INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Project-Team Adam

Adaptive Distributed Applications and Middleware

Lille - Nord Europe

THEME COM

Activity Report

2008

Table of contents

1. Team 1
2. Overall Objectives 2
2.1. Introduction 2
2.2. Challenges 2
2.2.1. Life-cycle of Adaptive Systems 3
2.2.2. Multi-paradigm Applications 3
2.2.3. On Supporting Multi-scale Environment 3
2.3. Highlights of the year 3
3. Scientific Foundations 4
3.1. Introduction 4
3.1.1. Aspect-Oriented Software Development (AOSD) 4
3.1.2. Component-Based Software Engineering (CBSE) 5
3.1.3. Context-Aware Computing (CAC) 5
3.2. Two Research Directions 6
3.2.1. Adaptable Component Frameworks for Middleware 6
3.2.2. Distributed Application Design for Adaptive Platforms 6
4. Application Domains 7
4.1.1. Electronic Commerce 7
4.1.2. Embedded Systems 7
4.1.3. Health Care Information Systems 8
4.1.4. Information Systems for Terrestrial Transport 8
5. Software 8
5.1. Introduction 8
5.2. CALICO 8
5.3. Fractal Deployment Framework 9
5.4. Fractal 9
5.5. FraSCAti 10
5.6. Hulotte 11
5.7. Soleil 11
5.8. Spoon 11
6. New Results 12
6.1. Components for Embedded and Real-Time Systems 12
6.2. Deployment of distributed and heterogeneous software systems 13
6.3. Software Engineering for Ambient Intelligence 14
6.3.1. COSMOS 14
6.3.2. Annotation-Oriented Programming 14
6.3.3. Context-Aware Modelling 15
6.3.4. Adaptive Middleware for Context-Awareness 15
6.4. Software Evolution 15
6.4.1. Static and Dynamic Checking of Component Interactions 15
6.4.2. Detection of Design Defects 16
6.5. INRETS-LEOST 16
7. Other Grants and Activities 17
7.1. National Initiatives 17
7.1.1. ANR ARA REVE 17
7.1.2. ANR TLog FAROS 17
7.1.3. ANR TLog Flex-eWare 17
7.1.4. ANR TLog SCOrWare 17
7.1.5. INRIA ADT galaxy 18

7.1.6. INRIA ARC BROCCOLI 18
7.1.7. Trade cluster CAPPUCINO 18
7.1.8. Trade cluster ICOM 19
7.1.9. Trade cluster MIND 19
7.2. European Initiatives 19
7.2.1. ERCIM Working Group Software Evolution 19
7.2.2. IAP MoVES 19
7.2.3. INRIA associate team CALA 20
7.2.4. IST FP6 NoE AOSD-Europe 20
7.2.5. IST FP7 SOA4All Integrated Project 20
7.3. International Initiatives 21
7.3.1. OW2 21
7.3.2. University of Montreal 21
8. Dissemination **21**
8.1. Visiting researchers 21
8.2. Scientific Community Animation 21
8.2.1. Examination Committees 21
8.2.2. Journals, Conferences, Workshop 22
8.3. Scientific and Administrative Responsibilities 24
8.4. Teaching 24
8.5. Miscellaneous 24
9. Bibliography **25**

ADAM is a joint project-team between INRIA, CNRS, and Université des Sciences et Technologies de Lille (USTL), via the Computer Science Laboratory of Lille : LIFL (UMR 8022).

1. Team

Research Scientist

Alexandre Bergel [Research Associate (CR2), INRIA, until June 30th 2008]
Stéphane Ducasse [Research Director (DR2), INRIA, until June 30th 2008, HdR]
Christophe Gransart [Research Associate (CR1), INRETS, external collaborator]
Philippe Merle [Research Associate (CR1), INRIA, vice-head of project-team]

Faculty Member

Xavier Blanc [Associate Professor, UPMC, secondment INRIA, since November 1st 2008]
Laurence Duchien [Professor, USTL, secondment INRIA, project-team leader, HdR]
Jean-Marc Geib [Professor, USTL, external collaborator, HdR]
Anne-Françoise Le Meur [Associate Professor, USTL]
Romain Rouvoy [Associate Professor, USTL, since September 1st 2008]
Lionel Seinturier [Professor, USTL, HdR]

Technical Staff

Pierre Carton [Expert Engineer, INRIA, CAPPUCINO project, since October 1st 2008]
Yann Davin [Development Engineer, INRIA, INRIA ODL]
Christophe Demarey [Research Engineer, INRIA, Galaxy ADT, since July 1st 2008]
Nicolas Dolet [Expert Engineer, INRIA, ANR TLog SCOrWare]
Julien Ellart [Expert Engineer, INRIA, ICOM Project, since October 1st 2008]
Areski Flissi [Research Engineer, CNRS, until August 31th 2008]
Damien Fournier [Expert Engineer, INRIA, ANR TLog SCOrWare]
Frédéric Loiret [Expert Engineer, INRIA, ANR TLog Flex-eWare]
Nicolas Pessemier [Expert Engineer, INRIA, CAPPUCINO project, since December 1st 2008]

PhD Student

Hani Abdeen [KHAN grant, until June 30th 2008]
Aurélien Bocquet [INRETS / Région Nord/Pas-de-Calais grant]
Jérémy Dubus [MERT grant, until June 30th 2008]
Guillaume Dufrêne [ANR TLog FAROS grant, until March 31th 2008]
Gabriel Hermosillo [INRIA CORDIS grant, since October 1st 2008]
Guillaume Libersat [MERT grant, since October 1st 2008]
Naouel Moha [University of Montreal, co-supervision, until August 31th 2008]
Carlos Francisco Noguera Garcia [IST AOSD Network of Excellence grant]
Carlos Andres Parra Acevedo [INRIA / Région Nord/Pas-de-Calais grant]
Aleš Plšek [INRIA CORDIS grant]
Daniel Romero [EIFFEL grant]
Mathieu Suen [CNRS/DGA grant, until June 30th 2008]
Alban Tiberghien [INRIA / Région Nord-Pas de Calais grant, since October 1st 2008]
Guillaume Waignier [MERT grant]

Post-Doctoral Fellow

Prawee Sriplakich [INRIA grant, until August 31th 2008]

Visiting Scientist

Rubby Casallas [University of Los Andes, Bogota, June 2008]
Angela Lozano [PhD student, Open University, United Kingdom, since July 1st 2008]
Michal Malohlava [PhD student, Charles University, Prague, June–September 2008]
Rafael Leaño [Master student, University of Los Andes, Bogota, September–December 2008]

Administrative Assistant

Corinne Davoust [INRIA, until August 31th 2008]
Malika Debuysschere [INRIA, since September 1st 2008]

2. Overall Objectives

2.1. Introduction

Keywords: *Adaptation*, *Architecture Description Language (ADL)*, *Aspect-Oriented Software Development (AOSD)*, *Attribute-Oriented Programming*, *Component-Based Software Engineering (CBSE)*, *Context-Aware Computing (CAC)*, *Distributed Applications*, *Middleware*, *Model-Driven Engineering (MDE)*, *Separation of Concerns (SoC)*, *Software Architecture*.

With the increasing need of self-managed systems and the emergence of multi-scale environments, software developers have to cope with variability. Software must be developed to be adapted and reconfigured automatically on heterogeneous platforms in accordance with the unavoidable evolution of information and communication technologies. Therefore, the adaptation is now considered as a first-class problem that must be taken into account throughout the entire software life-cycle.

An *adaptive system* is a software-intensive system that can adjust and respond to changes in its environment, evolving requirements, removal of obsolete technologies or introduction of new technologies, and new knowledge. The objective of the ADAM project-team is to provide a set of paradigms, approaches and frameworks based on advanced software engineering techniques, such as *Component-Based Software Engineering* (CBSE), *Aspect-Oriented Software Development* (AOSD), or *Context-Aware Computing* (CAC), to build distributed adaptive software systems evolving in multi-scale environments and to take into account the adaptation all along the software life-cycle. We propose to follow two research directions: the definition of adaptable component frameworks for middleware and the design of distributed applications for adaptive platforms.

2.2. Challenges

Software, whatever its objectives, has become a key element in all sectors of the economic activity. It must cope more and more with the variability of computing platforms and must be usable in various environments, in accordance with the unavoidable evolution of communication and information technologies. Software must now be developed to be adapted and reconfigured automatically on heterogeneous platforms that base themselves on different network and system paradigms, such as volatile or reliable connection, ambient or distributed reference, or energy-constrained or not. Depending on the application domains, software interacts with its execution context—*i.e.*, the end-users, their local environment, and the execution platform, which involve adaptation. Therefore, a system will not be built in only one step but has to be seen as a composition and evolution of existing systems or sub-systems. These systems are defined by software engineering paradigms, such as components, services or aspects. Moreover, these adaptations can be performed at design, compile or run time. Consequently, the entire life-cycle, from design to deployment and execution of software, has to be considered not only when the execution context changes, but also when functionalities are added or withdrawn in accordance with user needs. Thus, software life-cycle has to be defined to support the expression of adaptation at each of its levels. Consequently, there is a strong requirement to consider *adaptation* as a first-class problem in the development and execution of software.

As such, an *adaptive system* is a software-intensive system (see software-intensive systems report [73]) that can adjust and respond to changes in its environment, evolving requirements, removal of obsolete technologies or introduction of new technologies, and new knowledge [62]. It will have multiple applications considering technical changes, such as heterogeneous complex and large systems, embedded software systems for wireless networks, and in application domains, such as transport, home, communication, health, education, and service-oriented business applications based on Web Services.

Supporting the entire life-cycle of adaptive systems raises several challenges, from elaborating methods and tools for system design to the definition of middleware infrastructures to support the target system on heterogeneous and dynamic platforms. In current software engineering, paradigms, such as *Component-Based Software Engineering* (CBSE) and *Aspect-Oriented Software Development* (AOSD), allow the expression of software adaptation by composition or by weaving. Moreover, various abstractions for modeling structural and behavioral adaptation of applications in response to changes in the existing environment have been studied recently from resource-aware programming to adaptive software architecture. The ADAM challenges will be to study the entire life-cycle of adaptive multi-paradigm systems for multi-scale environments.

2.2.1. Life-cycle of Adaptive Systems

The first challenge will be to master the adaptation of distributed middleware and applications in accordance with the evolution of runtime constraints, while following this evolution from the design to the run time. Although occurring at different levels, adaptation must be managed consistently across the whole life-cycle of a system. In particular, in a MDE context, if a model at some level of the software life-cycle changes, then this change may have an impact on other models at the same level that are related to it, as well as on related models in earlier and later phases of the life-cycle. To reach this challenge, techniques and formalisms are needed to assess the impact of such changes, to synchronize models within and between different phases, to co-evolve models and implementations and to adapt the implementation at run time. Because of the mobility and the high dynamicity in adaptive systems, these tools and approaches should be more effective.

2.2.2. Multi-paradigm Applications

The second challenge will be to develop and process applications written with different software paradigms at the same time and to allow their co-existence. Developing adaptable applications requires the use of good practices on software engineering. However, in such context, applications are often composed of multiple sub-systems and services written in different languages and supported by systems using several paradigms. Implementation-level solutions, like AOP [58], patterns [56], generative programming [54], or reflective approaches [57], [72] are already employed and co-exist in nowadays software. However, in order to better support adaptation, application must be specified and developed with different paradigms, such as CBSE, AOSD, or services and to be deployed on different runtime platforms, such as J2EE, JBI, FRACTAL, SCA or on new technological platforms, such as FAC [68] and AOKell [70]. For reaching this challenge, we will consider heterogeneity on paradigms, their unification on generic models that can be projected on specific platforms or their co-existence through interconnection of platforms.

2.2.3. On Supporting Multi-scale Environment

Finally, our research context considers multi-scale networks and runtime environment properties in the design and the execution of middleware and applications. We do not claim to propose advances in the fields of the ambient computing core or the grid core, but we take into account their definition and their properties as interesting adaptive systems on which we will test our multi-paradigm approaches and our life-cycle steps. They are for us experimentation fields. Environments, such as sensor networks, wireless networks for embedded systems, the Internet, or environments for grid computing do not have the same properties in terms of data and computation capacities. Considering data, sensor or mobile networks will manipulate small amounts of data, but with a large distribution and some inconsistencies. Internet networks and grid computing, on the other hand, will operate on large quantities of data, which is mostly homogeneous, consistent, and centralized. Computation capacity must also be considered. PDAs, phones or sensors have small capacities in terms of battery, memory, or computation units, consequently the application and its run-time platform design will be designed and distributed according to these capacities.

