In the seventies, the growing complexity of software made it necessary to consider a new level of structuring and programming and led to the notions of information hiding, *modules*, and module interconnection languages [93], [71]. Information hiding promotes a black-box model of program development whereby a module implementation, basically a collection of procedures, is strongly encapsulated behind an interface. This makes it possible to guarantee logical invariant *properties* of the data managed by the procedures and, more generally, makes *modular reasoning* possible.

In the context of today's Internet-based information society, components and modules have given rise to *software services* whose compositions are governed by explicit *orchestration or choreography* specifications that support notions of global properties of a service-oriented architecture. These horizontal compositions have, however, to be frequently adapted dynamically. Dynamic adaptations, in particular in the context of software evolution processes, often conflict with a black-box composition model either because of the need for invasive modifications, for instance, in order to optimize resource utilization or modifications to the vertical compositions implementing the high-level services.

Object-Oriented Programming.*Classes* and *objects* provide another kind of software component, which makes it necessary to distinguish between *component types* (classes) and *component instances* (objects). Indeed, unlike modules, objects can be created dynamically. Although it is also possible to talk about classes in terms of interfaces and implementations, the encapsulation provided by classes is not as strong as the one provided by modules. This is because, through the use of inheritance, object-oriented languages put the emphasis on *incremental programming* to the detriment of modular programming. This introduces a white-box model of software development and more flexibility is traded for safety as demonstrated by the *fragile base class* issue [89].

Architecture Description Languages. The advent of distributed applications made it necessary to consider more sophisticated connections between the various building blocks of a system. The *software architecture* [97] of a software system describes the system as a composition of *components* and *connectors*, where the connectors capture the *interaction protocols* between the components [62]. It also describes the rationale behind such a given architecture, linking the properties required from the system to its implementation. *Architecture Description Languages* (ADLs) are languages that support architecture-based development [87]. A number of these languages make it possible to generate executable systems from architectural descriptions, provided implementations for the primitive components are available. However, guaranteeing that the implementation conforms to the architecture is an issue.

Protocols. Today, protocols constitute a frequently used means to precisely define, implement, and analyze contracts between two or more hardware or software entities. They have been used to define interactions between communication layers, security properties of distributed communications, interactions between objects and components, and business processes.

Object interactions [91], component interactions [103], [95] and service orchestrations [72] are most frequently expressed in terms of *regular interaction protocols* that enable basic properties, such as compatibility, substitutability, and deadlocks between components to be defined in terms of basic operations and closure properties of finite-state automata. Furthermore, such properties may be analyzed automatically using, e.g., model checking techniques [69], [78].

However, the limited expressive power of regular languages has led to a number of approaches using more expressive *non-regular* interaction protocols that often provide distribution-specific abstractions, e.g., session types [80], or context-free or turing-complete expressiveness [96], [67]. While these protocol types allow conformance between components to be defined (e.g., using unbounded counters), property verification can only be performed manually or semi-automatically.

3.3. Programming languages for advanced modularization

The main driving force for the structuring means, such as components and modules, is the quest for clean *separation of concerns* [73] on the architectural and programming levels. It has, however, early been noted that concern separation in the presence of crosscutting functionalities requires specific language and implementation level support. Techniques of so-called *computational reflection*, for instance, Smith's 3-Lisp or Kiczales's CLOS meta-object protocol [98], [83] as well as metaprogramming techniques have been developed to cope with this problem but proven unwieldy to use and not amenable to formalization and property analysis due to their generality. Methods and techniques from two fields have been particularly useful in addressing such advanced modularization problems: Aspect-Oriented Software Development as the field concerned with the systematic handling of modularization issues and domain-specific languages that provide declarative and efficient means for the definition of crosscutting functionalities.

Aspect-Oriented Software Development [82], [60] has emerged over the previous decade as the domain of systematic exploration of crosscutting concerns and corresponding support throughout the software development process. The corresponding research efforts have resulted, in particular, in the recognition of *crosscutting* as a fundamental problem of virtually any large-scale application, and the definition and implementation of a large number of aspect-oriented models and languages.

However, most current aspect-oriented models, notably AspectJ [81], rely on pointcuts and advice defined in terms of individual execution events. These models are subject to serious limitations concerning the modularization of crosscutting functionalities in distributed applications, the integration of aspects with other modularization mechanisms such as components, and the provision of correctness guarantees of the resulting AO applications. They do, in particular, only permit the manipulation of distributed applications on a per-host basis, that is, without direct expression of coordination properties relating different distributed entities [99]. Similarly, current approaches for the integration of aspects and (distributed) components do not directly express interaction properties between sets of components but rather seemingly unrelated modifications to individual components [70]. Finally, current formalizations of such aspect models are formulated in terms of low-level semantic abstractions (see, e.g., Wand's et al semantics for AspectJ [102]) and provide only limited support for the analysis of fundamental aspect properties.

Recently, first approaches have been put forward to tackle these problems, in particular, in the context of so-called *stateful* or *history-based aspect languages* [74], [75], which provide pointcut and advice languages that directly express rich relationships between execution events. Such languages have been proposed to directly express coordination and synchronization issues of distributed and concurrent applications [92], [65], [77], provide more concise formal semantics for aspects and enable analysis of their properties [63], [76], [74], [61]. Furthermore, first approaches for the definition of *aspects over protocols* have been proposed, as well as over regular structures [74] and non-regular ones [101], [90], which are helpful for the modular definition and verification of protocols over crosscutting functionalities.

Due to the novelty of these approaches, they represent, however, only first results and many important questions concerning these fundamental issues remain open.

Domain-specific languages (DSLs) represent domain knowledge in terms of suitable basic language constructs and their compositions at the language level. By trading generality for abstraction, they enable complex relationships among domain concepts to be expressed concisely and their properties to be expressed and formally analyzed. DSLs have been applied to a large number of domains; they have been particularly popular in the domain of software generation and maintenance [88], [104].

Many modularization techniques and tasks can be naturally expressed by DSLs that are either specialized with respect to the type of modularization constructs, such as a specific brand of software component, or to the compositions that are admissible in the context of an application domain that is targeted by a modular implementation. Moreover, software development and evolution processes can frequently be expressed by transformations between applications implemented using different DSLs that represent an implementation at different abstraction levels or different parts of one application.

Functionalities that crosscut a component-based application, however, complicate such a DSL-based transformational software development process. Since such functionalities belong to another domain than that captured by the components, different DSLs should be composed. Such compositions (including their syntactic expression, semantics and property analysis) have only very partially been explored until now. Furthermore, restricted composition languages and many aspect languages that only match execution events of a specific domain (e.g., specific file accesses in the case of security functionality) and trigger only domain-specific actions clearly are quite similar to DSLs but remain to be explored.

3.4. Distribution and Concurrency

While ASCOLA does not investigate distribution and concurrency as research domains per se (but rather from a software engineering and modularization viewpoint), there are several specific problems and corresponding approaches in these domains that are directly related to its core interests that include the structuring and modularization of large-scale distributed infrastructures and applications. These problems include crosscutting functionalities of distributed and concurrent systems, support for the evolution of distributed software systems, and correctness guarantees for the resulting software systems.

Underlying our interest in these domains is the well-known observation that large-scale distributed applications are subject to *numerous crosscutting functionalities* (such as the transactional behavior in enterprise information systems, the implementation of security policies, and fault recovery strategies). These functionalities are typically partially encapsulated in distributed infrastructures and partially handled in an ad hoc manner by using infrastructure services at the application level. Support for a more principled approach to the development and evolution of distributed software systems in the presence of crosscutting functionalities has been investigated in the field of *open adaptable middleware* [66], [85]. Open middleware design exploits the concept of reflection to provide the desired level of configurability and openness. However, these approaches are subject to several fundamental problems. One important problem is their insufficient, framework-based support that only allows partial modularization of crosscutting functionalities.

There has been some *criticism* on the use of *AspectJ-like aspect models* (which middleware aspect models like that of JBoss AOP are an instance of) for the modularization of distribution and concurrency related concerns, in particular, for transaction concerns [84] and the modularization of the distribution concern itself [99]. Both criticisms are essentially grounded in AspectJ's inability to explicitly represent sophisticated relationships between execution events in a distributed system: such aspects therefore cannot capture the semantic relationships that are essential for the corresponding concerns. History-based aspects, as those proposed by the ASCOLA project-team provide a starting point that is not subject to this problem.

From a point of view of language design and implementation, aspect languages, as well as domain specific languages for distributed and concurrent environments share many characteristics with existing distributed languages: for instance, event monitoring is fundamental for pointcut matching, different synchronization strategies and strategies for code mobility [79] may be used in actions triggered by pointcuts. However, these relationships have only been explored to a small degree. Similarly, the formal semantics and formal properties of aspect languages have not been studied yet for the distributed case and only rudimentarily for the concurrent one [63], [77].

3.5. Security

Security properties and policies over complex service-oriented and standalone applications become ever more important in the context of asynchronous and decentralized communicating systems. Furthermore, they constitute prime examples of crosscutting functionalities that can only be modularized in highly insufficient ways with existing programming language and service models. Security properties and related properties, such as accountability properties, are therefore very frequently awkward to express and difficult to analyze and enforce (provided they can be made explicit in the first place).

Two main issues in this space are particularly problematic from a compositional point of view. First, information flow properties of programming languages, such as flow properties of Javascript [64], and service-based systems [68] are typically specially-tailored to specific properties, as well as difficult to express and analyze. Second, the enforcement of security properties and security policies, especially accountability-related properties [94], [100], is only supported using ad hoc means with rudimentary support for property verification.

The ASCOLA team has recently started to work on providing formal methods, language support and implementation techniques for the modular definition and implementation of information flow properties as well as policy enforcement in service-oriented systems as well as, mostly object-oriented, programming languages.

3.6. Capacity Planning for Large Scale Distributed System

Since the last decade, cloud computing has emerged as both a new economic model for software (provision) and as flexible tools for the management of computing capacity. Nowadays, the major cloud features have become part of the mainstream (virtualization, storage and software image management) and the big market players offer effective cloud-based solutions for resource pooling. It is now possible to deploy virtual infrastructures that involve virtual machines (VMs), middleware, applications, and networks in such a simple manner that a new problem has emerged over the last two years: VM sprawl (virtual machine proliferation) that consumes valuable computing, memory, storage and energy resources, thus menacing serious resource shortages. Scientific approaches that address VM sprawl are both based on classical administration techniques like the lifecycle management of a large number of VMs as well as the arbitration and the careful management of all resources consumed and provided by the hosting infrastructure (energy, power, computing, memory, network etc.).

The ASCOLA team investigates fundamental techniques for cloud computing and capacity planning, from infrastructures to the application level. Capacity planning is the process of planning for, analyzing, sizing, managing and optimizing capacity to satisfy demand in a timely manner and at a reasonable cost. Applied to distributed systems like clouds, a capacity planning solution must mainly provide the minimal set of resources necessary for the proper execution of the applications (i.e., to ensure SLA). The main challenges in this context are: scalability, fault tolerance and reactivity of the solution in a large-scale distributed system, the analysis and optimization of resources to minimize the cost (mainly costs related to the energy consumption of datacenters), as well as the profiling and adaptation of applications to ensure useful levels of quality of service (throughput, response time, availability etc.).