2.3. Highlights of the year

- **CALICO** is a Component AssembLy Interaction Control FramewOrk, which enables software architects to first specify component-based architectures and contractual properties, and then check the consistency of the component assembly specifications [35]. The validation of the specifications may be performed at execution time when static verification is not sufficient [34]. CALICO is generic, it supports different target component platforms, such as Fractal, OpenCCM, and OpenCom, and different kinds of constraints, such as structural constraints expressed with OCL or behavioral constraints. We have implemented CALICO in Java and used the *Eclipse Modeling Framework* (EMF) to enable the editing and manipulation of the various description models offered by the framework to describe an architecture. CALICO is freely available at http://calico.gforge.inria.fr. For more details, see sections 5.2 and 6.4. Our work on CALICO partially falls within the context of the RNTL project FAROS (cf. section 7.1), which involves EDF R&D, France Télécom, Alicante, IRISA, I3S and LIFL. This project addresses issues related to the safe integration of services in service-oriented architectures and has been presented at the Research Projects Symposium of MODELS'08.

- **FraSCAti** is our open source runtime platform for supporting *Service Component Architecture* (SCA) specifications. SCA is an initiative supported by both OASIS and Open SOA international industrial consortiums for unifying *Service Oriented Architectures* (SOA) and *Component-Based Software Engineering* (CBSE). FraSCAti is one of the main results of the ANR TLog SCOrWare project (cf. section 7.1) and will be hosted by the OW2 consortium at http://forge.objectweb.org/projects/frascati. As far as we know, FraSCAti is the first original result to concretely demonstrate that the FRACTAL component model is open and flexible enough to implement different component models on top of it. For more details, see section 5.5.
- **Soleil** is a component-based framework for developing systems which conform to the Sun *Real-Time Specification for Java* (RTSJ). The key motivation for the project is to shield developers from the complexities of RTSJ related code and allow them to focus only on development of the application logic. A methodology and a component-based programming model are proposed. A code generation framework is available for producing an executable system where a tradeoff can be made between reconfigurability and performance [29], [30]. The Soleil framework is presented in more details in section 5.7.

3. Scientific Foundations

3.1. Introduction

Mastering the complexity of the adaptation will be possible by self-adaptive knowledge systems that can obtain and adapt knowledge from different sources. In order to cope with this objective, we will consider software paradigms that will help us in our approach at the various levels of our life-cycle of adaptive systems, but also in the tools themselves for their composition. We will also study these paradigms in the middleware and application design in order to extend them and to have a better understanding. These extensions will be formalized as much as possible.

3.1.1. Aspect-Oriented Software Development (AOSD)

In modern software engineering, language constructs are classified according to how they recombine partial solutions for subproblems of a problem decomposition. Some constructs (*e.g.*, methods and classes) recombine partial solutions using classic hierarchical composition. Others recombine the partial solution using what is known as crosscutting (a.k.a. aspectual) composition. With crosscutting composition, two partial solutions (called aspects) are woven into each other in a way that is dictated by so-called pointcut languages. The necessity of crosscutting composition is the main motivation for the AOSD [51], [59] paradigm. The challenge will be first to study new expressive pointcut languages in order to have a better description of composition locations in adaptable software. The second objective will be to extend and to integrate new techniques of weaving at design time, but also at run time in order to compose software safely. The third objective will be to go beyond simple aspects as persistence and logging services. We plan to study complex aspects such as transactions or replication and to control their weaving in order to master the evolution of complex software.

3.1.2. Component-Based Software Engineering (CBSE)

In a post-object world [55], software components [61] are, with other artifacts such as aspects, one of the approaches that aims at overcoming the limitations of objects and providing more flexibility and dynamicity to complex applications. For that, software components present many interesting properties, such as modularity, encapsulation, and composability. Yet, many different component models and frameworks exist. A survey of the literature references more than 20 different models (including the most well-known, such as EJB [50] and CCM [49]), but the exact number is certainly closer to 30. Indeed, each new author proposes a model to address her/his own need related to a particular execution environment (from grid computing to embedded systems) or the technical services (from advanced transactions to real-time properties), which must be provided to the application components. These different component models seldom interoperate and their design and implementation are never founded on a common ground. The research challenge that we identify is to define and implement solutions for adaptive software components. These components will be adaptive in the sense that they will be able to accommodate execution environments of various granularities (from grid computing, to Internet-based applications, to mobile applications, to embedded systems) and incorporate on-demand different technical services. This challenge will be conducted by designing a micro-kernel for software components. This micro-kernel will contain a well-defined set of core concepts, which are at the root of all component models. Several concrete software component models will then be derived from this micro-kernel.

3.1.3. Context-Aware Computing (CAC)

In adaptive systems, the notion of "*context*" becomes increasingly important. For example, mobile devices sense the environment they are in and react accordingly. This is usually enabled by a set of rules that infer how to react given a certain situation. In the Ambient/Ubiquitous/Pervasive domain[1], CAC is commonly referred to as the new paradigm that employs this idea of context in order to enmesh computing in our daily lives [63]. Many efforts exist today that focus on human-computer interaction based on context. On the other hand, computational models, middleware, and programming languages are being developed to take the inherent characteristics of multi-scale environments into account, such as connection volatility, ambient resources, etc. An important challenge is to bridge the gap between the domain level and the computational level. The former is concerned with the expected behavior of the system from a user's viewpoint, such as how and when a system responds to changes in the context, when information can be made public, etc. On the other hand, the computational level deals with the inherent and very stringent hardware phenomena of multi-scale environments. Nevertheless, both levels have to coexist: the computational level needs to be steered by the concepts, behavior and rules which exist at the domain level, whereas the domain needs to adapt to the particulars of the ever changing environment that is monitored and managed by the computational level. In order to address this challenge, we first intend to investigate representations at the domain level of concepts such as user profile, local positioning information and execution context [71]. Furthermore, a mapping has to be devised between these concepts and generic concepts at the computational level, the latter being as independent as possible from concrete platforms or languages. This mapping has to be bidirectional: the computational level needs to be steered by the concepts, behavior and rules that exist at the domain level, whereas the domain needs to adapt to the particulars of the ever-changing environment that is monitored and managed at the computational level. Furthermore, the mapping has to be dynamic since the changes have to be propagated between the levels at run time. An explicit domain level is not only useful for bridging the aforementioned gap, but also for designing and developing open task-specific languages at the domain level, which allow users to dynamically adapt the behavior of the applications in multi-scale environments in well-defined ways.

We will base the design approach of the future implementation prototype on MDE. The goal of MDE [69] consists of developing, maintaining and evolving complex software systems by raising the level of abstraction from source code to models. The latter is in our case the domain level, which will be connected to the computational level by means of MDE techniques. One added benefit of MDE is that it provides means for managing model inconsistencies.

[1] These terms are more or less equivalent.

3.2. Two Research Directions

We propose to follow two research directions to foster software reuse and adaptation. The first direction, that could be coined as the spatial dimension of adaptation, will provide middleware platforms to let applications be adapted to changing execution contexts. The second direction, the so-called temporal dimension of adaptation, will provide concepts and artifacts to let designers specify evolvable applications.

3.2.1. Adaptable Component Frameworks for Middleware

As a cornerstone of next generation software, adaptation is a property which must be present throughout the entire life cycle, from design to execution. We develop then a vision where adaptation is not only a property that is desirable for end-user applications, but also for the middleware platform that executes these applications. Until now, middleware is a rather specialized activity where each new environment forces the development of a corresponding platform, which is specific to the given environment. This has led to a large number of platforms (from Web Services, to EJB, to CORBA, to ad hoc middleware for embedded systems). Although at a high level, solutions for communication interoperability often exist between these platforms, they stay loosely coupled and separated. Furthermore, the concepts which are at the core of these platforms and their architectures are too different to allow, for example, sharing technical services.

The research challenge that we propose here is to define and develop middleware and associated services which could be adapted to a broad range of environments from grid computing, to Internet-based applications, to local networks, to mobile applications on PDA's and smart phones, to embedded systems. The benefits of that are twofold. First, it enables the easier deployment of mobile applications in different environments by taking advantage of the common ground provided by adaptable middleware. Second, middleware is a rapidly changing domain where new technologies appear frequently. Yet, up to now, each new technological shift has imposed a complete re-development of the middleware. Having a common ground on which middleware is built would help in such transitions by fostering reuse. In terms of industrial output, the impact of these results will also be helpful for software editors and companies to adapt their products more rapidly to new and emerging middleware technologies.

This research challenge has close links with MDE and product line families. We believe that the added value of our proposal is to cover a more integrated solution: we are not only interested in middleware design with MDE technologies, but we also wish to integrate them with software component technology and advanced programming techniques, such as AOP. We will then cover a broad spectrum of middleware construction, from design (MDE) to implementation (CBSE) to application development (AOP).

3.2.2. Distributed Application Design for Adaptive Platforms

Considering adaptation in the first design steps of an application allows for the preparation and follow-up of it during the entire life-cycle. As mentioned previously, some software paradigms help already in the design and the development of adaptable applications. AOSD proposes separation of concerns and weaving of models in order to increase the mastering and the evolution of software. MDE consists of evolving complex software systems by raising the level of abstraction from source code to models. Several programming approaches, such as AOP or reflective approaches, have gained in popularity to implement flexibility. Other approaches, such as CBSE, propose compositional way for reuse and compose sub-systems in the application building. Finally, context-aware programming for mobile environment proposes solutions in order to consider context evolution. Overall, the objective of these approaches is to assist the development of applications that are generic and that can be adapted with respect to the properties of the domain or the context.

The research challenge that we propose here is similar to static points of variation in product line families. We plan to study dynamic points of variation in order to take into account adaptation in the first design steps and to match this variation. The first research challenge is the introduction of elements in the modeling phase that allow the expression of properties related to evolution. These properties must make it possible to build safe and dynamic software architectures. We wish to express and validate properties in the entire software life cycle. These properties are functional, non-functional, static, behavioral, but also qualitative properties. We also want to be able to check that all the properties are present, that the obtained behavior is one which is expected, and that the quality of service is not degraded after the addition or the withdrawal of functionalities. We will base our approach on the definition of contracts expressed in various formalisms (*e.g.*, first order logic, temporal logic, state automata) and we will propose a composition of these contracts.

The second challenge will be to implement design processes that maintain coherence between the various stages of modeling in a MDE approach of the applications, as well as maintaining coherence between the phases of modeling and implementation. To do so, we will design and implement tools that will allow traceability and checking of coherence between models, and between models and the execution.

Finally, we will introduce context information in the development process. At the modeling level, we will represent concepts, behavior and rules of adaptive systems to express adaptation abstraction. These models will be dynamic and connected to implementation levels at the computational level and they will consider context knowledge. The goal is to bridge the gap between the computational level and the domain level in adaptive systems by synchronization of models and implementations, but also by representation of such common knowledge.

4. Application Domains

4.1. Application Domains

Keywords: *ambient intelligence*, *electronic commerce*, *embedded systems*, *health care information systems*, *terrestrial transport information systems*.

The ADAM project-team targets the software engineering of adaptive service-oriented applications and middleware. The application domain covered by ADAM is broad and spans from distribution applications to middleware. In all these cases, adaptability is the property which is sought: applications and middleware must be adaptable to new execution contexts, they must react to changes in the environment and they must be able to discover and integrate new services.

The ADAM project-team produces software and middleware building blocks. This explains why the application domain is broad, yet targeting applications where adaptability is the key requirement. This includes electronic commerce, embedded systems, health care information systems, and terrestrial transport information systems. These domains are in direct relation with our currently funded activities. They act as testbeds for the solutions that we propose in terms of middleware services, middleware platforms, runtime kernels, component libraries, languages design or domain modeling.

4.1.1. Electronic Commerce

Applications in the domain of electronic commerce are by essence distributed. They involve many different participants with heterogeneous information systems which can not be changed. The challenge is then to provide an adaptation layer to be able to compose and let these systems interoperate. In the context of the ANR TLog SCOrWare, the IST SOA4ALL and the CAPPUCINO projects (cf. sections 7.1 and 7.2), our activities in this domain will aim at supporting service-oriented architectures. We want to have adaptive architectures which can be composed and orchestrated seamlessly. In this domain, the business relationship with customers is vital and many different usage scenarios must be supported. Customers are roaming, and the services must be kept operational across different devices. This puts some constraints on the server tier where technical services must be adapted to manage, for example, long lasting transactions. The application server infrastructure must then provide a support for adapting technical services.