Our solutions are mainly based on virtualized infrastructures that we apply from the IaaS to the SaaS levels. We are mainly concerned by the management and the execution of the applications by harnessing virtualization capabilities, the investigation of alternative solutions that aim at optimizing the trade-off between performance and energy costs of both applications and cloud resources, as well as arbitration policies in the cloud in the presence of energy-constrained resources.

4. Application Domains

4.1. Enterprise Information Systems and Services

Large IT infrastructures typically evolve by adding new third-party or internally-developed components, but also frequently by integrating already existing information systems. Integration frequently requires the addition of glue code that mediates between different software components and infrastructures but may also consist in more invasive modifications to implementations, in particular to implement crosscutting functionalities. In more abstract terms, enterprise information systems are subject to structuring problems involving horizontal composition (composition of top-level functionalities) as well as vertical composition (reuse and sharing of implementations among several top-level functionalities). Moreover, information systems have to be more and more dynamic.

Service-Oriented Computing (SOC) that is frequently used for solving some of the integration problems discussed above. Indeed, service-oriented computing has two main advantages:

- Loose-coupling: services are autonomous, in that they do not require other services to be executed;
- Ease of integration: services communicate over standard protocols.

Our current work is based on the following observation: similar to other compositional structuring mechanisms, SOAs are subject to the problem of crosscutting functionalities, that is, functionalities that are scattered and tangled over large parts of the architecture and the underlying implementation. Security functionalities, such as access control and monitoring for intrusion detection, are a prime example of such a functionality in that it is not possible to modularize security issues in a well-separated module. Aspect-Oriented Software Development is precisely an application-structuring method that addresses in a systemic way the problem of the lack of modularization facilities for crosscutting functionalities.

We are considering solutions to secure SOAs by providing an aspect-oriented structuring and programming model that allows security functionalities to be modularized. Two levels of research have been identified:

- Service level: as services can be composed to build processes, aspect weaving will deal with the orchestration and the choreography of services.
- Implementation level: as services are abstractly specified, aspect weaving will require to extend service interfaces in order to describe the effects of the executed services on the sensitive resources they control.

In 2013, we have developed techniques for the Service-Level Agreement (SLA) management for Cloud elasticity, see Sec. 6.3, as well as models and type systems for service-oriented systems, see Sec. 6.1. Furthermore, we take part in the European project A4Cloud on accountability challenges, that is, the responsible stewardship of third-party data and computations, see Sec. 8.2.

4.2. Capacity Planning in Cluster, Grid and Cloud Computing

Cluster, Grid and more recently Cloud computing platforms aim at delivering large capacities of computing power. These capacities can be used to improve performance (for scientific applications) or availability (e.g., for Internet services hosted by datacenters). These distributed infrastructures consist of a group of coupled computers that work together and may be spread across a LAN (cluster), across a WAN (Grid), and across the Internet (Clouds). Due to their large scale, these architectures require permanent adaptation, from the application to the system level and call for automation of the corresponding adaptation processes. We focus on self-configuration and self-optimization functionalities across the whole software stack: from the lower levels (systems mechanisms such as distributed file systems for instance) to the higher ones (i.e. the applications themselves such as J2EE clustered servers or scientific grid applications).

In 2013, we have confirmed the scalability of the DVMS proposal by conducting experiments on a very large scale involving up to 5K virtual machines (VM)s upon 500 nodes, thus establishing it as one of the most scalable placement algorithm for virtual machines. Moreover, we have extended the SimGrid framework by adding virtualization abstractions for hundreds of thousands of VMs. Finally, we have also provided several results on the energy efficient management of Cloud applications and infrastructures, see Sec. 6.3.

In the energy field, we have designed a set of techniques, named Optiplace, for cloud management with flexible power models through constraint programming. OptiPlace supports external models, named views. Specifically, we have developed a power view, based on generic server models, to define and reduce the power consumption of a datacenter's physical servers. We have shown that OptiPlace behaves at least as good as our previous system, Entropy, requiring as low as half the time to find a solution for the constrained-based placement of tasks for large datacenters.

4.3. Pervasive Systems

Pervasive systems are another class of systems raising interesting challenges in terms of software structuring. Such systems are highly concurrent and distributed. Moreover, they assume a high-level of mobility and context-aware interactions between numerous and heterogeneous devices (laptops, PDAs, smartphones, cameras, electronic appliances...). Programming such systems requires proper support for handling various interfering concerns like software customization and evolution, security, privacy, context-awareness... Additionally, service composition occurs spontaneously at runtime.

In 2013, we have extended the language EScala, which integrates reactive programming through events with aspect-oriented and object-oriented mechanisms, see Sec. 6.1.

5. Software and Platforms

5.1. btrCloud (and Entropy)

Participants: Jean-Marc Menaud [correspondent], Guillaume Le Louët, Thierry Bernard, Frédéric Dumont.

Orchestration, virtualization, energy, autonomic system, placement, cloud computing, cluster, data center, scheduler, grid

btrCloud is a virtual machine manager for clusters and provides a complete solution for the management and optimization of virtualized data center. btrCloud (acronym of better cloud) is composed of three parts.

The analysis function enables operatives and people in charge to monitor and analyze how a data-center works, be it on a daily basis or on the long run and predict future trends. This feature includes a performance, an analysis and a trends board.

btrCloud, by the integration of btrScript, provides (semi-)automated VM lifecycle management, including provisioning, resource pool management, VM tracking, cost accounting, and scheduled deprovisioning. Key features include a thin client interface, template-based provisioning, approval workflows, and policy-based VM placement.

Finally, several kinds of optimizations are currently available, such as energy and load balancing. The former can help save up to around 20% of the data-center energy consumption. The latter provides optimized quality of service properties for applications that are hosted in the virtualized datacenters.

btrCloud is available at http://www.btrcloud.org.

5.2. EScala and JEScala

Participants: Jacques Noyé [correspondent], Jurgen Van Ham.

AOP, inheritance, event-based programming, events, declarative events, asynchronous events, join operator, Scala

EScala is an extension of the programming language Scala with support for events as object members. EScala combines ideas of event-driven, aspect-oriented and functional reactive programming.

Events are natural abstractions for describing interactive behavior as part of an object interface. In conventional object-oriented languages, events are implemented indirectly, typically using the Observer pattern. C# eliminates the corresponding glue code and directly supports events as object members. However, events are still *explicitly* triggered at specific locations within the program.

EScala goes much further. First, it also supports *implicit* events. Akin to join points in aspect-oriented languages, these events are implicitly produced at specific execution points, such as the beginning or the end of the execution of a method. Second, *declarative events* make it possible to compose events using logical operators as well as to filter them and alter their content.

EScala events are fully integrated with object-oriented features. An event is defined in the context of its owner object. Event definitions are inherited in subclasses and event uses are late-bound. Unlike typical aspect-oriented languages, EScala preserves object-oriented encapsulation and modular reasoning.

JEScala extends EScala with support for concurrent programming (see Sec. 6.2). Events can be declared as *asynchronous* so that their handling takes place concurrently. A new composition operator, the *join* operator, inspired by the join calculus, can also be used to synchronize the concurrent activities created by asynchronous events and communicate between them.

This is joint work with the Software Technology Group at TU Darmstadt.

Prototype implementations of these languages are available through http://www.stg.tu-darmstadt.de/research.

5.3. CSLA

Participants: Thomas Ledoux [correspondent], Yousri Kouki.

Service-level agreement, Cloud computing, elasticity

Verifying non-functional properties like performance, dependability, energy consumption and economical costs of Clouds is challenging today due to ad-hoc management in terms of Quality-of-Service (QoS). We believe that a differentiating element between Cloud computing environments will be the QoS and the service-level agreement (SLA) provided by the Cloud.

CSLA, the Cloud Service Level Agreement language, allows the definition of SLA properties for arbitrary Cloud services (XaaS). CSLA addresses QoS uncertainty in unpredictable and dynamic environment and provides a cost model of Cloud computing. Besides the standard formal definition of contracts – comprising validity, parties, services definition and guarantees/violations – CSLA is enriched with features, such as QoS degradation and an advanced penalty model, thus introducing fine-grained language support for Cloud elasticity management [13].

CSLA is available at http://www.emn.fr/z-info/csla.

5.4. SAdapt

Participants: Ronan-Alexandre Cherrueau [correspondent], Mario Südholt.

Service-oriented systems, distributed programming, event-based programming, workflow patterns

The SAdapt tool provides an implementation of workflow adaptation patterns and allows the transformation of service-oriented systems implemented using Apache's CXF service infrastructure in terms of high-level declarative service transformations. The transformations are defined using an expressive language that supports matching of the execution of service-based systems in terms of flexible patterns over service compositions.

The SAdapt tool has partially been developed and is employed in the A4Cloud EU project (see Sec. 8.2) as a basis for our work on the enforcement of accountability properties in complex cloud-based systems.

The SAdapt tool and its application, notably to the security hardening of service systems that use OAuth 2 for the authorization of resource accesses is available at http://a4cloud.gforge.inria.fr/doku.php?id=start:advservcomp.

6. New Results

6.1. Software composition

Participants: Akram Ajouli, Diana Allam, Ronan-Alexandre Cherrueau, Rémi Douence, Hervé Grall, Florent Marchand de Kerchove de Denterghem, Jacques Noyé, Jean-Claude Royer, Mario Südholt.

6.1.1. Service-oriented computing

Services are frequently implemented using object-oriented frameworks. In this context, two properties are particularly important: (i) a loose coupling between the service layer and the object layer, allowing evolution of the service layer with a minimal impact on the object layer, (ii) interoperability induced by the substitution principle associated to subtyping in the object layer, thus allowing to freely convert a value of a subtype into a supertype. However, through experimentation with Apache's popular service framework CXF, we observed some undesirable coupling and interoperability issues due to the failure of the substitution principle [23]. Therefore we have proposed a new specification method for the data binding used to translate data between the object and service layers [24]. We have shown that if the CXF framework follows the specification, the substitution principle is satisfied, with all its advantages.

6.1.2. Modularity and program transformations

Refactoring tools are commonly used for remodularization tasks. Basic refactoring operations are combined to perform complex program transformations, but the resulting composed operations are rarely reused, even partially, because popular tools have few support for composition. In [31], we have recast two calculus for static composition of refactorings in a type system framework and we have discussed their use for inferring useful properties. We have illustrated the value of support for static composition in refactoring tools with a complex remodularization use case: a round-trip transformation between programs conforming to the Composite and Visitor patterns. Composite and Visitor design patterns have dual properties with respect to modularity, thus they are good candidates to explore their transformations. In [22] we have extended our initial refactoring-based round-trip transformation between these two structures and we have studied how that transformation is impacted by four variations in the implementation of these patterns. We have validated that study by computing the smallest preconditions for the resulting transformations. We have also automated the transformation and applied it to JHotDraw, where the studied variations occur. Finally, [11] presents more exhaustively modular transformations and design patterns. We have also proposed a reversible transformation in the Singleton pattern to benefit from optimization by introducing this pattern and flexibility by its suppression according to the requirements of the software user.

6.1.3. Domain specific languages

In the context of Charles Prud'hommes' PhD Thesis, we have developed a domain specific language in order to specify strategies of filtering propagation in constraint solvers. Indeed, constraint programming replaces brute force generate-and-test by the exploration of the solution space based on incremental instantiation and constraint propagation. Strategies of incremental instantiation (also known as heuristics) have been heavily studied. However, most solvers propagate constraints with a simple fix point computation based on a queue of constraints to propagate (or several queues in order to deal with the grain/cost of filtering algorithms). This technique has a good behavior in general but for a given problem a dedicated strategy can be more efficient. Our declarative DSL and its support in the new version of the constraint solver Choco [19], [52] enables us to easily experiment with different propagation strategies. Moreover, our DSL supports properties such as completeness, intended incompleteness or non ambiguity.

6.1.4. Constructive security

In the field of techniques for the development of secure software systems we have presented results on the enforcement of security properties in service-oriented systems and Javascript programs.

Concerning the security of service-based systems, we have first presented a software framework that harnesses a type based policy language and aspect-based support for protocol adaptation in service-oriented systems by means of flexible reference monitors [29], [28]. We have shown how this framework improves the security, interoperability and evolution issues of service systems using the OAuth 2.0 standard for the authorization of resource accesses. The OAuth 2 protocol is a recent IETF standard devoted to providing authorization to clients requiring access to specific resources over HTTP. It was recently adopted by major internet companies and software editors, such as Google, Facebook, Microsoft, and SAP. We have shown how to improve the security of software systems that use OAuth 2 in the presence of different kinds of clients.

Furthermore, we have developed a new notion of transformation operators, so-called workflow adaptation schemas (WASs) for service compositions that facilitates the integration and modification of security functionalities of service-oriented systems [30]. These schemas may be generic and specialized through parameter instantiation. A set of schemas therefore effectively provides a domain-specific language for the transformation of service-oriented applications. We have developed a set of specific schemas and applied them to the OAuth 2 standard in order to implement state-based security hardening strategies. We have also implemented tool support for WASs and implemented some of the security scenarios involving OAuth 2 (see Sec. 5.4).

Finally, we have shown that a wide range of strategies to make secure JavaScript-based applications can be described pertinently using aspects [42]. To this end, we have reviewed major categories of approaches to make client-side applications secure and have discussed uses of aspects that exist for some of them. We also propose aspect-based techniques for the categories that have not been studied previously. We have given examples of applications where aspects are useful as a general means to flexibly express and implement security policies for JavaScript.

6.2. Aspect-Oriented Programming

Participants: Rémi Douence, Ismael Figueroa, Jacques Noyé, Mario Südholt, Nicolas Tabareau, Jurgen Van Ham.

6.2.1. Aspects in a concurrent and distributed setting

Aspect oriented programming modularizes crosscutting concerns by gathering several join points. In the context of distributed applications these point cuts can be on different machines. In this case, a sequence of join points must be defined as a sequence of logical joint points (à la Lamport). We propose an aspect oriented languages to define distributed aspects in JavaScript in a distributed context. Our proposal [18] is based on vector clocks in order to logically relate join points and can ignore "illogical" (that is late or early) join points. In can also enforce causal communications when no join point must be discarded. We have exemplified the advantages of our technique with different applications such as a discussion forum, a retweet scenario and a web browser.

Multiparty session types allow the definition of distributed processes with strong communication safety properties. A global type is a choreographic specification of the interactions between peers, which is then projected locally in each peer. Well-typed processes behave accordingly to the global protocol specification. Multiparty session types are however monolithic entities that are not amenable to modular extensions. Also, session types impose conservative requirements to prevent any race condition, which prohibit the uniform application of extensions at different points in a protocol. We have proposed a means to support modular extensions with *aspectual session types* [47], a static pointcut/advice mechanism at the session type level. To support the modular definition of crosscutting concerns, we augment the expressivity of session types to allow harmless race conditions. We formally prove that well-formed aspectual session types entail communication safety. As a result, aspectual session types make multiparty session types more flexible, modular, and extensible.

We have added dedicated concurrency support to EScala, our extension of Scala that introduces composable *declarative events* as a way to integrate Aspect-Oriented Programming and Event-Based Programming in the context of Object-Oriented Programming. In JEScala, Events, which were synchronous in EScala, can be declared as *asynchronous* so that they are handled concurrently to their emitter. Moreover, two new operators, a join and a choice operator, inherited from the join calculus - hence the name of the new prototype, can now be used to compose events and control concurrency. In [48], we present JEScala, show that it captures coordination schemas in a more expressive and modular way than plain join languages and provide a first performance assessment.

6.2.2. Effective aspects

We have proposed a novel approach to embed pointcut/advice aspects in a typed functional programming language like Haskell. Aspects are first-class, can be deployed dynamically, and the pointcut language is extensible. Type soundness is guaranteed by exploiting the underlying type system, in particular phantom types and a new anti-unification type class. The use of monads brings type-based reasoning about effects for the first time in the pointcut/advice setting and enables modular extensions of the aspect language [46], [16].

To allow a type-safe embedding of aspects in Haskell, we had to develop a notion of anti-unification in Haskell type system. The anti-unification problem is that of finding the most specific pattern of two terms. While dual to the unification problem, anti-unification has rarely been considered at the level of types. We have developed an algorithm to compute the least general type of two types in Haskell, using the logic programming power of type classes [53]. That is, we have defined a type class for which the type class instances resolution performs anti-unification.

6.2.3. Reasoning about aspect interference

When a software system is developed using several aspects, special care must be taken to ensure that the resulting behavior is correct. This is known as the *aspect interference problem*, and existing approaches essentially aim to detect whether a system exhibits problematic interferences of aspects. We have described how to control aspect interference by construction by relying on the type system. More precisely, we combine a monadic embedding of the pointcut/advice model in Haskell with the notion of membranes for aspect-oriented programming [34]. Aspects must explicitly declare the side effects and the context they can act upon. Allowed patterns of control flow interference are declared at the membrane level and statically enforced. Finally, computational interference between aspects is controlled by the membrane topology. To combine independent and reusable aspects and monadic components into a program specification we use *monad views*, a recent technique for conveniently handling the monadic stack.

Oliveira and colleagues recently developed a powerful model to reason about mixin-based composition of effectful components and their interference, exploiting a wide variety of techniques such as equational reasoning, parametricity, and algebraic laws about monadic effects. Our work addresses the issue of reasoning about interference with effectful aspects in the presence of unrestricted quantification through pointcuts. While global reasoning is required, we have shown that it is possible to reason in a compositional manner, which is key for the scalability of the approach in the face of large and evolving systems. We have established a general equivalence theorem that is based on a few conditions that can be established, reused, and adapted separately as the system evolves. Interestingly, one of these conditions, local harmlessness, can be proven by a translation to the mixin setting, making it possible to directly exploit previously established results about certain kinds of harmless extensions [33].

In aspect-oriented programming (AOP) languages, advice evaluation is usually considered as part of the base program evaluation. While viewing aspects as part of base level computation clearly distinguishes AOP from reflection, it also comes at a price: because aspects observe base level computation, evaluating pointcuts and advice at the base level can trigger infinite regression. To avoid these pitfalls, we have introduced levels of execution in the programming language, thereby allowing aspects to observe and run at specific, possibly different, levels. We adopt a defensive default that avoids infinite regression, and gives advanced programmers the means to override this default using level-shifting operators [21].

6.3. Resource management in Cloud computing

Participants: Frederico Alvares, Gustavo Bervian Brand, Yousri Kouki, Adrien Lèbre, Thomas Ledoux, Guillaume Le Louët, Jean-Marc Menaud, Jonathan Pastor, Flavien Quesnel, Mario Südholt.

We have contributed on several topics: multiple autonomic managers for Cloud infrastructure, SLA management for Cloud elasticity, fully distributed and autonomous virtual machine scheduling, and simulator toolkits for IaaS platforms.

6.3.1. Cloud infrastructure based on multiple autonomic managers

One of the main reasons for the wide adoption of Cloud Computing is the concept of elasticity. Implementing elasticity to tackle varying workloads while optimizing infrastructures (e.g. utilization rate) and fulfilling the application requirements on Quality of Service should be addressed by self-adaptation techniques able to manage complexity and dynamism. However, since Cloud systems are organized in different but dependent Cloud layers, self-management decisions taken in isolation in a certain layer may indirectly interfere with the decision taken by an other layer. Indeed, non-coordinated managers may lead to conflicting decisions and consequently to non-desired states.

We have proposed a framework for the coordination of multiple autonomic managers in cloud environments [25]. The PhD thesis of Frederico Alvares [12], defended in April 2013, is based on this framework. This thesis proposes a self-adaptation approach that considers both application internals (architectural elasticity) and infrastructure (resource elasticity), managed by multiple autonomic managers, to reduce the energy footprint in Cloud infrastructures.

6.3.2. SLA Management for Cloud elasticity

Elasticity is the intrinsic element that differentiates Cloud Computing from traditional computing paradigms, since it allows service providers to rapidly adjust their needs for resources to absorb the demand and hence guarantee a minimum level of Quality of Service (QoS) that respects the Service Level Agreements (SLAs) previously defined with their clients. However, due to non-negligible resource initiation time, network fluctuations or unpredictable workload, it becomes hard to guarantee QoS levels and SLA violations may occur. The main challenge of service providers is to maintain its consumer's satisfaction while minimizing the service costs due to resources fees. The PhD thesis of Yousri Kouki [13], defended in December, proposes different contributions to address this issue: CSLA, a specific language to describe SLA for Cloud services ; HybridScale, an auto-scaling framework driven by SLA [39], [17].

6.3.3. Fully Distributed and Autonomous Virtualized Environments

We have consolidated the DVMS system to obtain a fully distributed virtual machine scheduler [44]. This system makes it possible to schedule VMs cooperatively and dynamically in large scale distributed systems. Simulations (up to 64K VMs) and real experiments both conducted on the Grid'5000 large-scale distributed system [44] showed that DVMS is scalable. This building block is a first element of a more complete cloud OS, entitled DISCOVERY (DIStributed and COoperative mechanisms to manage Virtual EnviRonments autonomicallY) [56]. The ultimate goal of this system is to overcome the main limitations of the traditional server-centric solutions. The system, currently under investigation in the context of the Jonathan Pastor's PhD, relies on a peer-to-peer model where each agent can efficiently deploy, dynamically schedule and periodically checkpoint the virtual environments it manages.

6.3.4. Testing the cloud

Computer science, as other sciences, needs instruments to validate theoretical research results, as well as software developments. Although simulation and emulation are generally used to get a glance of the behavior of new algorithms, they use over-simplified models in order to reduce their execution time and thus cannot be accurate enough. Leveraging a scientific instrument to perform actual experiments is an undeniable advantage. However, conducting experiments on real environments is still too often a challenge for researchers, students, and practitioners: first, because of the unavailability of dedicated resources, and second, because of the inability to create controlled experimental conditions, and to deal with the wide variability of software requirements. During 2013, we have contributed to a new topic addressing the "testing the cloud" challenge. First, we have presented the latest mechanisms we have designed to enable the automated deployment of the major open-source IaaS cloudkits (i.e., Nimbus, OpenNebula, CloudStack, and OpenStack) on Grid'5000 [26]. Providing automatic, isolated and reproducible deployments of cloud environments lets end-users study and compare each solution or simply leverage one of them to perform higher-level cloud experiments (such as investigating Map/Reduce frameworks or applications). Moreover, we have presented EXECO, a library that provides easy and efficient control of large scale experiments through a set of tools well as tools designed for scripting distributed computing experiments on any computing platform. We have illustrated its interest by presenting two experiments dealing with virtualization technologies on the Grid'5000 testbed [37].