4.1.2. Embedded Systems

Embedded systems is a domain where adaptation is a key requirement. The design and the implementation of modern embedded software uses advanced software engineering techniques such model-driven development or software component frameworks. In this domain, we are involved in several projects, such as the ANR TLog Flex-eWare, the ANR ARA REVE, and the trade cluster MIND (cf. section 7.1). Several challenges must be addressed here. For example, when a model-driven developed application is adapted, designers have to ensure that the models and the operational level are kept synchronized. The co-evolution of these two levels is one of the challenges that we are addressing. A second challenge is related to software components which need to be customized in order to fit the requirements imposed by constrained environments. It is, for example, a matter of providing component frameworks which can accomodate various granularities of services.

4.1.3. Health Care Information Systems

Health care information systems is a third application domain in which the ADAM project-team is involved, for example through demonstrators which will be implemented in the context of the ANR TLog FAROS project (cf. section 7.1). The challenge is here to provide a distributed infrastructure where information will be available to the medical staff wherever they are. This imposes to be able to provide this information on many different devices (from high resolution screens to embedded devices on the scene of an accident), while ensuring the privacy of the medical data of a patient (several level of data access must be granted depending on the categories of medical staff). Given the vital role of such an information system, we want to provide guarantees that the services will be highly available and trustworthy. We envision to provide a service-oriented architecture which will be extended to support software contracts and multi-scale environments.

4.1.4. Information Systems for Terrestrial Transport

Information systems for terrestrial transport is also a domain that we are relying on to apply our research activities in accordance with the ANR ARA REVE project and the INRETS collaboration (cf. section 7.1). Applications are here characterized by frequent disconnections, poor quality network links, and high mobility. We want to provide an infrastructure where the technical services, and among others the communication services, can be adapted to support new requirements. One of the path that we propose to investigate is to include such a scenario in the general context of the adaptiveness of component frameworks.

5. Software

5.1. Introduction

We intend to develop a number of software to evaluate and validate our solutions. We will complete our development by experimentation, benchmarks and deployment in multi-paradigm platforms. We list our actual software that we intend to continue and to extend in the ADAM project-team.

5.2. CALICO

Keywords: *component software architecture*.

Participants: Laurence Duchien, Anne-Françoise Le Meur, Prawee Sriplakich, Guillaume Waignier [correspondant].

Following our previous work on component-based architecture design, we have proposed CALICO.

CALICO aims to support the design and validation of component-based assemblies. It enables software architects to specify their architectures as models, and to analyse them with respect to application constraints and platform constraints. Analysis may be static or dynamic depending on whether runtime information are needed to check the compatibility of the specified constraints. CALICO can automatically instantiate the running system from the architecture specification, as well as the system code instrumentation that is required to perform constraint check at execution time.

CALICO is generic, it supports different target component platforms, such as FRACTAL, OpenCCM and OpenCom, and different kinds of constraints, such as structural constraints expressed with *Object Constraint Language* (OCL) or behavioral constraints.

We have implemented CALICO in Java. The *Eclipse Modeling Framework* (EMF) is used for generating an editor, which enables architects to visualize or edit the architecture description models. Consequently, architects can graphically manipulate their architecture and do not need to directly use the CALICO metamodels. EMF enables also the generation of model repository code, which offers programmatic support to manipulate the architecture description models. The EMF model repository code forms the core of all CALICO tools that require manipulating the models. Furthermore, the instrumentation of the system code has implemented with Spoon [67].

Calico has been developped in the context of Guillaume Waignier's PhD thesis and the postdoctoral internship of Prawee Sriplakich. CALICO is an open source software available at http://calico.gforge.inria.fr.

5.3. Fractal Deployment Framework

Keywords: *Component*, *Deployment*, *Fractal*, *Middleware*.

Participants: Nicolas Dolet, Jérémy Dubus, Areski Flissi, Damien Fournier, Philippe Merle [correspondant].

The work on FRACTAL Deployment Framework (FDF) is a part of the Jérémy Dubus's PhD thesis [12]. FDF is a component-based software framework to facilitate the deployment of distributed and heterogenous software systems. FDF implements the virtual machine of the DeployWare framework (cf. section 6.2). FDF supports any kind of deployment activities like uploading, installing, configuring, starting, stopping and uninstalling any kind of software stacks. For that, FDF is composed of a high level deployment language, a library of deployment components, and a set of end-user tools.

The FDF language is a kind of high level scripting language allowing end-users to describe their deployments (*i.e.*, hosts of the target system and software to deploy on them). This language relies on a library of deployment components wrapping various file transfer protocols (*e.g.*, FTP, HTTP, SCP), remote access protocols (*e.g.*, TELNET, SSH), Unix and Windows shells, Internet-related notions (*e.g.*, hostname, port, host), and software (*e.g.*, middleware, services, daemons, application servers, application components). FDF also provides a graphical user interface allowing end-users to load their deployment descriptions, execute and manage them.

Currently, FDF supports the deployment of SOA-based systems (BPEL processes, ActiveBPEL and OW2 Orchestra BPEL engines, SCA and both FraSCAti and Apache Tuscany runtimes, JBI components and OW2 PEtALS distributed ESB), JEE-based systems (EAR, WAR, EJB and RAR, application servers, such as Apache Geronimo, JBoss, OW2 JOnAS or SUN GlassFish, OW2 JASMINe administration tool), Web-based systems (Apache Tomcat servlet container, HTTPd server), CORBA-based systems (CORBA middleware, OW2 OpenCCM, business components, and applications), OW2 FRACTAL-based systems (Julia, FRACTAL ADL, and RMI), Java-based systems (Apache Ant, SUN JRE, JamVM for Linux PDA), database systems (MySQL), network services (OpenLDAP server), and operating systems (QEMU).

In [23], we have shown how FDF can be self-deployed on Grid'5000, the french grid infrastructure dedicated to computer science research, in order to deploy large-scale distributed software systems.

FDF is designed with the OW2 FRACTAL component model, see section 5.4, and is implemented on top of its Java-based tools (Julia, Fraclet, FRACTAL ADL, RMI, and Explorer).

FDF is LGPL open source software hosted by the INRIA Forge at http://fdf.gforge.inria.fr, and is being developed since 2006. Known usages of FDF by external people include the deployment of the Jade autonomous platform (INRIA SARDES project), and the integration of FDF into the JOnAS JEE application server (http://jonas.ow2.org) and the JASMINe SOA platform management tool (http://jasmine.ow2.org) developed by Bull SA, and the PEtALS JBI implementation developed by EBM WebSourcing (http://petals.ow2.org). All these open source projects are hosted by the OW2 consortium. FDF is composed of around 25.000 lines of Java code and 21.000 lines of FRACTAL ADL descriptions.

5.4. Fractal

Keywords: *Component model*.

Participants: Yann Davin, Philippe Merle [correspondant], Romain Rouvoy, Lionel Seinturier.

FRACTAL is a modular, extensible and programming language agnostic component model that can be used to design, implement, deploy and reconfigure systems and applications, from operating systems, middleware platforms to graphical user interfaces [52], [53]. FRACTAL has been designed by both INRIA and France Telecom R&D [46]. FRACTAL is also an open source software project hosted by the OW2 international consortium and is available at http://fractal.ow2.org.

Philippe Merle is the leader of the OW2 FRACTAL open source project. The ADAM project-team actively contributes to this project, and more specifically on the following modules:

- **AOKell** is an aspect-oriented implementation of the FRACTAL component model [70],
- **Fraclet** is an attribute-oriented programming model enabling the rapid development of FRACTAL components [19],
- **Fractal ADL** is the extensible architecture definition language for FRACTAL associated to an open FRACTAL component-based toolchain,
- **Fractal Distribution** is the module to produce packaged releases of the FRACTAL project,
- **Fractal Documentation** is the module to produce the whole documentation of the FRACTAL project,
- **Fractal Eclipse Plugin** is a plugin to create FRACTAL projects within the Eclipse IDE,
- **Fractal Explorer** is a framework to build graphical consoles to introspect and manage FRACTAL components dynamically at runtime,
- **Fractal RMI** is a binding framework for distributed FRACTAL components,
- **Juliac** is a FRACTAL ADL backend to generate and compile all the source code needed to run FRACTAL applications,
- **Koch** is an implementation of the FRACTAL component model where components have a component-based control membrane.

5.5. FraSCAti

Keywords: *Component*, *Fractal*, *Middleware*, *Service Component Architecture*, *Service Oriented Architecture*.

Participants: Pierre Carton, Christophe Demarey, Nicolas Dolet, Damien Fournier, Philippe Merle [correspondant], Lionel Seinturier.

FraSCAti is a runtime platform for the *Service Component Architecture* (SCA) component framework. SCA is an initiative for unifying *Service Oriented Architectures* (SOA) and *Component-Based Software Engineering* (CBSE). SCA is supported by the Open SOA consortium, which includes partners, such as IBM, Oracle, Sun and Iona, and will become a set of standardized OASIS's specifications. FraSCAti includes Tinfi, which provides a SCA personality for the FRACTAL component model. Thanks to the openness of this latter model, the necessary code elements (so called *controllers* and *membranes*) have been designed and developed to customize FRACTAL and to end up with components owning both a FRACTAL personality and a SCA personality. As far as we know, this result is original and is the first one to concretely demonstrates that FRACTAL is open and flexible enough to implement different component personalities. Moreover, Tinfi reuses the aspect-oriented concepts defined in FAC [16] for component-based programming and allows integrating smoothly non functional concerns (so called intents and policy sets in SCA terms). FraSCAti and Tinfi have been implemented by reusing modules developed in the context of the FRACTAL project, and among others, the Juliac FRACTAL compiler. FraSCAti and Tinfi are developed in the context of the ANR TLog SCOrWare project (cf. section 7.1).

FraSCAti is LGPL open source software, available currently at http://www.scorware.org, and hosted by the OW2 consortium since November 2008 at http://frascati.ow2.org.

5.6. Hulotte

Keywords: *Real-Time Java*.

Participants: Frédéric Loiret [correspondant], Michal Malohlava, Aleš Plšek, Lionel Seinturier.

Component-based technologies are widely used in various application domains (from grid computing to embedded systems) for which the functional needs are heterogeneous. To cope with such an heterogeneity, the execution infrastructures that provide the basic services required by the applications should be adapted to these targeted domains. Within ADAM, various works have been conducted to carry out this adaptation needed at infrastructure level, in particular :

- by using generative approaches allowing the generation of the sub-set of the infrastructure dedicated to the targeted domain,
- by exploiting the component paradigm not only at applicative level but also at infrastructure level. This allows us to capture the invariants of the infrastructure that could be reused independently of the targeted domain.

Hulotte is a generic framework focused on these two aspects. The originality of Hulotte is to be based on a flattened and unified representation of the whole system—*i.e.*, it allows reasoning on the applicative level as well as the infrastructure level. It is a mandatory requirement in order to ease the generation of domain-dependent components based on global properties of the system. The experiment described in [41] has been conducted using Hulotte. Hulotte is also based on several software developed within the ADAM team: Juliac, Koch, Spoon.

5.7. Soleil

Keywords: *Real-Time Java*.

Participants: Frédéric Loiret, Michal Malohlava, Philippe Merle, Aleš Plšek [correspondant], Lionel Seinturier.

The work on Soleil is the subject of Aleš Plšek's Ph.D. thesis and is a component-based software framework to support development of *Real-Time Java* (RTJ) based applications in the FRACTAL component model. The goal of the framework is to automatically generate an execution infrastructure of a software system that will be executed under real-time conditions as it is described in [29], [30]. The Soleil framework is able to substantially alleviate the whole process of developing RTSJ-based applications. The key motivation for the project is to shield developers from the complexities of RTSJ related code and allow them to focus only on development of the application logic. The Soleil framework is divided into several parts. First, we introduce a component model which is specifically designed for RTSJ and allows designers to manipulate with RTSJ concerns at the architectural level. Second, Soleil proposes a methodology of designing architectures of real-time Java applications. Finally, when the application is designed, the framework is used to automatically generate an infrastructure that corresponds to this real-time architecture.

5.8. Spoon

Keywords: *Attribute-Oriented Programming*, *Java annotations*, *Program transformation*.

Participants: Carlos Francisco Noguera Garcia [correspondant], Lionel Seinturier.

Spoon [67] is a project that started in March 2005 and has been officially hosted by INRIA Gforge since September 2005. The goal of Spoon is to provide a core API and associated tools for static analysis and generative programming within the Java 5+ environment. Spoon must be seen as a basis to ensure Software Quality through code validation and generation. It can be used in the software development process during the validation phase, as well as for engineering or re-engineering software.

The first key point of Spoon is to provide a well-typed and comprehensive AST API which is designed to facilitate analysis and transformation work for programmers. Scanners and processors allow the programmer to implement various program traversal strategies on the Java program. Also, the program representation is built with a well-known and well-tested open source Java compiler: the Eclipse JDT compiler, which ensures the support of the latest Java features.