6.3.5. Adding virtualization abstractions into the Simgrid toolkit

In the context of the ANR SONGS project and in collaboration with Takahiro Hirofuchi, researcher at AIST (Japan), we have extended the Simgrid framework to be able to simulate virtualized distributed infrastructures [35]. In addition, we have proposed the first class support of live migration operations within such a simulator toolkit for large scale distributed infrastructures. We have developed a resource share calculation mechanism for VMs and a live migration model implementing the precopy migration algorithm of Qemu/KVM. We have confirmed that our simulation framework correctly reproduced live migration behaviors of the real world under various conditions [36].

6.3.6. Power and energy management in the cloud

Power management has become one of the main challenges for data center infrastructures. Currently, the cost of powering a server is approaching the cost of the server hardware itself, and, in a near future, the former will continue to increase, while the latter will go down. In this context, virtualization is used to decrease the number of servers, and increase the efficiency of the remaining ones.

First, in [43] we have proposed an approach and a model to estimate the total power consumption of a virtual machine, by taking into account its static (e.g. memory) and dynamic (e.g. CPU) consumption of resources. Second, we have rewritten the Entropy framework (in OptiPlace) to give it the support of external models, named views. Entropy, based on the Constraint Programming solver Choco written in Java, does not really scale well. We have studied Entropy's scalability properties [32] and have then integrated heuristics and constraints in OptiPlace [40].

The evaluation of these policies on real infrastructures has become an important and difficult issue. The corresponding techniques have become so complex that there is a need for load injection frameworks able to inject resource load in a tested datacenter instead of model-driven simulation. For this reason we have developed StressCloud [41], [51], a framework to manipulate the activities of a group of Virtual Machines and observe the resulting performance.

7. Bilateral Contracts and Grants with Industry

7.1. Cooperation with SIGMA group

Participants: Thomas Ledoux, Simon Dupont.

In 2012, we have started a two-fold cooperation with Sigma Group (http://www.sigma.fr), a software editor and consulting enterprise. The cooperation consists in a joint (a so-called Cifre) PhD on eco-elasticity of software for the Cloud and the sponsorship of several engineering students at the MSc-level.

As a direct consequence of the increasing popularity of cloud computing solutions, data centers are amazingly growing and hence have to urgently face with the energy consumption issue. The aim of Simon Dupont's PhD, started in November 2012, is to explore the *software elasticity* capability in Software-as-a-Service (SaaS) development to promote the management of SaaS applications that are more flexible, more reactive to environment changes and therefore self-adaptive for a wider range of contexts. As a result, SaaS applications become more elastic and by transitivity more susceptible to energy constraints and optimization issues. In 2013, we presented our first results at ECOCITY, the World Summit on sustainable cities (http://www.ecocity-2013.com/en) and at NEM Summit (http://nem-summit.eu).

8. Partnerships and Cooperations

8.1. National Initiatives

8.1.1. CominLabs laboratory of excellence

8.1.1.1. EPOC

Participants: Jean-Marc Menaud [coordinator], Thomas Ledoux.

The project EPOC (Energy Proportional and Opportunistic Computing system) is an (academic) Labex CominLabs project running for 4 years. Four other partners collaborate within the project that is coordinated by ASCOLA: Myriads team, ENIB, ENSTB and University of Nantes. In this project, partners aim at focusing on energy-aware task execution from the hardware to application's components in the context of a *mono-site* data center (all resources are in the same physical location) which is connected to the *regular electric Grid and to renewable energy sources* (such as windmills or solar cells). Three major challenges are addressed in this context: Optimize the energy consumption of distributed infrastructures and service compositions in the presence of ever more dynamic service applications and ever more stringent availability requirements for services; Design a clever cloud's resource management which takes advantage of renewable energy availability to perform opportunistic tasks, then exploring the trade-off between energy saving and performance aspects in large-scale distributed system; Investigate energy-aware optical ultra high-speed interconnection networks to exchange large volumes of data (VM memory and storage) over very short periods of time.

One of the strengths of the proposal is to provide a systematic approach, and use a single model for the system (from hard to soft) by mixing constraint programming and behavioral models to manage energy consumption in data centers.

8.1.1.2. SecCloud

Participants: Jacques Noyé [coordinator], Florent Marchand de Kerchove de Denterghem, Mario Südholt.

The high-level objective of the 3-year SecCloud (Secure Scripting for the Cloud) project is to enhance the security of devices on which web applications can be downloaded, i.e. to enhance client-side security in the context of the Cloud. In order to do so, the project relies on a language-based approach, focusing on three related issues:

- The definition of security policies for web architectures, especially on the client-side.
- Formally-proven analyses of web programming languages.
- Multi-level enforcement mechanisms for the security policies (based on static and dynamic analysis encompassing application-level and system-level software).

ASCOLA members are mainly interested in JavaScript as a programming language as well as the use of aspects as a seamless path from the definition of security policies and their composition to their implementation.

This year we have proposed new means for the modularization of JavaScript-based security mechanisms and policies (see 6.1).

8.1.2. ANR

8.1.2.1. CESSA (ANR/ARPEGE)

Participants: Mario Südholt [coordinator], Diana Allam, Rémi Douence, Hervé Grall, Jean-Claude Royer.

The project CESSA (Compositional Evolution of Secure Services with Aspects) is an (industrial) ANR project running for 3 years months, with funding amounting to 290 KEUR for ASCOLA from Jan. 10 on. Three other partners collaborate within the project that is coordinated by ASCOLA: a security research team from Eurecom, Sophia-Antipolis, the Security and Trust team from SAP Labs, also located at Sophia-Antipolis, and IS2T, an innovative start-up company developing middleware technologies located at Nantes. The project deals with security in service-oriented architectures.

This year our group has contributed several scientific publications as part of the project. All partners have been involved in the publication of a unifying model for WD*/SOAP-based and RESTful web services. Furthermore, we have formally defined a type system that is safe in the presence of malicious attackers and insecure communication channels (see 6.1).

All information is available from the CESSA web site: http://cessa.gforge.inria.fr.

8.1.2.2. MyCloud (ANR/ARPEGE)

Participants: Thomas Ledoux [coordinator], Jean-Marc Menaud, Yousri Kouki, Frederico Alvares.

The MyCloud project is an ANR/ARPEGE project running for 42 months, starting in Nov. 2010. It was accepted in Jul. 2010 for funding amounting to 190 KEUR (ASCOLA only). MyCloud involves a consortium with three academic partners (Inria, LIP6, EMN) and one industrial partner (We Are Cloud).

Cloud Computing provides a convenient means of remote on-demand and pay-per-use access to computing resources. However, its ad-hoc management of quality-of-service (QoS) and SLA poses significant challenges to the performance, dependability and costs of online cloud services.

The objective of MyCloud (http://mycloud.inrialpes.fr) is to define and implement a novel cloud model: SLAaaS (SLA as a Service). The SLAaaS model enriches the general paradigm of Cloud Computing and enables systematic and transparent integration of SLA to the cloud [45], [50]. From the cloud provider's point of view, MyCloud proposes autonomic SLA management to handle performance, availability, energy and cost issues in the cloud. From the cloud customer's point of view, MyCloud provides SLA governance allowing cloud customers to be part of the loop and to be automatically notified about the state of the cloud, such as SLA violation and cloud energy consumption.

This year, the ASCOLA project-team has proposed: (i) SCAling, an auto-scaling approach driven by SLA and based on a MAPE-K control loop framework [39]; (ii) RightCapacity, a cross-layer (application-resource) Cloud elasticity approach based on queueing network model, taking into account the SLA concept and the Cloud economic model [17].

8.1.2.3. SONGS (ANR/INFRA)

Participants: Adrien Lèbre [coordinator], Flavien Quesnel, Jonathan Pastor, Takahiro Hirofuchi.

The SONGS project (Simulation of Next Generation Systems) is an ANR/INFRA project running for 48 months (starting in January 2012 with an allocated budget of 1.8MEuro, 95KEuro for ASCOLA).

The consortium is composed of 11 academic partners from Nancy (AlGorille, coordinator), Grenoble (MESCAL), Villeurbanne (IN2P3 Computing Center, GRAAL/Avalon - LIP), Bordeaux (CEPAGE, HiePACS, RUNTIME), Strasbourg (ICPS - LSIIT), Nantes (ASCOLA), Nice (MASCOTTE, MODALIS).

The goal of the SONGS project (http://infra-songs.gforge.inria.fr) is to extend the applicability of the SimGrid simulation framework from Grids and Peer-to-Peer systems to Clouds and High Performance Computation systems. Between January and December 2013, we have hosted Takahiro Hirofuchi, Researcher at AIST (Japan). During his stay, we have extended the Simgrid toolkit with VM abtractions [35]. In addition to elementary functionalities such as VM start/stop, we have delivered the first accurate model of live migration operations within IaaS systems [36].

8.1.3. FSN

8.1.3.1. OpenCloudware (FSN)

Participants: Jean-Marc Menaud [coordinator], Thomas Ledoux, Yousri Kouki.

The OpenCloudware project is coordinated by France Telecom, funded by the French Fonds National pour la Société Numérique (FSN, call Cloud n°1) and endorsed by competitiveness clusters Minalogic, Systematic and SCS. OpenCloudware is developed by a consortium of 18 partners bringing together industry and academic leaders, innovative technology start-ups and open source community expertise. Duration: 36 months - 2012–2014.

The OpenCloudware project aims at building an open software engineering platform, for the collaborative development of distributed applications to be deployed on multiple Cloud infrastructures. It will be available through a self-service portal. We target virtualized multi-tier applications such as JavaEE - OSGi. The results of OpenCloudware will contain a set of software components to manage the lifecycle of such applications, from modelling(Think), developing and building images (Build), to a multi-IaaS compliant PaaS platform (Run).

The ASCOLA project-team is mainly involved in the sub-projects "Think" (SLA model accross Cloud layers) and "Run" (virtual machine manager for datacenters and placement constraints). In 2013, the team has developed btrCloudStack, a private cloud based on the OpenSource CloudStack and integrating the work on placement rules and energy optimization.

8.2. European Initiatives

8.2.1. FP7 Projects

8.2.1.1. A4Cloud (IP)

Participants: Mario Südholt [coordinator], Ronan-Alexandre Cherrueau, Rémi Douence, Hervé Grall, Jean-Claude Royer, Mohamed Sellami.

The A4Cloud project (Accountability for the Cloud) is an integrated EU project, coordinated by HP, UK, on the topic of accountability, that is, the responsible stewardship of private data, in the Cloud. This 42-months project started in Oct. 2012 and Ascola's funding amounts to 600 KEuro.

The project involves 13 partners: in addition to HP, two enterprises (SAP AG, Germany; ATC, Greece), a non-governmental organisation (the Cloud Security Alliance, CSA) and 9 universities and research organisations (EMNantes and Eurecom, France; HFU. Furtwangen, Germany; Karlstadt U., Sweden; U. Malaga, Spain; Queen Mary U., U.K.; U. Stavanger and Sintef, Norway; Tilburg U., The Netherlands).

A4Cloud focuses on accountability properties for the cloud and other future internet services as the most critical prerequisite for effective governance and control of corporate and private data processed by cloud-based IT services. The research being conducted in the project will increase trust in cloud computing by devising methods and tools, through which cloud stakeholders can be made accountable for the privacy and confidentiality of information held in the cloud. These methods and tools will combine risk analysis, policy enforcement, monitoring and compliance auditing. They will contribute to the governance of cloud activities, providing transparency and assisting legal, regulatory and socio-economic policy enforcement. For further information, see http://www.a4cloud.eu.

ASCOLA is mainly involved in the sub-projects on the enforcement of accountability and security policies, as well as tool validation efforts.

8.2.2. Collaborations in European Programs, except FP7

8.2.2.1. SCALUS (MC ITN)

Participants: Adrien Lèbre [coordinator], Mario Südholt, Gustavo Bervian Brand.

The vision of the Scalus (SCALing by means of Ubiquitous Storage) (MC international training network) was to deliver the foundation for ubiquitous storage systems, which can be scaled with respect to multiple characteristics (capacity, performance, distance, security, ...). Providing ubiquitous storage will become a major demand for future IT systems and leadership in this area can have significant impact on European competitiveness in IT technology. To get this leadership, it is necessary to invest into storage education and research and to bridge the current gap between local storage, cluster storage, grid storage, and cloud storage. During the four last years, the consortium proceeded into this direction by building the first interdisciplinary teaching and research network on storage issues. It consisted of top European institutes and companies in storage and cluster technology, building a demanding but rewarding interdisciplinary environment for young researchers.

The network involved the following partners: University of Paderborn (Germany, coordinator), Barcelona Super Computing (Spain), University of Durham (England), University of Frankfurt (Germany), ICS-FORTH (Greece), Universidad Polytecnica de Madrid (Spain), EMN/ARMINES (France), Inria Rennes Bretagne Atlantique (France), XLAB (Slovenia), University of Hamburg (Germany), Fujistu Technology Systems (Germany).

The overall funding of the project by the European Union was closed to 3,3 MEUR. ASCOLA's share amounts to 200 KEUR. The project ended in October.

8.3. International Initiatives

8.3.1. Inria Associate Teams

8.3.1.1. REAL

Title: Reasoning about Effects in Aspect Languages

Inria principal investigator: Jacques Noyé

International Partner (Institution - Laboratory - Researcher):

University of Chile (Chile) - PLEIAD - Éric Tanter

Duration: 2013 - 2015

See also: http://real.gforge.inria.fr

The scientific goals of this collaboration are twofold. On the one hand, we plan to develop a theory to reason about aspect interference in general, i.e. covering both base/aspect and aspect/aspect composition and, more precisely, to reason about effects that can be used by aspects. This provides foundations for secure aspects. On the other hand, we plan to study how secure aspects can help construct security aspects, i.e. aspects ensuring security policies. Our case study is web applications built with web scripting languages such as JavaScript.

8.3.2. Inria International Partners

8.3.2.1. Informal International Partners

ASCOLA is closely cooperating with several other international partners:

- **AIST, Japan; Dr. Takahiro Hirofuchi.** This year we have started a cooperation on the simulation of Cloud infrastructures and new scheduling algorithms for virtual environments 6.3. Dr. Takahiro Hirofuchi has visited the team in 2013 in the context of this collaboration.
- **Soft team, VU Brussel, Belgium; Prof. Wolfgang De Meuter.** In the context of a joint PhD thesis that started in Dec. 2013, we are working on new means for the declarative definition and efficient implementation of event-based systems.
- **Software Technology Group, TU Darmstadt, Germany; Prof. Mira Mezini.** In the context of a joint PhD thesis we are working on a common model for object-oriented programming, event-based programming and aspect-oriented programming, see 6.2.

8.4. International Research Visitors

8.4.1. Visits of International Scientists

- Éric Tanter (University of Chile), 2-12 July 2013, in the context of the Associate Team REAL.
- Takahiro Hirofuchi (AIST, Japan), 1 Jan-31 Dec 2013, in the context and with the financial support of the ANR SONGS project.

8.4.1.1. Internships

Ismael FIGUEROA (from Apr 2013 until Jun 2013)

Compositional Reasoning About Aspect Interference (this leads to a publication in Modularity'14 [33]).

Institution: University of Chile (Chile)

9. Dissemination

9.1. Scientific Animation

9.1.1. Event organization, animation of scientific community

9.1.1.1. International

Board memberships

- **AITO:** P. Cointe is a member of this international organization that manages the ECOOP conference series.
- **AOSD-Europe:** M. Südholt is a member of the governing board of AOSD-Europe, the European Network of Excellence in AOSD, which provides financial support and organizes scientific events on Modularity.
- **Aspect-oriented Software Association (AOSA):** M. Südholt is a member of this international organization (headquarted in California) that manages the international conference on Modularity and provides support for scientific and industrial events on modularity in a large sens. He has been its chair for two years until April 2013.
- **COST Action:** J.-M. Menaud has been a management committee member of the European COST Action IC0804: Energy efficiency in large scale distributed systems from 2009 to 2013.
- **OW2:** Jean-Marc Menaud has been a member of the OW2 board committee since 2010.
- **Transactions of AOSD.** M. Südholt is a member of the editorial board of this Springer journal.

Events

- **REM 2013:** M. Südholt has been a general co-chair of the International Workshop on Reactivity, Events and Modularity, collocated with the ACM SPLASH conference in Indianapolis. http://soft.vub.ac.be/REM13
- **VTDC 2013:** A. Lèbre has been the general chair of the 7th edition of the International Workshop on Virtualization Technologies in Distributed Computing, collocated with the ACM HPDC conference in New York http://www.grid-appliance.org/wiki/index.php/VTDC13.

9.1.1.2. National

Board memberships

- **CominLabs laboratory of excellence:** P. Cointe is a member of its management board. J.-M. Menaud has been the co-coordinator of the focus on Energy and resource efficiency in ICT since 2011 of this regional excellence cluster in research. M. Südholt has been co-coordinating its security focus since 2011.
- **Green Lab Center:** T. Ledoux and J.-M. Menaud are members of the board of the Green Lab Center association. This association promotes and disseminates Green IT practices and research prototypes to the world of education, research and companies http://www.greenlabcenter.fr.
- **Innovation cluster Images & Réseaux:**: P. Cointe is a member of the mangagement board of this cluster. M. Südholt serves on its board for the selection and validation of industrial and academic projects.

Events

- **Eco-conception of software.** J.-M. Menaud has co-organized two events on GreenIT, *HPC & Cloud : Are they green ?* in Lille and *What else / what next ?* in Luxembourg.
- **Regional Doctoral School STIM:** T. Ledoux has co-organized JDOC, an event promoting contact between PhD students of the Doctoral School (Apr. 11, 2013).
- **Workshop on SLA Management in Cloud computing:** T. Ledoux has been the main coordinator of first workshop on SLA management in the Cloud organized with both industrial and research speakers http://compas2013.inrialpes.fr/evenements-satellites/ateliers/slacloud.
- **Workshop on Software Quality.** The Ascola team has co-organized a workshop on Software Quality in June 2013, which has been attended by 60 participants. http://www.lina.sciences.univ-nantes.fr/apcb/JSQ2013

9.1.2. Committee participations, reviewing activities, collective duties, etc.

- **P. Cointe:** He is the head of the LINA computer science laboratory (UMR 6241) http://www.lina.univ-nantes.fr/.

  He was a member of the ECOOP 2013 program committee. He served the French programming language community as member of the GDR GPL “Phd Thesis” prize committee

  This year he was a member of the AERES visiting evaluation committee for the LTCI laboratory and a member of the PES national committee for computer sciences (CNRS section 27).

  He is a member of the board of the Doctoral School MATISSE in Rennes. He is also the chair of the STIC-MATHS committee working for the Pays de la Loire Council (CCRRDT) and the PRES L’UNAM.

- **A. Lèbre** has been the general chair of the 7th international workshop on “Virtualization Technologies in Distributed Computing” (VTDC 2013) co-located with HPDC 2013 in New York, USA. He also was a member of the program committee of the ACM/IEEE International Conference on Cluster, Cloud and Grid Computing (CCGRID 2013), the International ACM SuperComputing Conference (SC 2013), the IEEE International Conference on Cloud and Green Computing (CGC 2013), the IEEE International Conference on Big Data (BigData 2013), the Euro-Par Conference (Euro-Par 2013), and the International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC 2013). He served as Publicity Chair for the International Conference on Network and Parallel Computing (NPC 2013). He also served as external reviewers for the Austrian Starting Grant Program and the European FP7 Program. He represented Inria during the Post Consultation Workshop on Cloud Computing and Services for H2020 at Brussels.

  He is a member of the architect board of the G5K consortium and also acts the local contact point for the Nantes site. He is also member of the directory board of the CNRS/GDR ASR System group since 2012. He leads the Discovery Initiative, an open-source collaborative action, which aims at providing a fully distributed IaaS Cloudkit http://beyondtheclouds.github.io. He presented this initiative at the Summer School of Cloud Computing at Telecom SudParis, Evry France and at the Inria Alumni Jam Session on Cloud Computing.

- **T. Ledoux** has been a member of the program committees of the following conferences: the 4th International Workshop on Green and Cloud Computing Management (GCM’13), the 12th Workshop on Adaptive and Reflective Middleware (ARM’13), the 1st International Workshop on Green In Software Engineering - Green By Software Engineering (GIBSE’13) the 28th Symposium On Applied Computing (SAC’13) - track Software Engineering Aspects of Green Computing.

- **J.-M. Menaud** has served on the program committee of the 9th French Conference on Operating Systems (CFSE-9), the 8th Workshop on Virtualization in High-Performance Cloud Computing (VHPC’13) the ninth Advanced International Conference on Telecommunications (AICT 2013), the second Workshop on Collaborative and Autonomic Green computing (CAGing 2013), the third Green Communications and IT Energy-aware Technologies (ENERGY 2013), the IEEE/ACM Inter. Conference on Green Computing and Communications (GreenCom2013), the 1st Conference on Energy Efficiency on Large Scale Distributed Systems (EE-LSDS 2013), the third IEEE International Conference on Cloud and Green Computing (GCG 2013), the 6th International Conference on Contemporary Computing (IC3 2013), the International Workshop on Cloud Computing Systems, Networks, and Applications (Globecom 2013), and the second International Conference on Smart Grids and Green IT Systems (SMARTGREENS 2013).

  J.-M. Menaud is a management committee member of the european COST Action IC0804 : Energy efficiency in large scale distributed systems since 2009, the co-animator of the focus Energy and resource efficiency in ICT, in the Labex COMIN Labs since June 2011, the animator of the CNRS/GDR ASR System group since June 2009, member of the OW2 board committee from 2010 and a member of the (RenPar/CFSE/Sympa) steering committee since 2008.

- **J. Noyé** has been a program committee member of the International Conference on Software Composition (Budapest Hungary, June 2013). He is a program committee member of the 13 International Conference on Modularity (Modularity'14), which will take place in Lugano, Switzerland, in April 2014.
- **J.-C. Royer:** He has been a member of the program committees of CAL 2013, CIEL 2013, WETICE 2013, SNPD 2013, ICIS 2013, MULTIPLE 2013, and SERA 2013 and did a review for the TSI journal. He participated in the evaluation of a project for Israel Science Foundation.
- **M. Südholt** has been a member of the management board of the *Aspect-Oriented Software Association* since 2006 and management board of **AOSD-Europe** since 2008.

  He has been a member of the the steering committees of the internal conference on Modularity since 2006 and has chaired it for two years until 2013. He has been a member of the steering committee of the international conference on Software Composition since 2008.