The second key point of Spoon is to provide a pure Java API to specify program transformations using a well-typed generative programming technique (called Spoon Templates). By using well-typed templates, Spoon makes programming of transformations easier and safer for the end-user programmers.

Finally, the third key point of Spoon is that it provides an Eclipse plugin (SpoonJDT) that allows the programmers to package validations and transformations into compilation components called Spoonlets. These components can be deployed in the Eclipse plugin to enhance the Java compiler in a seamless and well-integrated way. For example, thanks to Eclipse's incremental compilation, errors and warnings coming from Spoonlets are reported as regular compilation mistakes, along with the typing of the program. The fact that Spoonlet-defined errors can be reported as the programmer types in the code (exactly like spelling or grammar mistakes are reported real-time by modern text editors) is of primary importance to produce high-quality code. Indeed, it is a recognized fact that the exact moment a programmer introduces a defect in the program is also the best time to fix it - because it is the moment when the programmer has the best understanding of what has just been written.

Many projects and experiments have been conducted around Spoon in the ADAM project-team, but also in other INRIA projects and outside of INRIA, by independent developers. A non-exhaustive list follows:

- **Spoon-AOP**: a work on implementing AOP with Spoon in the context of middleware [67].
- **AVal**: a work done within the Ph.D. of Carlos Francisco Noguera Garcia and that consists of a framework for validating annotation sets forming DSLs. This work was published in [64], [65].
- **Fraclet**: a work done in the context of Nicolas Pessemier's Ph.D., in collaboration with Romain Rouvoy, that defines and implements a Java annotation-based DSL for FRACTAL, the OW2 component model. This work was published in [19].
- **AOKell**: an alternative implementation of the OW2 FRACTAL container using AOP and Spoon [70].
- **Spoon-EMF**: a work done by Olivier Barais (TRISKELL INRIA project) to provide an EMF compliant implementation of the Spoon API.
- **Spoon-JMX**: a work done by Didier Donsez (Grenoble University) for automatically transforming Java programs to support JMX.
- **JUnit Suite Maker**: a project to control your JUnit tests with Spoon and started by an independent developer. Read more details at http://perso.orange.fr/peupeu/sw/java/junitsuitemaker/junitsuitemaker.html.
- **VSuite**: a validation suite for Java started by Renaud Pawlak and Nicolas Petitprez.
- **Spoon-Grafitti** : a transformation suite for Java applications to support ambient environment.
- **ModelAn**: a work done within the Ph.D. of Carlos Francisco Noguera Garcia and that consists of an annotation framework for describing a model in accordance with the annotations defined in the framework, and to express the constraints in terms of this model [25].
- Other work using Spoon are a migration compiler based on annotations, Heinz-Nixdorf-Institute in Paderborn, Germany and, finally, a loader for Spoon, University of Illinois at Urbana-Champaign.

Spoon is an LGPL open source software available at http://spoon.gforge.inria.fr, which is being developed since 2005. Spoon is written in Java and is composed of 36,000 lines of code.

6. New Results

6.1. Components for Embedded and Real-Time Systems

Participants: Frédéric Loiret, Michal Malohlava, Philippe Merle, Aleš Plšek, Lionel Seinturier [correspondant].

The *Real-Time Specification for Java* (RTSJ) is becoming a popular choice in the world of real-time and embedded programming. However, RTSJ introduces many non-intuitive rules and restrictions which prevent its wide adoption. Moreover, current state-of-the-art frameworks usually fail to alleviate the development process into higher layers of the software development life-cycle. Our result consists in considering RTSJ concepts at early stages of software development in a framework that provides continuum between the design and implementation process [30], [29]. SOLEIL (cf. section 5.7) is a component model designed specially for RTSJ and serves here as a cornerstone. As the first contribution of this work, we propose a development process where RTSJ concepts are manipulated independently on functional aspects. Second, we mitigate complexities of RTSJ-development by automatically generating execution infrastructure where real-time concerns are transparently managed. We thus allow developers to create systems for variously constrained real-time and embedded environments. Performed benchmarks show that the overhead of the framework is minimal in comparison to manually written object-oriented approach, while providing more extensive functionality. Finally, the framework is designed with the stress on dynamic adaptability of target systems, a property we envisage as a fundamental in an upcoming era of massively developed real-time systems.

6.2. Deployment of distributed and heterogeneous software systems

Participants: Nicolas Dolet, Jérémy Dubus, Areski Flissi, Damien Fournier, Philippe Merle.

Deployment is one of the most difficult software lifecycle step, and the less standardized. First, in the Jérémy Dubus Ph.D. thesis [12], we identify four challenges to solve to handle software systems deployment. The first challenge is about to initiate consensus for standard generic software deployment language. The second challenge consists in static verification of software deployment processes described using this language. These verifications are supposed to ensure the coherency of the described deployment process. Third challenge is about implementing a middleware platform able to interpret this language and perform deployment of any software system. Finally, fourth challenge is to transpose these deployment processes into open distributed environments which are fluctuating, such as ubiquitous and grid environments. Our contribution then consists to define a distributed systems deployment process divided in four roles to handle these challenges: the network expert, the software expert, system administrator and business architect. On the one hand, the DeployWare approach is defined by a multi-roles metamodel to describe deployment of the middleware layer of a system, and by the virtual machine able to automatically execute the described deployment, in conformance with the model driven engineering. Using a metamodeling language allows to implement static verification programs of the deployment models. On the other side, the DACAR approach proposes a generic architecture model to express and execute the deployment of a component-based application. The DeployWare and DACAR approaches allows to take into account during the deployment description the open distributed environments properties, in conformance with the autonomic computing approach. Our contribution is validated through many experiences in ubiquitous environments and in enterprise services world.

Deployment of distributed systems involves many heterogeneous technologies. The system administrator has to 1) master the deployment of each technology, 2) adapt it to machine properties and 3) execute it in respect with its order dependencies. These tasks are strongly prone to errors. In the article [21], we present DeployWare, a model-based approach for complex distributed systems deployment. This approach relies on a metamodel split in two parts. The first allows us to describe properties, dependencies, and actions to perform to deploy software. The second allows us to compose many software instances. This metamodel allows some behavioural deployment verifications to be performed. DeployWare models can be projected onto a component-based execution platform which manages automatically machines heterogeneity and orchestration of dependencies.

In the paper [23], we present DeployWare to address the deployment of distributed and heterogeneous software systems on large scale infrastructures, such as grids. Deployment of software systems on grids raises many challenges like 1) the complexity to take into account orchestration of all the deployment tasks and management of software dependencies, 2) the heterogeneity of both physical infrastructures and software composing the system to deploy, 3) the validation to early detect errors before concrete deployments and 4) scalability to tackle thousands of nodes. To address these challenges, DeployWare provides a metamodel that abstracts concepts of the deployment, a virtual machine that executes deployment processes on grids from DeployWare descriptions, and a graphical console that allows to manage deployed systems, at runtime. To validate our approach, we have experimented DeployWare with a lot of software technologies, such as CORBA and SOA-based systems, on one thousand of nodes of Grid 5000, the french experimental grid infrastructure dedicated to computer science research.

The DeployWare virtual machine is implemented by the open source FDF software (cf. section 5.3). At the Super Computing conference 2008, we show a demonstration of DeployWare/FDF for large-scale deployment on Grid'5000.

6.3. Software Engineering for Ambient Intelligence

6.3.1. COSMOS

Participants: Denis Conan [Telecom SudParis], Romain Rouvoy [correspondant], Lionel Seinturier.

COSMOS (*COntext entitieS coMpositiOn and Sharing*) [20], [18] is a component-based framework for context information management. This framework targets ubiquitous, context-aware applications. It deals with the gathering, the processing and the interpretation of context data. This is a middle layer between the infrastructure and the applications. The idea is that context-aware applications have to deal with numerous information coming from the hardware (CPU, memory, network, etc.) and the service resources available in their environment. COSMOS enables to structure this information and to represent adaptation policies as graphs of components. Each node in the graph represents a basic action of collecting or processing of a context information. We have defined a library of components for representing such nodes and a library of architectural patterns for connecting these nodes. By this way, we are able to more easily design and reuse adaptation policies for context-aware applications. COSMOS can be downloaded from http://picoforge.int-evry.fr/projects/svn/cosmos.

6.3.2. Annotation-Oriented Programming

Participants: Laurence Duchien, Carlos Francisco Noguera Garcia [correspondant].

Annotations, in the Java programming language, are a way to embed meta-data into the source-code of a program. Annotations can be used to extend the Java language with concepts specific to a domain. When used in this manner, annotations serve as means to reduce the semantic gap between concepts in the problem domain and the concepts provided by the programming language. For the annotation framework user—*i.e.*, the application programmer—it is important to understand how different annotations relate to each other in order to correctly use them and to get errors as soon as possible when these annotations are not correctly used.

Annotation frameworks have already been adopted by industry; however, their development remains complex, and it is done largely in an ad-hoc manner. When developing the annotation framework, the programmer must make sure that the program that uses the annotation complies with the constraints defined for it. Such constraints are often relegated to the documentation of the framework, since current annotation processors do not provide a way to specify and check them. In addition to this, current annotation processors just offer to the framework programmer the AST of the program as manipulation entity. This forces the programmer to reify the annotations by himself if he wants higher abstraction elements.

To help the annotation framework developer, we have proposed two annotation frameworks [15]. The first one, called AVal, provides a number of reusable, declarative and extensible constraints that can be used to specify the annotation framework, and can be interpreted in order to validate an annotated program. The second one, called ModelAn, allows the annotation framework developer to describe a model that corresponds to the annotations in the framework, and to express the constraints in terms of this model [25]. Annotated programs are then represented as instances of the (annotation) model, and the constraints checked on it. From this model, Java classes that reify the annotations are generated. The reified annotations can serve as starting point for the interpretation of the annotated program.

To validate the approach, we construct the annotation model, and describe the constraints as AVal and OCL expressions of three industrial annotation frameworks: Fraclet, an annotation framework for the FRACTAL component model, JWS for the development of Web Services in Java, and the Java Persistence API, part of the EJB3 specification. This work is the result of Carlos Francisco Noguera Garcia's PhD thesis.

6.3.3. Context-Aware Modelling

Participants: Laurence Duchien, Carlos Francisco Noguera Garcia, Rafael Laeño, Carlos Andres Parra Acevedo [correspondant].

Modelling has always been an essential part of software development. Models represent the domain of the software or make abstractions from the implementation. A recent trend in software development is Model-Driven Engineering, where models play a pivotal role in that programs or other models are generated from them. As such, it seems that concepts, behaviour and rules of Ambient Computing, and Context-Aware Computing in particular, have to be expressed at the modelling level. Evidently, the models have to be active and intimately connected to the actual implementation at the computational level, rather than serving passive documentation purposes.

In an ubiquitous computing setting it is however more natural to assume that each device in the environment manages its own part of the context. Distributing such context implies that support is needed for connection volatility. Using Tagged Futures and Spoon Graffiti we have implemented such support, allowing context to be distributed without adding a significant overhead to development. Preliminary results can be found in [26]. We have also defined the challenges of building context aware applications [27], and a possible approach has been described. Several prototypes have also been built in order to validate the ideas presented in these research proposal papers. These prototypes are intended to run in mobile devices and initially use web services to interact with the environment. Two different platforms are used to evaluate the common and particular features of mobile applications. This experimentation simplifies the design of models and the setup of a Software Product Line that automatically generates parts of these applications. The perspectives of this work involve the combination of SCA and MDE, to achieve static and dynamic adaptation.

6.3.4. Adaptive Middleware for Context-Awareness

Participants: Rubby Casallas, Laurence Duchien, Rafael Laeño, Carlos Andres Parra Acevedo, Daniel Romero [correspondant], Lionel Seinturier.

In pervasive environments, several entities exist that provide information and services that can be useful for the users [17]. However, the providers of these information and services are heterogeneous considering aspects, such as communication and discovery. Hence, in order to deal with this heterogeneity we want to build an adaptable middleware platform that will be deployed on mobile devices. This middleware platform will enable those devices to access the different resources present in the environments. In order to do that, we started exploring the existent context-aware middleware. This allowed the identification of challenges that are necessary to tackle in ubiquitous computation, such as context retrieval, dynamic and static adaptation, mobility and service discovery. We also identified FRACTAL [52], *Service Component Architecure* (SCA) [66], and COSMOS [20], [18] (a component-based framework for managing context in pervasive environments) as tools that can help in this work. Furthermore, we have defined a first architecture of the middleware platform [31]. In the next year, we expect to work in the detail specification and implementation of this middleware platform in the context of the CAPPUCINO project (cf. section 7.1).

6.4. Software Evolution

6.4.1. Static and Dynamic Checking of Component Interactions

Participants: Laurence Duchien, Anne-Françoise Le Meur [correspondant], Prawee Sriplakich, Guillaume Waignier.

In the context of Guillaume Waignier's PhD thesis we are working on an approach that enables architects to specify component-based architectures and their associated properties, and check the consistency of the component assembly specifications. This work has led to the development of CALICO, a *Component AssembLy Interaction Control FramewOrk* (cf. sections 5.2). CALICO is a model-based framework that enables architects to both specify component contractual application properties and perform validation at execution time when static verification is not sufficient.

More concretely, CALICO analyses architecture models and creates contracts by composing contractual application properties (*e.g.*, behavioral, dataflow, QoS properties). This composition allows compatible, incompatible interaction to be identified, as well as partially compatible interactions, which require runtime checking [35].

When runtime checking is required, CALICO automatically instruments the application to reify runtime information to complete the resolution of the partially compatible interaction contract and thus detects if the given interaction may lead to an error. By using this framework in iterative software design processes, architects get design feedback—*i.e.,* information on identified interaction errors—and can then modify the models accordingly. Each modification performed on the model is propagated to the running system since CALICO ensures the synchronization between the model and the runtime system [33]. This feature of CALICO has been developed in the context of the postdoctoral internship of Prawee Sriplakich.

Furthermore, the solution offered by CALICO is generic regarding underlying platforms. This genericity is achieved through mappings from the generic models to platform-specific API calls for constructing the running system [34], [42]. CALICO is available on http://calico.gforge.inria.fr.

6.4.2. Detection of Design Defects

Participants: Laurence Duchien [correspondant], Anne-Françoise Le Meur, Naouel Moha.

The Ph.D thesis of Naouel Moha [14] is about the detection and correction of design defects in object-oriented architectures. Design defects are bad solutions to recurring design problems in object-oriented systems, in opposition to design patterns, which are good solutions. In this context, we have performed an analysis of the domain of design defects and proposed a language to specify detection rules. From defect specifications, our approach enables the automatic generation of design defect detection algorithms. Generated algorithms have been validated in terms of precision and recall on XERCES v2.7.0, an open-source object-oriented system [24], [38].

6.5. INRETS-LEOST

Participants: Aurélien Bocquet, Christophe Gransart [correspondant].

Since several years, we collaborate with the *Laboratoire Electronique, Ondes et Signaux pour les Transports* (LEOST laboratory) of the french *Institut National de REcherche sur les Transports et leur Sécurité* (INRETS institute), and especially with Christophe Gransart in the context of the french ANR ARA REVE funded project [45]. This collaboration focuses on the investigation of middleware solutions enabling adaptive mobility. In particular, in the domain of railway communicative applications, Aurélien Bocquet works on an approach composed by a generic programming model, which proposes different kinds of synchronous, asynchronous, and design-pattern-based communications [39], [40], [22]. It is also composed by a combination of communication models, ensuring interoperability with standard middlewares, and offering capabilities of enhanced communications, able to adapt to changes of context. Adaptation policies define the rules of combination of models, regarding context observations, in order to behave the most efficient way, considering its evolution. Dynamic adaptation mechanisms allow the approach to propose to handle context changes in real-time, and allow the reconfiguration of the running system in order to answer to deployment needs. This approach has been validated on problems caused by the use of an embedded Internet proxy in trains. Furthermore, we have implemented a multi-model software infrastructure, offering all the concepts of the approach. This work has been done in the context of Aurelien's PhD Thesis [11].

7. Other Grants and Activities

7.1. National Initiatives

7.1.1. ANR ARA REVE

Participants: Laurence Duchien, Anne-Françoise Le Meur, Frédéric Loiret, Aleš Plšek, Lionel Seinturier.

REVE (*safe Reuse of Embedded components in heterogeneous enVironmEnts*) is a 3-years project funded by the ANR ARA SI (Sécurité, Systèmes embarqués et Intelligence Ambiante) program of the ANR. Five partners are involved: CEA, CNAM, INRETS, INRIA, and INSA. The objective of the project is to define a component model, a type system, and an execution platform for context-aware embedded applications [60]. Lionel Seinturier is the scientific leader of this project. The project has started in January 2006 and is scheduled for a 42-months period. Further information is available on the website of the project: http://www.ara-reve.org.

7.1.2. ANR TLog FAROS

Participants: Laurence Duchien, Guillaume Dufrêne, Anne-Françoise Le Meur, Lionel Seinturier, Guillaume Waignier.

FAROS (composition de contrats pour la Fiabilité d'ARchitectures Orientées Services) is an ANR TLog project involving EDF R&D, France Telecom, Alicante, IRISA, I3S and LIFL. This project addresses issues related to the safe integration of services in service-oriented architectures. The overall goal of this project is to provide a methodology to guide and automate the different tasks involved in the integration process, thus enabling the integration process to become reproducible. Our approach will be based on the definition of contracts, which will allow static and dynamic verifications to be performed. The feasibility of our approach will be demonstrated through the development of three dynamic and constrained applications (Health, Education, Electricity).

The project has started in January 2006 and is scheduled for a 36-months period. Several deliverables have been produced this year and the project has been presented at the Research Projects Symposium of the MODELS conference. Further information is available on the website of the project: http://www.lifl.fr/faros.

7.1.3. ANR TLog Flex-eWare

Participants: Frédéric Loiret, Philippe Merle, Aleš Plšek, Lionel Seinturier.

Flex-eWare is a 3-years ANR TLog project (of type "*plate-forme exploratoire*"), which has started on January 1st 2007. The project aims at defining an open and adaptable middleware platform for component-based applications in the domain of embedded systems. This project addresses issues that are related to components, software architectures, and adaptable middleware. One of the research challenges of this project is to be able to define software components, which are context-aware and can take into account the specificities in terms of resource management of the targeted embedded systems. Besides, Flex-eWare aims at federating and unifying the component approaches (Lw-CCM and FRACTAL) of the industrial and academic partners of the project, consolidating these technologies for the domain of embedded systems, and opening and fostering the use of these technologies by contributing some reference open source implementation. The partners of this project are: Thales (leader), France Telecom, Schneider, STMicroelectronics, Teamlog, Trialog, CEA, ENST, INRIA, Univ. Paris 6. Further information is available on the website of the project: http://www.flex-eware.org.

7.1.4. ANR TLog SCOrWare

Participants: Nicolas Dolet, Damien Fournier, Philippe Merle, Lionel Seinturier.

SCOrWare is a 24-months ANR TLog project started in January 2007 and involving Artenum, EBM WebSourcing, Edifixio, INRIA (ADAM, OW, and Sardes project-teams), INT, IRIT, Obeo, Open Wide, and Thales Communications. Philippe Merle is the scientific leader of the project.

The SCOrWare project has tree main goals: 1) Studying links between *Service-Oriented Architectures* (SOA) and architectural component-based approaches, and especially similarities/differences between *Service Component Architecture* (SCA), *Java Business Integration* (JBI), and FRACTAL component models, 2) developing a component-based implementation of the SCA specification, and 3) developing high-level MDE tools that will help end-users to adopt this new technology. For more details, see at http://www.scorware.org.

The main ADAM's contributions are: FraSCAti (see section 5.5) to support at runtime the SCA component model, and FDF (see section 5.3) to deploy distributed SCOrWare infrastructures and applications.

7.1.5. INRIA ADT galaxy

Participants: Christophe Demarey, Philippe Merle, Lionel Seinturier, Alban Tiberghien.

The galaxy ADT (*Technology Development Action*) contributes to make INRIA a value-added player in the SOA arena, by designing and developing an Open Framework for Agile and Dynamic Software Architecture. This ADT will work for INRIA and INRIA's research project-teams direct benefit, and aims at pre-assembling technological bricks from various teams, projects and preparing them to be transferred through the open source software channel.

The goal that the galaxy ADT aims at achieving is to provide an IT agile platform, built on dynamic software architecture principles, and fitting for flexibility, dynamical reconfiguration, adaptability, continuity and autonomic computing. FRACTAL, SCA-FraSCAti and GCM-ProActive are the major technologies which will be the technological drivers of this ADT. The different usage scenarios as well as the different tools which will be developed at infrastructure, application and business levels will demonstrate that this platform is able to support the design, modelling, deployment and execution of business processes. At the same time, the ADT will target the definition of a new common language to manipulate dynamically adaptive distributed SOA-based systems, encompassing application and middleware layers. This common language will take different forms, inherited from works done by several project-teams with their distinct skills, and illustrates a new kind of collaboration between teams, coupling research and development works.

Contributors to this ADT are mainly research project-teams, including ADAM, ECOO, OASIS, OBASCO, OW@INRIA, SARDES and TRISKELL, and the galaxy ADT is led and managed by the OW@INRIA team. For more details, see at http://galaxy.gforge.inria.fr.

7.1.6. INRIA ARC BROCCOLI

Participants: Philippe Merle, Romain Rouvoy, Lionel Seinturier, Alban Tiberghien.

The goal of the BROCCOLI ARC project is to design a platform for describing, deploying, executing, observing, administrating, and reconfiguring large-scale component-based software architectures, in particular for building discrete event simulation applications. Both the platform and the applications considered are based on software components and architecture description languages, and in particular on the FRACTAL component model. Therefore, the main issue addressed in this project is the scalability of such component-based architectures based on FRACTAL. Indeed, by allowing FRACTAL to scale-up to millions of components running on thousand of nodes, our aim is to establish the foundations of a massively distributed component-based simulator based on FRACTAL. However, the Broccoli road is paved of ambushes: existing ADLs are insufficient to describe very large architectures, the numerous data-flows and statistics generated by observations need smart context management policies, and distributed simulation is a long studied but still challenging problem, especially when combined with AOP techniques and FRACTAL special features such as shared components.

This project establishes a collaboration between the INRIA MASCOTTE and ADAM project-teams, and Télécom SudParis ACMES team. For more details, see at http://golgoth.inria.fr/wiki/Contrats/Broccoli.

7.1.7. Trade cluster CAPPUCINO

Participants: Pierre Carton, Jérémy Dubus, Laurence Duchien, Areski Flissi, Carlos Andres Parra Acevedo, Daniel Romero, Romain Rouvoy, Lionel Seinturier.

CAPPUCINO (Construction et Adaptation d'aPPlications Ubiquitaires et de Composants d'INtergiciels en environnement Ouvert pour l'industrie du commerce) is a 36-months project of the competitiveness cluster of trade industry of Nord/Pas-de-Calais PICOM (Pôle des Industries du COMmerce – http://www.picom.fr), which has started in September 2007. The Project CAPPUCINO aims to propose reliable solutions with the design, the deployment and execution problems of applications for ambient environments in the trade industry. This project addresses issues which are related to modeling the context-aware information for ambient environments and to take into account this information into runtime platforms. We propose to study the adaptation of the components - application components and execution platform supports - and their context evolution throughout the complete lifecycle of the application. The first ADAM's results are described in the section 6.3.

The partners of this project are NorSys (leader) which is a SME service company, TELCOM SUD Paris (School of Telecom industry), INRIA, AUCHAN and SI3SI (2 trade companies).

7.1.8. Trade cluster ICOM

Participants: Laurence Duchien, Julien Ellart, Gabriel Hermosillo, Lionel Seinturier.

This project is realized in the context of the competitiveness cluster of trade industry of Nord-Pas de Calais PICOM (Pôle des Industries du COMmerce – http://www.picom.fr) and it has started in March 2008 and is a 24-months project. Trade industry are being in constant evolution. The massive apparition of the Internet, the increasing exigence of quality of service, the ubiquitous and pervasive informatics shatter the traditional trade practices, their economical and organizational models.

ICOM (Infrastructure pour le COMmerce du futur) aims at helping enterprises to enable a fast and easy deployment of new applications using new technologies and infrastructures from ubiquitous informatics. It will provide a smart infrastructure that will hide the heterogeneity of identifiers (RFID, NFC, bar code) and manage data storage and request routing to provide scalability.

The partners of this project are Atos Origin, Auchan, Décathlon, La Poste, La Redoute, INRIA-ASAP, INRIA-ADAM, INRIA-POPS, GS1, ORANGE France.

7.1.9. Trade cluster MIND

Participants: Yann Davin, Frédéric Loiret, Philippe Merle, Lionel Seinturier.