  In 2013, he was a member of the program committes of the international conferences CloudCom'13 and Modularity:aosd'13 as well as of 1 international and 1 national workshop.

  He is a coordinator of the security domain within the CominLabs laboratory of excellence.

  He is a member of the council of the *Laboratoire Informatique de Nantes Atlantique* (LINA, UMR 6241). He also serves on the selection and evaluation committee of the competitiveness cluster Images & Réseaux. Finally, he has served on the evaluation committee for startup proposals of Atlanpole, a regional innovation body.
- **N. Tabareau** was a member of the program committees of the Journées Francophones des Langages applicatifs (JFLA'13) and Foundations of Aspect Languages (FOAL'13).

9.2. Teaching - Supervision - Juries

9.2.1. Teaching

The team is involved in the following undergraduate and graduate-level programs at EMN (the institution all of eaching staff belongs to):

- The team is a main contributor to the **engineering program of EMN**.
- Within this engineering program, the team is steering, chairing and the main contributor to a final-year **graduate-level informatics specialization**.
- Since 2009 our team has defined and set up a new three-year **engineering program on software engineering**

The team has also been involved involved in the following three MSc programs that have been carried out with partners from French and foreign universities:

- The team participates in the **MSc program "Alma"** on software architecture and distributed systems, a joint program steered by colleagues from University of Nantes. In this context, we are responsible for a 48-hour module on advanced software composition and take part in the program's governing board.
- Members of the team have taught different **courses at different study levels in Rennes** mainly organized by University of Rennes and the research institutes IRISA and Inria.

m members have taught each for about 195 hours on average in 2013 (hours of presence in front of students). Hereby, we have taken into account that researchers and some professors have not taught at times. In addition, a significant part of the program is taught by temporary staff, whose participation is managed by ASCOLA members.

9.2.2. Supervision

The team has been supervising 16 PhD thesis in 2013, of which four have been co-supervised with external partners (three with foreign partners from U. Chile; TU Darmstadt, Germany; Lancaster U., U.K.) and one with the French TASC team from Mines Nantes.

Several PhD have been defended this year: In Feb., Flavien Quesnel defended his thesis on "Toward Cooperative Management of Large-scale Virtualized Infrastructures: the Case of Scheduling". In April, Frederico Alvares defended his thesis on "Multi Autonomic Management for Optimizing Energy Consumption in Cloud Infrastructures". In Sep., Akram Ajouli defended his thesis on views and program transformations for software modularity. In Dec., Yousri Kouki defended his thesis on "SLA-driven Cloud Elasticity Management approach".

Two members of the team have been preparing an HDR in 2013 for a defense in 2014.

9.2.3. Juries

- **A. Lèbre** was a member of a selection committee for an Associate Professor position in the Telecom school in Nancy.
- **T. Ledoux** was member of the PhD committee of Erwan Daubert at Université Rennes 1.
- **J.-M. Menaud** was member of the PhD committees of Damien Borgetto at University of Toulouse, Rémi Druilhe at University of Lille, and Flavien Quesnel in Nantes.
- **J.-C. Royer** was member of the PhD committes of Akram Ajouli and Valerio Cosentino at Mines de Nantes.
- **M. Südholt** was part of a selection committee for a position of Associate Professor at University of Rennes. He was also a member of the PhD evaluation committees of Eline Philips at Vrije Universiteit Brussel and Zhoulai Fu at University of Rennes.

10. Bibliography

Major publications by the team in recent years

[1] F. J. Alvares De Oliveira, R. Sharrock, T. Ledoux. *Synchronization of Multiple Autonomic Control Loops: Application to Cloud Computing*, in "COORDINATION - International Conference on Coordination Models and Languages - 2012", Stockholm, Sweden, March 2012, http://hal.inria.fr/hal-00682914

[2] B. De Fraine, E. Ernst, M. Südholt. *Essential AOP: The A Calculus*, in "ACM Transactions on Programming Languages and Systems (TOPLAS)", December 2012, http://hal.inria.fr/hal-00676082

[3] A. El Rheddane, N. De Palma, F. Boyer, F. Dumont, J.-M. Menaud, A. Tchana. *Dynamic Scalability of a Consolidation Service*, in "International Conference on Cloud Computing, Indus Track", United States, July 2013, pp. 01-09, http://hal.inria.fr/hal-00835817

[4] G. Jaber, N. Tabareau, M. Sozeau. *Extending Type Theory with Forcing*, in "LICS 2012 : Logic In Computer Science", Dubrovnik, Croatia, June 2012, http://hal.inria.fr/hal-00685150

[5] M. Léger, T. Ledoux, T. Coupaye. *Reliable Dynamic Reconfiguration in a Reflective Component Model*, in "Proc. of the 13th Int. Symposium on Component Based Software Engineering (CBSE'10)", Tchèque, République, Lecture Notes in Computer Science, Springer-Verlag, June 2010, pp. 74-92

[6] F. Quesnel, A. Lèbre, M. Südholt. *Cooperative and Reactive Scheduling in Large-Scale Virtualized Platforms with DVMS*, in "Concurrency and Computation: Practice and Experience", December 2012, http://hal.inria.fr/hal-00675315

[7] D. SERRANO, S. BOUCHENAK, Y. KOUKI, T. LEDOUX, J. LEJEUNE, J. SOPENA, L. ARANTES, P. SENS. *Towards QoS-Oriented SLA Guarantees for Online Cloud Services*, in "IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, CCGrid 2013", Delft, Netherlands, May 2013, http://hal.inria.fr/hal-00780000

[8] N. TABAREAU, I. FIGUEROA, É. TANTER. *A Typed Monadic Embedding of Aspects*, in "Proceedings of the 12th annual international conference on Aspect-oriented software development (AOSD'13)", Fukuoka, Japan, March 2013, http://hal.inria.fr/hal-00763695

[9] N. TABAREAU. *A theory of distributed aspects*, in "9th International Conference on Aspect-Oriented Software Development (AOSD '10)", France Rennes, Saint-Malo, ACM, 2010, pp. 133–144, http://dx.doi.org/10.1145/1739230.1739246

[10] R. TOLEDO, A. NÚÑEZ, É. TANTER, J. NOYÉ. *Aspectizing Java Access Control*, in "IEEE Transactions on Software Engineering", January 2011, http://hal.inria.fr/inria-00567489/en

Publications of the year

Doctoral Dissertations and Habilitation Theses

[11] A. AJOULI. , *Vues et transformations de programmes pour la modularité des évolutions*, Ecole des Mines de Nantes, September 2013, http://hal.inria.fr/tel-00866997

[12] F. J. ALVARES DE OLIVEIRA. , *Gestion multi autonome pour l'optimisation de la consommation énergétique sur les infrastructures en nuage*, Université de Nantes, April 2013, http://hal.inria.fr/tel-00853575

[13] Y. KOUKI. , *Approche dirigée par les contrats de niveaux de service pour la gestion de l'élasticité du "nuage"*, Ecole des Mines de Nantes, December 2013, http://hal.inria.fr/tel-00919900

[14] F. QUESNEL. , *Vers une gestion coopérative des infrastructures virtualisées à large échelle : le cas de l'ordonnancement*, Ecole des Mines de Nantes, February 2013, http://hal.inria.fr/tel-00821103

Articles in International Peer-Reviewed Journals

[15] O. CHEBARO, P. CUOQ, N. KOSMATOV, B. MARRE, A. PACALET, N. WILLIAMS, B. YAKOBOWSKI. *Behind the Scenes in SANTE: A Combination of Static and Dynamic Analyses*, in "Automated Software Engineering", June 2013, 45 p. , http://www.springer.com/computer/ai/journal/10515, http://hal.inria.fr/hal-00818147

[16] I. FIGUEROA, N. TABAREAU, É. TANTER. *Effective Aspects: A Typed Monadic Embedding of Pointcuts and Advice*, in "Transactions on Aspect-Oriented Software Development", 2013, http://hal.inria.fr/hal-00872782

[17] Y. KOUKI, T. LEDOUX. *RightCapacity: SLA-driven Cross-Layer Cloud Elasticity Management*, in "International journal of next-generation computing (IJNGC)", December 2013, vol. 4, n^o 3, pp. 250-262, http://hal.inria.fr/hal-00920968

[18] P. LEGER, É. TANTER, R. DOUENCE. *Modular and Flexible Causality Control on the Web*, in "Science of Computer Programming", January 2013, http://hal.inria.fr/hal-00755127

[19] C. PRUD'HOMME, X. LORCA, R. DOUENCE, N. JUSSIEN. *Propagation Engine Prototyping with a Domain Specific Language*, in "Constraints", 2013, 21 p. [*DOI :* 10.1007/s10601-013-9151-5], http://hal.inria.fr/hal-00867604

[20] M. SELLAMI, P. DE VETTOR, M. MRISSA, D. BENSLIMANE, B. DEFUDE. *DMaaS : Syntactic, Structural and Semantic Mediation for Service Composition*, in "International Journal of Autonomous and Adaptive Communications Systems", 2014, http://hal.inria.fr/hal-00937178

[21] É. TANTER, I. FIGUEROA, N. TABAREAU. *Execution Levels for Aspect-Oriented Programming: Design, Semantics, Implementations and Applications*, in "Science of Computer Programming", 2013, http://hal.inria.fr/hal-00872786

International Conferences with Proceedings

[22] A. AJOULI, J. COHEN, J.-C. ROYER. *Transformations between Composite and Visitor implementations in Java*, in "39th Euromicro Conference on Software Engineering and Advanced Applications (SEAA 2013)", Santander, Spain, September 2013, 8 p. , http://hal.inria.fr/hal-00834149

[23] D. ALLAM, H. GRALL, J.-C. ROYER. *From Object-Oriented Programming to Service-Oriented Computing: How to Improve Interoperability by Preserving Subtyping*, in "WEBIST 2013 - 9th International Conference on Web Information Systems and Technologies", Aachen, Germany, K.-H. KREMPELS, A. STOCKER (editors), SciTePress Digital Library, March 2013, pp. 169-173, Short paper presented as a poster, http://hal.inria.fr/hal-00800153

[24] D. ALLAM, H. GRALL, J.-C. ROYER. *The Substitution Principle in an Object-Oriented Framework for Web Services: From Failure to Success*, in "The 15th International Conference on Information Integration and Web-based Applications & Services (iiWAS2013)", Vienna, Austria, G. KOTSIS (editor), Gabriele Kotsis, December 2013, Full regular paper, http://hal.inria.fr/hal-00908552

[25] F. J. ALVARES DE OLIVEIRA, T. LEDOUX, R. SHARROCK. *A framework for the coordination of multiple autonomic managers in cloud environments*, in "IEEE International Conference on Self-Adaptive and Self-Organizing Systems (SASO)", Philadelphia (PA), United States, 2013, pp. 179-188, http://hal.inria.fr/hal-00839090

[26] S. BADIA, A. CARPEN-AMARIE, A. LÈBRE, L. NUSSBAUM. *Enabling Large-Scale Testing of IaaS Cloud Platforms on the Grid'5000 Testbed*, in "TTC - 1st International Workshop on Testing The Cloud, co-located with ISSTA 2013", Lugano, Switzerland, ACM, July 2013, pp. 7-12 [*DOI :* 10.1145/2489295.2489298], http://hal.inria.fr/hal-00907888