MIND is a 2-years project funded by the Minalogic cluster on micro- and nano-technologies. The project started in 2008. It aims at consolidating the component-based technologies and the tools, which exist around the FRACTAL component model for building middleware and systems. The goal is to transfer these results into an industrial strength software tool suite in order to foster the adoption of the component-based technologies for designing and developing embedded applications and systems. The partners of this project include: STMicroelectronics (Leader), CEA, France Telecom R&D, Grenoble 1, INERIS, INRIA, IST, ISTIA, Itris Automation Square, LOGICA, Schneider Electric, Sogeti High Tech, VERIMAG.

7.2. European Initiatives

7.2.1. ERCIM Working Group Software Evolution

Participants: Stéphane Ducasse, Laurence Duchien.

The Working Group (WG) on Software Evolution is one of the many working groups supported by ERCIM. The main goal of the WG is to identify a set of formally-founded techniques and associated tools to support software developers with the common problems they encounter when evolving large and complex software systems. With this initiative, the WG plans to become a Virtual European Research and Training Centre on Software Evolution. We are active members of this WG. Read more at http://www.planet-evolution.org.

7.2.2. IAP MoVES

Participants: Laurence Duchien, Carlos Francisco Noguera Garcia, Carlos Parra, Aleš Plšek, Daniel Romero, Guillaume Waignier.

The Belgium IAP (Interuniversity Attraction Poles) MoVES (Fundamental Issues in Software Engineering: Modeling, Verification and Evolution of Software) is a project whose partners are the Belgium universities (VUB, KUL, UA, UCB, ULB, FUNDP, ULg, UMH) and three European institutes (INRIA, IC and TUD) respectively from France, Great Britain and Netherlands. This consortium combines the leading Belgian research teams and their neighbors in software engineering, with recognized scientific excellence in MDE, software evolution, formal modeling and verification, and AOSD. The long term objective of our network is to strengthen existing collaborations and forge new links between those teams, and to leverage and disseminate our research expertise in this domain at an European level. The project focusses on the development, integration and extension of state-of-the-art languages, formalisms and techniques for modeling and verifying dependable software systems and supporting the evolution of Software-intensive systems. The project has started in January 2007 and is scheduled for a 60-months period. Read more at http://moves.vub.ac.be

7.2.3. INRIA associate team CALA

Participants: Laurence Duchien, Anne-Françoise Le Meur, Carlos Francisco Noguera Garcia, Carlos Andres Parra Acevedo, Daniel Romero, Lionel Seinturier.

The ADAM project-team together with the Prog-Ssel team from Vrije Universiteit Brussel (VUB) focus on crosscutting concerns at the level of software architectures, component models and object-oriented programs. More specifically, we work together in CALA (Aspect-Oriented Software Development in Languages, Component Models and Architectures), an INRIA Associate Team project, (http://jacquard.lifl.fr/CALA) for elaborating and extending joint work we have already started in the field of AOSD on four particular topics: (1) logic pointcuts at the object-oriented programming level, (2) complex aspects, (3) the unification of aspects and components, and (4) pointcuts at the architectural level. The associate team has started in January 2006 and will ended at the end of 2008.

7.2.4. IST FP6 NoE AOSD-Europe

Participants: Laurence Duchien, Carlos Francisco Noguera Garcia, Lionel Seinturier.

AOSD-Europe is an ongoing proposal to set up a Network of Excellence (NoE) on AOSD within IST-FP6. The proposal brings together 11 research groups and among them members of the ADAM project-team and other members from OBASCO, Pop-Art and Triskell INRIA project-teams. The proposal is led by Lancaster University, Darmstadt University and University of Twente. The goal of the NoE is to harmonize, integrate and strengthen European research activities on all issues related to aspect orientation: analysis, design, development, formalization, applications, empirical studies. The project has started in January 2004 and is scheduled for a 48-months period. http://www.aosd-europe.net

7.2.5. IST FP7 SOA4All Integrated Project

Participants: Philippe Merle, Alban Tiberghien.

Service-Oriented Architectures for All (SOA4All) is a large-scale Integrating Project funded by the European Seventh Framework Programme, under the Service and Software Architectures, Infrastructures and Engineering research area. This is a 36-months project started in March 2008. Partners are: Atos Origin (Spain), British Telecommunications (UK), CEFRIEL (Italy), EBM WebSourcing (France), Hanival Internet Services GmbH (Austria), INRIA (France), Intelligent Software Components (Spain), Ontotext Lab (Bulgaria), Open University (UK), SAP AG (Germany), Seekda OG (Austria), TIE (Nederlands), The University of Manchester (UK), TXT e-Solutions Spa (Italy), Universitaet Karlsruhe (Germany), University Innsbruck (Austria).

SOA4All will help to realize a world where billions of parties are exposing and consuming services via advanced Web technology: the main objective of the project is to provide a comprehensive framework and infrastructure that integrates complementary and evolutionary technical advances—*i.e.*, SOA, context management, Web principles, Web 2.0 and Semantic Web—into a coherent and domain-independent service delivery platform. For more details, see at http://www.soa4all.eu.

7.3. International Initiatives

7.3.1. OW2

Participants: Yann Davin, Nicolas Dolet, Philippe Merle, Romain Rouvoy, Lionel Seinturier.

OW2, previously ObjectWeb, is an international consortium to promote high quality open source middleware (see at http://www.ow2.org). The vision of OW2 is that of a set of components which can be assembled to offer high-quality middleware.

We are members of this consortium since a long time ago. Philippe Merle is the leader of both FRACTAL and FraSCAti projects, which are hosted by this consortium.

7.3.2. University of Montreal

Participants: Laurence Duchien, Anne-Françoise Le Meur, Naouel Moha.

The Ph.D. Student Naouel Moha is co-supervized by Y.G. Gueheneuc, University of Montreal, and Laurence Duchien and Anne-Françoise Le Meur from University of Lille 1. The objective is to study the design defect and to correct them automatically. Design defects are poor design choices that degrade the quality of object-oriented designs and present opportunities for improvements. However, the detection and correction of design defects are difficult because of the lack of precise specifications and tools. Our goal is to provide a systematic method to specify design defects precisely and to generate detection and correction algorithms from their specifications. The detection algorithms are based not only on metrics but also on semantical and structural properties. The correction algorithms are based on refactorings and use our Spoon tool. These algorithms are applied and validated on open-source object-oriented programs to show that our method allows a systematic description, a precise detection, and a suitable correction of design defects. The student has been in the ADAM project-team during one year and the Ph.D. defense has taken place in August 2008 [14]. Several exchanges of one or two weeks for the advisors have allowed the coordination of the Ph.D. co-supervision.

8. Dissemination

8.1. Visiting researchers

- R. Casallas, University of Los Andes, Colombia, June 2008,
- A. Lozano, Open University, United Kingdom, since July 2008,
- M. Malohlava, Charles University, Prague, June–September 2008,
- R. Leaño, University of Los Andes, Bogota, September–December 2008.

8.2. Scientific Community Animation

8.2.1. Examination Committees

- **Laurence Duchien** was in the examination committee of the following Ph.D. thesis:
  - Meriem Belguedoum, February 2008, Telecom Bretagne (referee),
  - Frédéric Loiret, May 2008, University of Lille 1 (co-advisor),
  - Cristina Marin, May 2008, University of Grenoble 1 (referee),
  - Nicolas Denos, June 2008, University of Montpellier (referee),
  - François Merkerke, July 2008, Telecom Bretagne (referee),
  - Naouel Moha, August 2008, University of Lille 1 and University of Montreal (co-advisor),
  - Chantal Kaboré, September 2008, Telecom Bretagne (chair),
  - Gauthier Loyauté, September 2008, University of Marne La Vallée (referee),

  - Sebastian Pavel, October 2008, University of Nantes (referee),
  - Emmanuel Filiot, October 2008, University of Lille 1 (chair),
  - Carlos Francisco Noguera Garcia, November 2008, University of Lille 1 (advisor).
- **Jean-Marc Geib** was in the examination committee of the following Ph.D. thesis:
  - Jérémy Dubus, October 2008, University of Lille 1 (co-advisor).
  - Aurélien Bocquet, December 2008, University of Lille 1 (co-advisor).
- **Christophe Gransart** was in the examination committee of the following Ph.D. thesis:
  - Aurélien Bocquet, December 2008, University of Lille 1 (co-advisor).
- **Philippe Merle** was in the examination committee of the following Ph.D. thesis:
  - Hinde Bouziane, February 2008, University of Rennes (referee),
  - Jérémy Dubus, October 2008, University of Lille 1 (co-advisor).
- **Lionel Seinturier** was in the examination committee of the following HDR thesis:
  - Stéphane Frénot, December 2008, INSA Lyon (member),
  - Samia Bouzefrane, December 2008, University of Lille 1 (member).
- **Lionel Seinturier** was in the examination committee of the following Ph.D. thesis:
  - Frédéric Loiret, May 2008, University of Lille 1 (co-advisor),
  - Fabien Baligand, June 2008, Ecole des Mines de Paris (referee),
  - Nagapraveen Jayaprakash, June 2008, Institut National Polytechnique Grenoble (referee),
  - Franck Chauvel, September 2008, University of Bretagne Sud (referee),
  - Khaled Barbaria, September 2008, Telecom Paris (referee),
  - Jakub Kornas, October 2008, University of Grenoble 1 (referee),
  - Béchir Zalila, November 2008, Telecom Paris (referee),
  - André Bottaro, December 2008, University of Grenoble 1 (referee),
  - Antonio Cansado, December 2008, University of Nice (member).

8.2.2. Journals, Conferences, Workshop

- **Laurence Duchien** has been
  - member of the following committees:
    - Program committee of the 8th Annual International Conference on New Technologies of Distributed Systems (NOTERE'08), http://liris.cnrs.fr/notere08,
    - Program committee of the 6th International Conference on Software Engineering Research, Management and Applications (SERA'08), http://dsrg.mff.cuni.cz/sera,
    - Program committee of the 14th International Symposium on Object-oriented Languages and Models (LMO'08), http://lmo08.iro.umontreal.ca,
    - Program committee of the 4th French Workshop on Model-Driven Engineering (IDM'08), http://planet-mde.org/idm08,
    - Program committee of the 6th French Conference on Operating Systems (CFSE'08), http://www-src.lip6.fr/homepages/Pierre.Sens/cfse6,
    - Program committee of the 4th International Conference on e-Technologies (MCETECH'08), http://www.mcetech.org,

    * Program committee of the 2nd European Conference on Software Architecture (ECSA'08), http://www.ecsa2008.cs.ucy.ac.cy,
    * Program committee of the 9th Mexican International Conference on Computer Science (ENC'08), http://enc.smcc.org.mx,
    * Program committee of the special issue on Architectures Logicielles of journal RSTI-L'Objet, Hermès, March 2008,
    * Editorial board of the International Journal Software Architecture, since 2007.
  – reviewer for the following journals:
    * The Computer Journal,
    * International Journal Software Architecture,
    * Journal Computing and Informatics.

- **Anne-Françoise Le Meur** has been member of the following committees:
  – Organizing committee and program committee of the 3rd AOSD Workshop on Domain Specific Aspect Languages (DSAL'08), http://dsal.dcc.uchile.cl/2008,
  – Program committee of the 7th International Conference on Generative Programming and Component Engineering (GPCE'08),
  – Program committee of the 18th International Workshop on Partial Evaluation and Program Manipulation (PEPM'08), http://www.program-transformation.org/PEPM08.

- **Philippe Merle** has been member of the following committees:
  – Editorial board of the RSTI-L'Objet journal, Hermès.
  – Program committee of the 6th Conférence Française sur les Systèmes d'Exploitation (CFSE'2008), http://gridgroup.tic.hefr.ch/renpar,
  – Program committee of the 7th Middleware Workshop on Adaptive and Reflective Middleware (ARM'08), http://www.comp.lancs.ac.uk/computing/arm2008.

- **Lionel Seinturier** has been a member of the following committees:
  – Editorial board of the TSI (Hermes) journal,
  – Program Chair of the International Workshop on Advanced Information Systems for Enterprises (IWAISE'08), http://www.lifl.fr/iwaise08
  – Tutorial Chair of the 7th International Conference on Aspect-Oriented Software Development (AOSD'08), http://aosd.net/2008,
  – Program committee of the 13th International Conference on Reliable Software Technologies (Ada-Europe'08), http://www.math.unipd.it/ae2008,
  – Program committee of the 1st International DisCoTec Workshop on Context-aware Adaptation Mechanisms for Pervasive and Ubiquitous Services (CAMPUS'08), http://discotec08.ifi.uio.no/Campus08,
  – Program committee of the 2nd French Conference on Architectures Logicielles (CAL'08), http://cal08.iro.umontreal.ca.