[27] W. BENGHABRIT, H. GRALL, J.-C. ROYER, M. SELLAMI, M. AZRAOUI, K. ELKHIYAOUI, M. ÖNEN, A. SANTANA DE OLIVEIRA, K. BERNSMED. *A Cloud Accountability Policy Representation Framework*, in "CLOSER - 4th International Conference on Cloud Computing and Services Science", Barcelone, Spain, 2014, http://hal.inria.fr/hal-00941872

[28] R.-A. CHERRUEAU, O. CHEBARO, M. SÜDHOLT. *Flexible and expressive aspect-based control over service compositions in the Cloud*, in "4th International Workshop on Variability & Composition", Fukuoka, Japan, Digital Library, ACM, March 2013, http://hal.inria.fr/hal-00782427

[29] R.-A. CHERRUEAU, R. DOUENCE, J.-C. ROYER, M. SÜDHOLT, A. SANTANA DE OLIVEIRA, Y. ROUDIER, M. DELL'AMICO. *Reference monitors for security and interoperability in OAuth 2.0*, in "SETOP - 6th International Workshop on Autonomous and Spontaneous Security - 2013", Egham, United Kingdom, September 2013, http://hal.inria.fr/hal-00846810

[30] R.-A. CHERRUEAU, M. SÜDHOLT, O. CHEBARO. *Adapting workflows using generic schemas: application to the security of business processes*, in "CloudCom - 5th IEEE International Conference on Cloud Computing Technology and Science - 2013", Bristol, United Kingdom, December 2013, pp. 519-524, http://hal.inria.fr/hal-00866485

[31] J. COHEN, A. AJOULI. *Practical use of static composition of refactoring operations*, in "ACM Symposium on Applied Computing (SAC)", Coimbra, Portugal, ACM, March 2013, pp. 1700-1705 [*DOI :* 10.1145/2480362.2480684], http://hal.inria.fr/hal-00751304

[32] A. EL RHEDDANE, N. DE PALMA, F. BOYER, F. DUMONT, J.-M. MENAUD, A. TCHANA. *Dynamic Scalability of a Consolidation Service*, in "International Conference on Cloud Computing, Indus Track", United States, July 2013, pp. 01-09, http://hal.inria.fr/hal-00835817

[33] I. FIGUEROA, T. SCHRIJVERS, N. TABAREAU, É. TANTER. *Compositional Reasoning About Aspect Interference*, in "13th International Conference on Modularity (Modularity'14)", Lugano, Switzerland, April 2014, http://hal.inria.fr/hal-00919935

[34] I. FIGUEROA, N. TABAREAU, É. TANTER. *Taming aspects with monads and membranes*, in "FOAL'13: Foundations of aspect-oriented languages", Fukuoka, Japan, ACM, March 2013 [*DOI :* 10.1145/2451598.2451600], http://hal.inria.fr/hal-00808983

[35] T. HIROFUCHI, A. LÈBRE. *Adding Virtual Machine Abstractions Into SimGrid, A First Step Toward the Simulation of Infrastructure-as-a-Service Concerns*, in "Third International Conference on Cloud and Green Computing (CGC 2013)", Karlsruhe, Germany, September 2013, http://hal.inria.fr/hal-00861848

[36] T. HIROFUCHI, A. LÈBRE, L. POUILLOUX. *Adding a Live Migration Model Into SimGrid, One More Step Toward the Simulation of Infrastructure-as-a-Service Concerns*, in "5th IEEE International Conference on Cloud Computing Technology and Science (IEEE CloudCom 2013)", Bristol, United Kingdom, December 2013, http://hal.inria.fr/hal-00861882

[37] M. IMBERT, L. POUILLOUX, J. ROUZAUD-CORNABAS, A. LÈBRE, T. HIROFUCHI. *Using the EXECO toolbox to perform automatic and reproducible cloud experiments*, in "1st International Workshop on UsiNg and building ClOud Testbeds (UNICO, collocated with IEEE CloudCom 2013", Bristol, United Kingdom, September 2013, http://hal.inria.fr/hal-00861886

[38] Y. KOUKI, F. J. ALVARES DE OLIVEIRA, S. DUPONT, T. LEDOUX. *A Language Support for Cloud Elasticity Management*, in "IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, CCGrid 2014", Chicago, United States, May 2014, pp. 1-8, http://hal.inria.fr/hal-00941945

[39] Y. KOUKI, T. LEDOUX. *SCAling : SLA-driven Cloud Auto-scaling*, in "ACM Symposium on Applied Computing, SAC 2013", Coimbra, Portugal, March 2013, pp. 411-414, http://hal.inria.fr/hal-00780003

[40] G. LE LOUET, J.-M. MENAUD. *OptiPlace: designing cloud management with flexible power models through constraint programing*, in "International Conference on Utility and Cloud Computing", Dresden, Germany, December 2013, http://hal.inria.fr/hal-00931797

[41] G. LE LOUET, J.-M. MENAUD. *StressCloud: An infrastructure stresser for Virtual Machine Managers*, in "Energy Efficiency in Large Scale Distributed Systems conference", Austria, April 2013, pp. 01-08, http://hal.inria.fr/hal-00836137

[42] F. MARCHAND DE KERCHOVE, J. NOYÉ, M. SÜDHOLT. *Aspectizing JavaScript Security*, in "MISS - 3rd Workshop on Modularity In Systems Software", Fukuoka, Japan, March 2013 [*DOI :* 10.1145/2451613.2451616], http://hal.inria.fr/hal-00786258

[43] F. QUESNEL, H. KUMAR MEHTA, J.-M. MENAUD. *Estimating the Power Consumption of an Idle Virtual Machine*, in "The 2013 IEEE International Conference on Green Computing and Communications (GreenCom 2013)", Beijing, China, August 2013, http://hal.inria.fr/hal-00838982

[44] F. QUESNEL, A. LÈBRE, J. PASTOR, M. SÜDHOLT, D. BALOUEK. *Advanced Validation of the DVMS Approach to Fully Distributed VM Scheduling*, in "ISPA' 13: The 11th IEEE International Symposium on Parallel and Distributed Processing with Applications", Melbourne, Australia, July 2013, http://hal.inria.fr/hal-00817369

[45] D. SERRANO, S. BOUCHENAK, Y. KOUKI, T. LEDOUX, J. LEJEUNE, J. SOPENA, L. ARANTES, P. SENS. *Towards QoS-Oriented SLA Guarantees for Online Cloud Services*, in "IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, CCGrid 2013", Delft, Netherlands, May 2013, pp. 50-57, http://hal.inria.fr/hal-00780000

[46] N. TABAREAU, I. FIGUEROA, É. TANTER. *A Typed Monadic Embedding of Aspects*, in "12th annual international conference on Aspect-Oriented Software Development (Modularity-AOSD'13)", Fukuoka, Japan, March 2013, http://hal.inria.fr/hal-00763695

[47] N. TABAREAU, M. SÜDHOLT, É. TANTER. *Aspectual Session Types*, in "Modularity - 13th International Conference on Modularity", Lugano, Switzerland, 2014, http://hal.inria.fr/hal-00872791

[48] J. VAN HAM, G. SALVANESCHI, M. MEZINI, J. NOYÉ. *JEScala: Modular Coordination with Declarative Events and Joins*, in "Modularity '14 - 13th International Conference on Modularity", Lugano, Switzerland, E. ERNST (editor), April 2014, http://hal.inria.fr/hal-00862332

National Conferences with Proceedings

[49] L. EYRAUD-DUBOIS, A. LÈBRE, P. MARTINEAU, A. SOUKHAL, V. T'KINDT, D. TRYSTRAM. *A Server Consolidation Problem: Definition and Model*, in "Proceedings of the 14th conference ROADEF", Troyes, France, 2013, http://hal.inria.fr/hal-00921363

[50] Y. KOUKI, T. LEDOUX, D. SERRANO, S. BOUCHENAK, J. LEJEUNE, L. ARANTES, J. SOPENA, P. SENS. *SLA et qualité de service pour le Cloud Computing*, in "Conférence d'informatique en Parallélisme, Architecture et Système, ComPAS 2013", Grenoble, France, January 2013, pp. 1-11, http://hal.inria.fr/hal-00764951

[51] G. LE LOUET, J.-M. MENAUD. *StressCloud, un outils d'évaluation pour les gestionnaires d'infrastructure virtualisée*, in "Conférence Française en Systèmes d'Exploitation", France, January 2013, pp. 01-12, http://hal.inria.fr/hal-00836139

[52] C. PRUD'HOMME, X. LORCA, R. DOUENCE, N. JUSSIEN. *Prototyper des Moteurs de Propagation avec un DSL*, in "Journées française de la Programmation par Contraintes", Aix-en-Provence, France, 2013, pp. 279-288, http://hal.inria.fr/hal-00867713

[53] N. TABAREAU, É. TANTER, I. FIGUEROA. *Anti-Unification with Type Classes*, in "Journées Francophones des Langages Applicatifs (JFLA)", Aussois, France, February 2013, http://hal.inria.fr/hal-00765862

Conferences without Proceedings

[54] D. BALOUEK, A. LÈBRE, F. QUESNEL. *Flauncher and DVMS – Deploying and Scheduling Thousands of Virtual Machines on Hundreds of Nodes Distributed Geographically*, in "IEEE International Scalable Computing Challenge (SCALE 2013), held in conjunction with CCGrid'2013", Delft, Netherlands, 2013, http://hal.inria.fr/hal-00920094

Research Reports

[55] A. AJOULI, J. COHEN. , *Refactoring Composite to Visitor and Inverse Transformation in Java*, July 2013, http://hal.inria.fr/hal-00652872

[56] A. LÈBRE, J. PASTOR, M. BERTIER, F. DESPREZ, J. ROUZAUD-CORNABAS, C. TEDESCHI, P. ANEDDA, G. ZANETTI, R. NOU, T. CORTES, E. RIVIÈRE, T. ROPARS. , *Beyond The Cloud, How Should Next Generation Utility Computing Infrastructures Be Designed?*, Inria, July 2013, n^o RR-8348, http://hal.inria.fr/hal-00854204

[57] N. TABAREAU, J.-J. SLOTINE. , *Contraction analysis of nonlinear random dynamical systems*, Inria, 2013, n^o RR-8368, 17 p. , http://hal.inria.fr/hal-00864079

Other Publications

[58] S. PEARSON, V. TOUNTOPOULOS, D. CATTEDDU, M. SÜDHOLT, R. MOLVA, C. REICH. , *Accountability for Cloud and Other Future Internet Services*, January 2013, The 4th IEEE Int. Conf. on Cloud Computing Technology and Science (CloudCom 2012), http://hal.inria.fr/hal-00778369

[59] M. SOZEAU, N. TABAREAU. , *Univalence for free*, 2013, http://hal.inria.fr/hal-00786589

References in notes

[60] M. AKŞIT, S. CLARKE, T. ELRAD, R. E. FILMAN (editors). , *Aspect-Oriented Software Development*, Addison-Wesley Professional, September 2004

[61] C. ALLAN, P. AVGUSTINOV, A. S. CHRISTENSEN, L. HENDREN, S. KUZINS, O. LHOTÁK, O. DE MOOR, D. SERENI, G. SITTAMPALAM, J. TIBBLE. *Adding trace matching with free variables to AspectJ*, in "ACM Conference on Object-Oriented Programming, Systems and Languages (OOPSLA)", R. P. GABRIEL (editor), ACM Press, 2005