- **Romain Rouvoy** has been a member of the following committees:
  – Organizing committee and Program chair of the 1st International DisCoTec Workshop on Context-aware Adaptation Mechanisms for Pervasive and Ubiquitous Services (CAMPUS'08), http://discotec08.ifi.uio.no/Campus08,

8.3. Scientific and Administrative Responsibilities

Team members have several scientific and administrative responsibilities in the university, the INRIA institute and at the national level:

- **Laurence Duchien** is member of the ERCIM Software Evolution Group since 2006. She is also Co-chair (with Jean-Louis Giavitto) of the Languages and Verification group of the GDR CNRS GPL (Génie de la Programmation et du Logiciel – http://gdr-gpl.imag.fr). In 2008, she has been chair of the Postdoc and PhD CORDIS Committee for the CR INRIA-Lille-Nord Europe. She has also been member of the scientific committee in the ANR ARPEGE Program and she has been member of the AERES evaluation committee for the Computer Science Laboratory LIASD, Paris 8;
- **Anne-Françoise Le Meur** is member of CSE 27nd section of University of Lille 1. She is elected member of the board of the LIFL laboratory. She has served as a jury during the recruitment of INRIA R&D engineers in june-july 2008 at INRIA Lille – Nord Europe;
- **Philippe Merle** is vice-president of *Technological Development Council* (CDT) of the INRIA Lille Nord Europe research center, and member of the *Incitative Action Working Group* (GTAI) of the *Scientific and Technological Orientation Council* (COST) of INRIA; He is member of the steering committee of the "Grilles, Système et Parallélisme" (GSP) working group of the CNRS ARS GdR.
- **Lionel Seinturier** is elected member of the Comité de Centre INRIA Lille Nord Europe and member of Comité Hygiène et Sécurité INRIA Lille Nord Europe. He is in charge of supervising the financial activities of the LIFL laboratory. The LIFL direction board is composed of three persons: S. Tison (director), P. Boulet (vice-director) and L. Seinturier (in charge of financial activities). Lionel Seinturier is member of the CSE 27 at the University of Nanterre and of CSE 27 at CNAM Paris. Lionel Seinturier is member of the board of the IEEA faculty (UFR) at the University of Lille 1 and member of the board of the LIFL laboratory. In May 2008, he has served as member of the Commission d'avancement of the IEEA faculty (UFR) at the University of Lille.

8.4. Teaching

Permanent members teach the following courses:

- **Laurence Duchien** teaches a course on Architecture Description Languages in Master Research Sciences et Technologies Mention Informatique at USTL, UFR IEEA;
- **Anne-Françoise Le Meur** teaches distributed application design (Master 1), databases and the Internet (Master 2 pro), systems programming (Licence 3) and a course on Domain-Specific Languages (DSL) and on "Being a researcher" in Master 2 Research at USTL, UFR IEEA;
- **Philippe Merle** teaches a course on Middleware and Design Patterns in Master Research Sciences et Technologies Mention Informatique at USTL, UFR IEEA;
- **Lionel Seinturier** heads the E-Service specialty of the Master of Computer Science at the University of Lille 1 (USTL). He teaches several courses on middleware, component-based software engineering, aspect-oriented programming, and object-oriented design in Master of Computer Science at the University of Lille 1.

8.5. Miscellaneous

- The team has organized its internal seminar, Montreuil-sur-Mer, France, September 10–12th, 2008.
- The team has organized a CALA seminar, Lille, INRIA, June 2008.
- The team has organized a research seminar on adaptive middleware, Lille, March 13th 2008 (http://www.fil.univ-lille1.fr/ seinturi/doct/AdaptLille2008.html).

9. Bibliography

Major publications by the team in recent years

[1] O. BARAIS, J. LAWALL, A.-F. LE MEUR, L. DUCHIEN. *Software Architecture Evolution*, in "Software Evolution", Springer-Verlag, February 2008, p. 233–262.

[2] C. DEMAREY, G. HARBONNIER, R. ROUVOY, P. MERLE. *Benchmarking the Round-Trip Latency of Various Java-Based Middleware Platforms*, in "Studia Informatica Universalis Regular Issue", ISBN: 2-912590-31-0, vol. 4, n^{o} 1, May 2005, p. 7–24.

[3] J.-M. GEIB, P. MERLE. *CORBA : des concepts à la pratique*, in "Techniques de l'Ingénieur. Informatique", vol. HB2, n^{o} H2758, February 2000, p. 1–31.

[4] R. PAWLAK, L. DUCHIEN, L. SEINTURIER. *CompAr: Ensuring Safe Around Advice Composition*, in "Proceedings of the 7th IFIP International Conference on Formal Methods for Open Object-based Distributed Systems (FMOODS 2005), Athens, Greece", June 2005, p. 163-178.

[5] R. PAWLAK, L. SEINTURIER, L. DUCHIEN, G. FLORIN. *JAC: A Flexible Solution for Aspect-Oriented Programming in Java*, in "REFLECTION '01: Proceedings of the Third International Conference on Metalevel Architectures and Separation of Crosscutting Concerns, London, UK", Springer-Verlag, 2001, p. 1–24.

[6] N. PESSEMIER, L. SEINTURIER, T. COUPAYE, L. DUCHIEN. *A Component-Based and Aspect-Oriented Model for Software Evolution*, in "International Journal of Computer Applications in Technology (IJCAT)", vol. 31, n^{o} 1-2, 2008, p. 94-105, http://hal.inria.fr/inria-00269895.

[7] A. PLSEK, F. LOIRET, P. MERLE, L. SEINTURIER. *A Component Framework for Java-based Real-time Embedded Systems*, in "Proceedings of ACM/IFIP/USENIX 9th International Middleware Conference, Leuven, Belgium", V. ISSARNY, R. SCHANTZ (editors), Lecture Notes in Computer Science, Rank (CORE) : A, Springer-Verlag, December 2008, http://hal.inria.fr/inria-00301410.

[8] R. ROUVOY, P. MERLE. *Abstraction of Transaction Demarcation in Component-Oriented Platforms*, in "Proceedings of 4th ACM/IFIP/USENIX International Middleware Conference (Middleware 2003), Rio de Janeiro, Brasil", Lecture Notes in Computer Science (LNCS), vol. 2672, Springer-Verlag, June 2003, p. 305–323.

[9] R. ROUVOY, P. SERRANO-ALVARADO, P. MERLE. *A Component-based Approach to Compose Transaction Standards*, in "Proceedings of the 5th International Symposium on Software Composition (SC'06), Vienna, Austria", Lecture Notes in Computer Science, vol. 4089, Springer-Verlag, March 2006, p. 114–130.

[10] G. WAIGNIER, P. SRIPLAKICH, A.-F. LE MEUR, L. DUCHIEN. *A Model-Based Framework for Statically and Dynamically Checking Component Interactions*, in "Proceedings of the ACM/IEEE 11th International Conference on Model Driven Engineering Languages and Systems (MODELS'08), Toulouse, France", Lecture Notes in Computer Science, Rank (CORE) : A, vol. 5301, Springer-Verlag, October 2008, p. 371-385, http://hal.inria.fr/inria-00311584.

Year Publications

Doctoral Dissertations and Habilitation Theses

[11] A. BOCQUET. *Infrastructure logicielle multi-modèles pour l'accès à des services en mobilité*, Ph. D. Thesis, Université des Sciences et Technologies de Lille (USTL), Laboratoire d'Informatique Fondamentale de Lille (LIFL), December 2008.

[12] J. DUBUS. *Une démarche orienté modèle pour le déploiement de systèmes en environnements ouverts distribués*, Ph. D. Thesis, Université des Sciences et Technologies de Lille (USTL), Laboratoire d'Informatique Fondamentale de Lille (LIFL), Villeneuve d'Ascq, France, October 2008.

[13] F. LOIRET. *Tinap : Modèle et infrastructure d'exécution orienté composant pour applications multi-tâches à contraintes temps réel souples et embarqué*, Ph. D. Thesis, Université des Sciences et Technologies de Lille (USTL), Laboratoire d'Informatique Fondamentale de Lille (LIFL), Lille, France, June 2008, http://tel.archives-ouvertes.fr/tel-00321745.

[14] N. MOHA. *DECOR : Détection et correction des défauts dans les systèmes orientés objet*, Ph. D. Thesis, Université des Sciences et Technologies de Lille (USTL), Laboratoire d'Informatique Fondamentale de Lille (LIFL) et Université de Montréal, Lille, France, August 2008, http://tel.archives-ouvertes.fr/tel-00321081.

[15] C. F. NOGUERA GARCIA. *A Model-driven tool-set for Developing and Validating Annotation Frameworks*, Ph. D. Thesis, Université des Sciences et Technologies de Lille (USTL), Laboratoire d'Informatique Fondamentale de Lille (LIFL), Lille, France, November 2008.

Articles in International Peer-Reviewed Journal

[16] N. PESSEMIER, L. SEINTURIER, T. COUPAYE, L. DUCHIEN. *A Component-Based and Aspect-Oriented Model for Software Evolution*, in "International Journal of Computer Applications in Technology (IJCAT)", vol. 31, n° 1-2, 2008, p. 94-105, http://hal.inria.fr/inria-00269895.

[17] D. ROMERO. *Context-aware Middleware: An overview*, in "PARADIGMA", vol. 2, n° 3, 2008.

[18] R. ROUVOY, D. CONAN, L. SEINTURIER. *Software Architecture Patterns for a Context Processing Middleware Framework*, in "IEEE Distributed Systems Online (DSO)", vol. 9, n° 6, June 2008, 12, http://hal.inria.fr/inria-00286616.

[19] R. ROUVOY, P. MERLE. *Leveraging component-based software engineering with Fraclet*, in "Annals of Telecommunications, Special Issue on Software Components - The Fractal Initiative (Springer)", Accepted for publication, vol. 72, 2008, 25.

Articles in National Peer-Reviewed Journal

[20] D. CONAN, R. ROUVOY, L. SEINTURIER. *COSMOS : composition de noeuds de contexte*, in "Techniques et Sciences Informatiques", vol. 27, n° 9-10, 2008, p. 1189–1224, http://hal.inria.fr/inria-00330574.

[21] J. DUBUS, A. FLISSI, N. DOLET, P. MERLE. *Une démarche orientée modèle pour déployer des systèmes logiciels répartis*, in "Revues des Sciences et Technologies de l'Information - L'Objet", vol. 14, July 2008, p. 35–59.

International Peer-Reviewed Conference/Proceedings

[22] A. BOCQUET, C. GRANSART. *Benefits of Multi-model Architecture-based Development for Railway Applications*, in "Proceedings of the 8th World Congress on Railway Research (WCRR'08), Seoul, Korea", May 2008.

[23] A. FLISSI, J. DUBUS, N. DOLET, P. MERLE. *Deploying on the Grid with DeployWare*, in "Proceedings of the 8th International Symposium on Cluster Computing and the Grid (CCGRID'08), Lyon, France", Rank (CORE) : A, IEEE, May 2008, p. 177–184, http://hal.inria.fr/hal-00259836.

[24] N. MOHA, Y.-G. GUÉHÉNEUC, A.-F. LE MEUR, L. DUCHIEN. *A Domain Analysis to Specify Design Defects and Generate Detection Algorithms*, in "Proceedings of the 11th International Conference on Fundamental Approaches to Software Engineering (FASE)", J. FIADEIRO, P. INVERARDI (editors), Acceptance rate : 26%, Rank (CORE) : B, Springer-Verlag, 2008.

[25] C. F. NOGUERA GARCIA, L. DUCHIEN. *Annotation Framework Validation using Domain Models*, in "Fourth European Conference on Model Driven Architecture Foundations and Applications (ECMDA-FA'08), Berlin, Germany", I. SCHIEFERDECKER, A. HARTMAN (editors), Lecture Notes on Computer Science (LNCS), Acceptance rate : 25% Rank (CORE) : B, n^{o} 5095, Springer-Verlag, June 2008, p. 48 - 62, http://hal.inria.fr/inria-00287856.

[26] C. F. NOGUERA GARCIA, J. FABRY, E. VAN PAESSCHEN, C. A. PARRA ACEVEDO. *Generation of Application Composition Support in Ubiquitous Computing Based on Context Models*, in "23rd Annual ACM Symposium on Applied Computing (SAC'08), Fortaleza, Ceará, Brazil", Rank (CORE) : B, Association for Computing Machinery ACM, March 2008, p. 1647-1648, http://hal.inria.fr/inria-00268299.

[27] C. A. PARRA ACEVEDO, L. DUCHIEN. *Model-Driven Adaptation of Ubiquitous Applications*, in "1st International Workshop on Context-aware Adaptation Mechanisms for Pervasive and Ubiquitous Services (CAMPUS 08), Oslo, Norway", June 2008, p. 97–102, http://hal.inria.fr/inria-00289953.

[28] A. PLSEK, J. ADÁMEK. *Carmen: Software Component Model Checker*, in "Proceedings of the 4th International Conference on the Quality of Software-Architectures (QoSA'08), Karlsruhe, Germany", S. BECKER, F. PLASIL, R. REUSSNER (editors), Lecture Notes in Computer Science, Rank (CORE) : A, vol. 5281, Springer-Verlag, October 2008, p. 71–85, http://hal.inria.fr/inria-00282373.

[29] A. PLSEK, F. LOIRET, P. MERLE, L. SEINTURIER. *A Component Framework for Java-based Real-time Embedded Systems*, in "Proceedings of ACM/IFIP/USENIX 9th International Middleware Conference, Leuven, Belgium", V. ISSARNY, R. SCHANTZ (editors), Lecture Notes in Computer Science, Rank (CORE) : A, Springer-Verlag, December 2008, http://hal.inria.fr/inria-00301410.

[30] A. PLSEK, P. MERLE, L. SEINTURIER. *A Real-Time Java Component Model*, in "Proceedings of the 11th International Symposium on Object/Component/Service-oriented Real-Time Distributed Computing (ISORC'08), Orlando, Florida, USA", Acceptance rate : 33%, 29/85, IEEE CS Press, May 2008, p. 281–288, http://hal.inria.fr/inria-00222039.

[31] D. ROMERO, C. A. PARRA ACEVEDO, L. SEINTURIER, L. DUCHIEN, R. CASALLAS. *An SCA-Based Middleware Platform for Mobile Devices*, in "2008 Middleware for Web Services (MWS'08) Workshop at EDOC2008, Munich, Germany", September 2008, http://hal.inria.fr/inria-00311807.

[32] P. SRIPLAKICH, X. BLANC, M.-P. GERVAIS. *Collaborative Software Engineering on Large-scale models: Requirements and Experience in ModelBus*, in "Proceedings of the 23st Annual ACM Symposium on Applied Computing (SAC'08), Fortaleza, Ceara, Brazil", Rank (CORE) : B, March 2008, p. 674–681.

[33] P. SRIPLAKICH, G. WAIGNIER, A.-F. LE MEUR. *Enabling Dynamic Co-Evolution of Models and Runtime Applications*, in "Proceedings of the 1st IEEE International Workshop on Model-Driven Development of Autonomic Systems (MDDAS'08), Turku, Finland", IEEE Computer Society, July 2008, p. 1116–1121, http://hal.inria.fr/inria-00311581.

[34] G. WAIGNIER, P. SRIPLAKICH, A.-F. LE MEUR, L. DUCHIEN. *A Model-Based Framework for Statically and Dynamically Checking Component Interactions*, in "Proceedings of the ACM/IEEE 11th International Conference on Model Driven Engineering Languages and Systems (MODELS'08), Toulouse, France", Lecture Notes in Computer Science, Rank (CORE) : A, vol. 5301, Springer-Verlag, October 2008, p. 371–385, http://hal.inria.fr/inria-00311584.

[35] G. WAIGNIER, A.-F. LE MEUR, L. DUCHIEN. *Architectural Specification and Static Analyses of Contractual Application Properties*, in "Proceedings of the 4th International Conference on the Quality of Software-Architectures (QoSA'08), Karlsruhe (TH), Germany", S. BECKER, F. PLASIL, R. REUSSNER (editors), Lecture Notes in Computer Science, Rank (CORE) : A, vol. 5281, Springer-Verlag, October 2008, p. 152-170, http://hal.inria.fr/inria-00284514.

[36] A. ZITOUNI, L. SEINTURIER, M. BOUFAIDA. *Contract-Based Approach to Analyze Software Components*, in "Workshop on UML&AADL at ICECCS'08", 2008, p. 237-242, http://hal.inria.fr/inria-00271533.

National Peer-Reviewed Conference/Proceedings

[37] G. DUFRÊNE, L. SEINTURIER. *Un ADL pour les Architectures Distribuées à Composants Hétérogènes*, in "2eme Conférence Francophone sur les Architectures Logicielles (CAL'08), Montréal, Quebec, Canada", Editions Cépaduès, March 2008, p. 79–90, http://hal.inria.fr/inria-00271530.

[38] N. MOHA, F. KHOMH, Y.-G. GUÉHÉNEUC, L. DUCHIEN, A.-F. LE MEUR. *Génération automatique d'algorithmes de détection des défauts de conception*, in "Proceedings of the French Conference on Langages et Modèles à Objets (LMO'08), Montreal, Canada", March 2008.

Workshops without Proceedings

[39] A. BOCQUET. *Infrastructure logicielle multi-modèles pour l'accès à des services en mobilité*, in "EuroDoc'Info'08", 2008.

[40] A. BOCQUET, C. GRANSART. *Approche et solution multi-modèles aux déconnexions satellitaires pour l'accès Internet à bord des trains*, in "Journées des Applications Spatiales, Toulouse Space'Show", 2008.

[41] M. MALOHLAVA, A. PLSEK, F. LOIRET, P. MERLE, L. SEINTURIER. *Introducing Distribution into a RTSJ-based Component Framework*, in "Proceedings of the 2nd Researcher Workshop on Real-Time Computing (JRWRTC'08), Rennes, France", October 2008, http://hal.inria.fr/inria-00327342.

[42] G. WAIGNIER, P. SRIPLAKICH, A.-F. LE MEUR, L. DUCHIEN. *A Framework for Bridging the Gap Between Design and Runtime Debugging of Component-Based Applications*, in "Proceedings of the 3rd International Workshop on Models at Runtime, Toulouse, France", October 2008, http://hal.inria.fr/inria-00321598.

Scientific Books (or Scientific Book chapters)

[43] O. BARAIS, J. LAWALL, A.-F. LE MEUR, L. DUCHIEN. *Software Architecture Evolution*, in "Software Evolution", Springer-Verlag, February 2008, p. 233–262.

Research Reports

[44] S. DIETZE, C. PEDRINACI, A. GUGLIOTTA, P. MERLE, I. MARTÍNEZ, G. Á. REY, C. R. MORENO, M. VILLA, S. ABEL. *Service Provisioning Platform Design*, Deliverable, n^{o} 2.1.1, FP7 IST SOA4All Integrated Project, August 2008.

[45] C. GRANSART, A. BOCQUET. *Modèles de modes de communication*, Livrable, n^{o} 1.2.2, Projet ARA REVE, 2008.

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

Other Publications

[47] P. CARTON. *Modèle SCA : Développement en environnement ubiquitaire*, Masters thesis, Université des Sciences et Technologies de Lille (USTL), Laboratoire d'Informatique Fondamentale de Lille (LIFL), Villeneuve d'Ascq, France, September 2008.

[48] A. TIBERGHIEN. *Description et déploiement de grandes architectures à composants*, Masters thesis, Université des Sciences et Technologies de Lille (USTL), Laboratoire d'Informatique Fondamentale de Lille (LIFL), Villeneuve d'Ascq, France, June 2008.

References in notes

[49] *CORBA Component Model*, OMG, February 1999, http://www.omg.org.

[50] *Enterprise Java Beans*, Sun Microsystems, 1997, http://www.javasoft.com/products/ejb.

[51] R. E. FILMAN, T. ELRAD, S. CLARKE, M. AKSIT (editors). *Aspect-Oriented Software Development*, Addison-Wesley, 2005.

[52] É. BRUNETON, T. COUPAYE, M. LECLERCQ, V. QUÉMA, J.-B. STEFANI. *The* FRACTAL *Component Model and its Support in Java*, in "Software: Practice and Experience – Special issue on Experiences with Auto-adaptive and Reconfigurable Systems", John Wiley & Sons, vol. 36, n^{o} 11-12, August 2006, p. 1257–1284.

[53] T. COUPAYE, V. QUÉMA, L. SEINTURIER, J.-B. STEFANI. *Le système de composants Fractal*, in "Intergiciel et Construction d'Applications Réparties", sardes.inrialpes.fr/ecole/livre/pub/, January 2007.

[54] K. CZARNECKI, U. EISENECKER. *Generative Programming - Methods, Tools, and Applications*, Addison-Wesley, 2000.

[55] R. GABRIEL, M. DEVOS, B. FOOTE, G. STEELE, J. NOBLE. *Objects Have Failed*, Object-Oriented Programming, Systems, Languages and Applications (OOPSLA'02). Seattle, USA., November 2002.

[56] E. GAMMA, R. HELM, R. JOHNSON, J. VLISSIDES. *Design Patterns: Elements od Reusable Object-Oriented Software*, Addison-Wesley Professional Computing Series, Addison-Wesley Publishing Company, New York, NY, 1995.

[57] G. KICZALES, J. DES RIVIERES, D. G. BOBROW. *The Art of Metaobject Protocol*, MIT Press, Cambridge, MA, USA, 1991.

[58] G. KICZALES, E. HILSDALE, J. HUGUNIN, M. KERSTEN, J. PALM, W. GRISWOLD. *An Overview of AspectJ*, AspectJ white paper submitted to the European Conference on Object-Oriented Programming (ECOOP'01), 2001.

[59] G. KICZALES, J. LAMPING, A. MENDHEKAR, C. MAEDA, C. LOPES, J. LOINGTIER, J. IRWIN. *Aspect-Oriented Programming*, in "Proceedings of the 11th European Conference on Object-Oriented Programming (ECOOP'97)", Lecture Notes in Computer Science, vol. 1241, Springer, June 1997, p. 220-242.

[60] F. LOIRET, D. SERVAT, L. SEINTURIER. *A First Experimentation on High-Level Tooling Support upon Fractal*, in "Proceedings of the 5th International ECOOP Workshop on Fractal Component Model (Fractal'06), Nantes, France", July 2006.

[61] D. MCILROY. *Mass Produced Software Component*, Technical report, Report on the NATO Software Engineering Conference, 1968.

[62] P. K. MCKINLEY, S. M. SADJADI, E. P. KASTEN, B. H. C. CHENG. *A Taxonomy of Compositional Adaptation*, http://citeseer.ist.psu.edu/mckinley04taxonomy.html.

[63] T. P. MORAN, P. DOURISH. *Introduction to This Special Issue on Context-Aware Computing*, in "Human-Computer Interaction", vol. 16, n^o 2–4, 2001, p. 87–95.

[64] C. F. NOGUERA GARCIA, R. PAWLAK. *AVal: an Extensible Attribute-Oriented Programming Validator for Java*, in "Proceedings of the 6th IEEE International Workshop on Source Code Analysis and Manipulation (SCAM'06), Philadelphia, PA, USA", IEEE Computer Society, September 2006, p. 175–183.

[65] C. F. NOGUERA GARCIA, R. PAWLAK. *AVal: an Extensible Attribute-Oriented Programming Validator for Java*, in "Software Maintenance and Evolution: Research and Practice", Impact factor (SCI06) : 0,636, July 2007.

[66] OPEN SOA. *SCA Service Component Architecture — Assembly Model Specification*, 1.0, November 2007.

[67] R. PAWLAK. *Spoon: Compile-time Annotation Processing for Middleware*, in "IEEE Distributed Systems Online", vol. 7, n^o 11, November 2006.

[68] N. PESSEMIER, L. SEINTURIER, T. COUPAYE, L. DUCHIEN. *A Model for Developing Component-based and Aspect-oriented Systems*, in "Proceedings of the 5th International Symposium on Software Composition (SC'06), Vienna, Austria", Lecture Notes in Computer Science, vol. 4089, Springer-Verlag, March 2006, p. 259–273.

[69] D. C. SCHMIDT. *Model-Driven Engineering*, in "IEEE Computer", vol. 39, n^o 2, February 2006, http://www.truststc.org/pubs/30.html.

[70] L. SEINTURIER, N. PESSEMIER, L. DUCHIEN, T. COUPAYE. *A Component Model Engineered with Components and Aspects*, in "Proceedings of the 9th International SIGSOFT Symposium on Component-Based Software Engineering (CBSE'06), Västeras, Sweden", Lecture Notes in Computer Science, vol. 4063, Springer, June 2006, p. 139–153.

[71] T. STRANG, C. LINNHOFF-POPIEN. *A Context Modeling Survey*, in "Workshop on Advanced Context Modelling, Reasoning and Management at the Sixth International Conference on Ubiquitous Computing (UbiComp)", 2004.

[72] D. THOMAS. *Reflective Software Engineering - From MOPS to AOSD*, in "Journal of Object Technology", vol. 1, n^{o} 4, 2002, p. 17–26.

[73] M. WIRSING, M. HOLZL. *A Taxonomy of Compositional Adaptation Beyond-The-Horizon, Thematic Group 6: Software-Intensive Systems*.