[62] R. ALLEN, D. GARLAN. *A Formal Basis for Architectural Connection*, in "ACM Transactions on Software Engineering and Methodology", July 1997, vol. 6, n^{o} 3, pp. 213–49

[63] J. H. ANDREWS. *Process-Algebraic Foundations of Aspect-Oriented Programming*, in "Proceedings of the 3rd International Conference on Metalevel Architectures and Separation of Crosscutting Concerns", Lecture Notes in Computer Science, 2001, vol. 2192, pp. 187–209

[64] T. H. AUSTIN, C. FLANAGAN. *Multiple facets for dynamic information flow*, in "Proceedings of the 39th annual ACM SIGPLAN-SIGACT symposium on Principles of programming languages", New York, NY, USA, POPL '12, ACM, 2012, pp. 165–178, http://doi.acm.org/10.1145/2103656.2103677

[65] L. D. BENAVIDES NAVARRO, M. SÜDHOLT, W. VANDERPERREN, B. DE FRAINE, D. SUVÉE. *Explicitly distributed AOP using AWED*, in "Aspect-Oriented Software Development (AOSD)", ACM Press, March 2006, pp. 51-62

[66] G. S. BLAIR, G. COULSON, P. ROBIN, M. PAPATHOMAS. *An architecture for next generation middleware*, in "Proceedings of the IFIP International Conference on Distributed Systems Platforms and Open Distributed Processing", Springer-Verlag, 1998

[67] A. BRACCIALIA, A. BROGI, C. CANAL. *A formal approach to component adaptation*, in "Journal of Systems and Software", 2005

[68] S. CAPECCHI, I. CASTELLANI, M. DEZANI-CIANCAGLINI, T. REZK. *Session Types for Access and Information Flow Control*, in "CONCUR 2010 - Concurrency Theory, 21th International Conference, CONCUR 2010, Paris, France, August 31-September 3, 2010. Proceedings", P. GASTIN, F. LAROUSSINIE (editors), Lecture Notes in Computer Science, Springer, 2010, vol. 6269, pp. 237–252, http://dx.doi.org/10.1007/978-3-642-15375-4_17

[69] E. M. CLARKE, O. GRUMBERG, D. A. PELED. , *Model Checking*, The MIT PressCambridge, Massachusetts, 1999

[70] A. COLYER, A. CLEMENT. *Large-scale AOSD for Middleware*, in "Proceedings of the 3rd ACM Int. Conf. on Aspect-Oriented Software Development (AOSD), Lancaster", K. LIEBERHERR (editor), ACM Press, 2004, pp. 56–65

[71] F. DEREMER, H. H. KRON. *Programming-in-the-large versus programming-in-the-small*, in "IEEE Transactions on Software Engineering", 1976, vol. SE-2, n^{o} 2, pp. 80-86

[72] G. DECKER, O. KOPP, F. LEYMANN, M. WESKE. *BPEL4Chor: Extending BPEL for Modeling Choreographies*, in "IEEE International Conference on Web Services (ICWS 2007)", IEEE Computer Society, 2007, pp. 296–303

[73] E. W. DIJKSTRA. *On the role of scientific thought*, in "Selected Writings on Computing: A Personal Perspective", Springer-Verlag, 1974, pp. 60–66, Published in 1982

[74] R. DOUENCE, P. FRADET, M. SÜDHOLT. *A framework for the detection and resolution of aspect interactions*, in "Proceedings of the ACM SIGPLAN/SIGSOFT Conference on Generative Programming and Component Engineering (GPCE'02)", Lecture Notes in Computer Science, Springer-Verlag, October 2002, vol. 2487, pp. 173–188, http://hal.inria.fr/inria-00072153

[75] R. DOUENCE, P. FRADET, M. SÜDHOLT. *Trace-Based Aspects*, in "Aspect-Oriented Software Development", M. AKŞIT, S. CLARKE, T. ELRAD, R. E. FILMAN (editors), Addison-Wesley, 2004, pp. 201-218

[76] R. DOUENCE, O. MOTELET, M. SÜDHOLT. *A formal definition of crosscuts*, in "Proceedings of the 3rd International Conference on Metalevel Architectures and Separation of Crosscutting Concerns", Lecture Notes in Computer Science, Springer-Verlag, 2001, vol. 2192, pp. 170–186

[77] R. DOUENCE, D. LE BOTLAN, J. NOYÉ, M. SÜDHOLT. *Concurrent Aspects*, in "Proc. of the Int. ACM Conf. on Generative Programming and Component Engineering (GPCE)", ACM Press, October 2006, pp. 79-88

[78] H. FOSTER, S. UCHITEL, J. MAGEE, J. KRAMER. *Model-based Verification of Web Service Compositions*, in "Proceedings of the 18th IEEE Int. Conf. on Automated Software Engineering (ASE'03)", IEEE Computer Society, 2003, pp. 152–163

[79] A. FUGGETTA, G. P. PICCO, G. VIGNA. *Understanding Code Mobility*, in "IEEE Transactions on Software Engineering", May 1998, vol. 24, n^{o} 5, pp. 342–361

[80] K. HONDA, N. YOSHIDA, M. CARBONE. *Multiparty asynchronous session types*, in "Proceedings of the 35th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2008, San Francisco, California, USA, January 7-12, 2008", G. C. NECULA, P. WADLER (editors), ACM, 2008, pp. 273–284, http://www.doc.ic.ac.uk/~yoshida/multiparty/multiparty.pdf, http://doi.acm.org/10.1145/1328438.1328472

[81] G. KICZALES, E. HILSDALE, J. HUGUNIN, M. KERSTEN, J. PALM, W. G. GRISWOLD. *An Overview of AspectJ*, in "ECOOP 2001 — Object-Oriented Programming 15th European Conference, Budapest Hungary", Berlin, J. L. KNUDSEN (editor), Lecture Notes in Computer Science, Springer-Verlag, June 2001, vol. 2072, pp. 327–353, http://www.eclipse.org/aspectj/

[82] G. KICZALES. *Aspect Oriented Programming*, in "Proc. of the Int. Workshop on Composability Issues in Object-Orientation (CIOO'96) at ECOOP", July 1996, Selected paper published by dpunkt press, Heidelberg, Germany

[83] G. KICZALES, J. DES RIVIERES, DANIEL G. BOBROW. , *The Art of the Meta-Object Protocol*, MIT PressCambridge (MA), USA, 1991

[84] J. KIENZLE, R. GUERRAOUI. *AOP - Does It Make Sense? The Case of Concurrency and Failures*, in "16th European Conference on Object-Oriented Programming (ECOOP'2002)", Malaga, Spain, B. MAGNUSSON (editor), Lecture Notes in Computer Science, Springer-Verlag, 2002

[85] T. LEDOUX. *OpenCorba: a Reflective Open Broker*, in "ACM Meta-Level Architectures and Reflection, Second International Conference, Reflection'99", Saint-Malo, France, P. COINTE (editor), Lecture Notes in Computer Science, Springer-Verlag, July 1999, vol. 1616, pp. 197–214

[86] M. MCILROY. *Mass produced software components*, in "Mass produced software components", Garmish, Germany, P. NAUR, B. RANDELL (editors), NATO Science Committee, October 1968, pp. 138-155

[87] N. MEDVIDOVIC, R. N. TAYLOR. *A Classification and Comparison Framework for Software Architecture Description Languages*, in "IEEE Transactions on Software Engineering", January 2000, vol. 26, n^{o} 1, pp. 70-93

[88] M. MERNIK, J. HEERING, A. M. SLOANE. *When and How to Develop Domain-Specific Languages*, in "ACM Computing Surveys", December 2005, vol. 37, n^{o} 4, pp. 316-344

[89] L. MIKHAJLOV, E. SEKERINSKI. *A study of the fragile base class*, in "A study of the fragile base class", Brussels, Belgium, E. JUL (editor), Lecture Notes in Computer Science, July 1998, vol. 1445, pp. 355-382

[90] D. H. NGUYEN, M. SÜDHOLT. *VPA-based aspects: better support for AOP over protocols*, in "4th IEEE International Conference on Software Engineering and Formal Methods (SEFM'06)", IEEE Computer Society Press, September 2006

[91] O. NIERSTRASZ. *Regular Types for Active Objects*, in "Object-Oriented Software Composition", O. NIERSTRASZ, D. TSICHRITZIS (editors), Prentice Hall, 1995, chap. 4, pp. 99–121

[92] M. NISHIZAWA, S. CHIBA, M. TATSUBORI. *Remote Pointcut - A Language Construct for Distributed AOP*, in "Proceedings of the 3rd ACM Int. Conf. on Aspect-Oriented Software Development (AOSD), Lancaster", ACM Press, 2004

[93] D. L. PARNAS. *On the criteria for decomposing systems into modules*, in "Communications of the ACM", December 1972, vol. 15, n^{o} 12, pp. 1053-1058

[94] S. PEARSON. *Toward Accountability in the Cloud*, in "Internet Computing, IEEE", July-Aug. 2011, vol. 15, n^{o} 4, pp. 64-69, http://dx.doi.org/10.1109/MIC.2011.98

[95] F. PLASIL, S. VISNOVSKY. *Behavior Protocols for Software Components*, in "Transactions on Software Engineering", January 2002, vol. 28, n^{o} 9

[96] F. PUNTIGAM. *Coordination Requirements Expressed in Types for Active Objects*, in "ECOOP'97—Object-Oriented Programming", M. AKŞIT, S. MATSUOKA (editors), Lecture Notes in Computer Science, Springer-Verlag, 1997, vol. 1241, pp. 367–388

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

[98] B. C. SMITH. , *Reflection and Semantics in LISP*, Xerox Palto Alto Research CenterPalo Alto, 1984, n^{o} P84-00030

[99] S. SOARES, E. LAUREANO, P. BORBA. *Implementing distribution and persistence aspects with AspectJ* , in "Proceedings of the 17th ACM conference on Object-oriented programming, systems, languages, and applications (OOPSLA-02)", C. NORRIS, J. J. B. FENWICK (editors), ACM SIGPLAN Notices, ACM Press, November 4–8 2002, vol. 37, 11, pp. 174–190

[100] S. SUNDARESWARAN. *Ensuring Distributed Accountability for Data Sharing in the Cloud*, in "Dependable and Secure Computing", 2012, vol. 9, http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6165313

[101] R. J. WALKER, K. VIGGERS. *Implementing Protocols via Declarative Event Patterns*, in "Proceedings of the ACM SIGSOFT International Symposium on Foundations of Software Engineering (FSE-12)", ACM Press, 2004, pp. 159 - 169

[102] M. WAND, G. KICZALES, C. DUTCHYN. *A Semantics for Advice and Dynamic Join Points in Aspect-Oriented Programming*, in "ACM Transactions on Programming Languages and Systems (TOPLAS)", 2004, vol. 26, n^{o} 5, pp. 890–910

[103] D. M. YELLIN, R. E. STROM. *Protocol specifications and component adaptors*, in "ACM Transactions of Programming Languages and Systems", March 1997, vol. 19, n^{o} 2, pp. 292–333

[104] A. VAN DEURSEN, P. KLINT, J. VISSER. *Domain-Specific Languages: An Annotated Bibliography*, in "ACM SIGPLAN Notices", June 2000, vol. 35, n^{o} 6, pp. 26-36