INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Team Adam

Adaptive Distributed Applications and Middleware

Futurs

THEME COM

Table of contents

1. Team

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).

Head of project-team
Laurence Duchien [Professor, USTL, Secondment INRIA, HdR]

Vice-head of project-team
Philippe Merle [Research Associate (CR1)]

Administrative assistant
Corinne Davoust

INRIA project-team staff
Maja D'Hondt [Research Associate (CR2), until June 30th 2007]
Stéphane Ducasse [Research Director (DR2), since September 1st 2007, HdR]

CNRS project-team staff
Areski Flissi [Research Engineer]

University project-team staff
Anne-Françoise Le Meur [Associate Professor, USTL]
Lionel Seinturier [Professor, USTL, HdR]

External collaborators
Jean-Marc Geib [Professor, USTL, HdR]
Christophe Gransart [Research Associate (CR1), INRETS]

Post-doctoral fellows
Johan Fabry [INRIA grant, until August 31th 2007]
Prawee Sriplakich [INRIA grant, since October 1st 2007]
Ellen Van Paesschen [ERCIM grant, until August 31th 2007]

Ph.D. students
Hani Abdeen [KHAN grant, since October 1st 2007]
Aurélien Bocquet [INRETS grant]
Dolorés Diaz [NorSys CIFRE grant]
Jérémy Dubus [MERT grant]
Guillaume Dufrêne [ANR TLog FAROS grant]
Frédéric Loiret [CEA grant, until October 31th 2007]
Naouel Moha [University of Montreal, co-supervision]
Carlos-Francisco Noguera-Garcia [IST AOSD Network of Excellence grant]
Carlos Andres Parra Acevedo [INRIA / Région Nord-Pas de Calais grant, since October 1st 2007]
Nicolas Pessemier [France Télécom grant, until June 30th 2007]
Ales Plsek [INRIA CORDIS grant]
Daniel Romero [EIFFEL grant, since October 1st 2007]
Mathieu Suen [CNRS/DGA grant, since October 1st 2007]
Guillaume Waignier [MERT grant]

Project-team technical staff
Yann Davin [Development engineer, INRIA ODL, since September 1st 2007]
Nicolas Dolet [Expert engineer, ANR TLog JOnES]
Damien Fournier [Expert engineer, ANR TLog SCOrWare, since March 1st 2007]
James Manley [Expert engineer, ITEA S4ALL, until June 30th 2007]
Frédéric Loiret [Expert engineer, ANR TLog Flex-eWare, since November 1st 2007]

Invited researchers
Yann-Gaël Guehéneuc [University of Montreal, March 2007]
Tom Mens [University of Mons, December 2006 until September 30th 2007]

Student internships

Carlos Andres Parra Acevedo [March - July 2007]
Loïc Schmidt [Master Research, February - June 2007]
Alban Tiberghien [Master, June - September 2007]

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 need 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 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 CBSE (Component-Based Software Engineering), AOSD (Aspect-Oriented Software Development) or CAC (Context-Aware Computing) to build distributed adaptive software systems involving 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 it must be usable in various environments, in accordance with the unavoidable evolution in 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 could 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 adding or withdrawing functionalities in accordance with the user needs. Thus the life-cycle has to define and to support the adaptation expression in each software life-cycle level. 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 [77]) that can adjust and respond to changes in its environment, evolving requirements, removal of obsolete technologies or introduction of new technologies, and new knowledge [65]. 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 the 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 CBSE (Component-Based Software Engineering) and AOSD (Aspect-Oriented Software Development) 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 mastering adaptation for 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. For reaching this challenge, techniques and formalisms are needed to assess the impact of such changes, to synchronize models in 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 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 [61], patterns [58], generative programming [51] or reflective approaches [60], [76] are already employed and co-exist in existing 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 [69] and AOKell [74]. 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. PDA's, phones or sensors have small capacities in terms of battery, memory, or computation units and the application and its run-time platform design will be designed and distributed according to these capacities.

2.3. Highlights of the year

- **Spoon** is a transformation suite based on annotations for Java applications. It provides a core API and associated tools for static analysis and generative programming within Java 5+ environment. It specifies program transformations using a well-typed generative programming technique called

Spoon Templates. Spoon also provides an Eclipse plugin (SpoonJDT) for packaging validations and transformations into compilation components. It is an open source software and it is the second downloaded software on the INRIA Gforge. For more details, see section 5.10 and [21].

- **FDF** is a component-based software framework to deploy distributed and heterogeneous software systems. This framework has been successfully integrated into three industrial open source projects hosted by the OW2 international consortium leading open source middleware: 1) **JOnAS** a Java Enterprise Edition (JEE) application server, 2) **JASMINe** an Service-Oriented Architecture (SOA) platform management tool, and 3) **PEtALs** an Enterprise Software Bus (ESB) for SOA. In [56], FDF has been successfully experimented for the deployment of large distributed CORBA-based software systems composed of more than 4500 application servers deployed on one thousand nodes of Grid'5000, the french grid infrastructure dedicated to computer science research. For more details, see sections 5.6 and 6.3.
- **Cappucino** (Construction et Adaptation d'aPPlications Ubiquitaires et de Composants Intergiciels en enviroNnement Ouvert pour l'industrie du commerce) is a new 36-months project started in September 2007. This project is funded by the cluster (pôle de compétitivité) of trade business (Industrie du commerce). The partners are Auchan, SI3SI, NorSys, INT/GET, and INRIA. This project addresses issues which are related to modeling the context-aware information for ambient computing and to take into account this information into runtime platforms. The adaptation of the components and the context evolution will be studied throughout the complete lifecycle of the application. For more details, see at http://cappucino.oqube.com and section 8.2.

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 [62], [48] 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 [57], software components [64] are with other artifacts such as aspects, one of the approaches which 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 quick

survey of the literature returns more than 20 different models (starting with the most well-known such as EJB [47] and CCM [46]), and 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 [66]. 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 that 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 [75]. 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 [73] 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 a means for managing model inconsistencies.

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.

[1] These terms are more or less equivalent.

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 which is desirable for end-user applications, but also for the middleware platform which executes these applications. So far, 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 two-fold. 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. For that 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 ITEA S4ALL and the ANR TLog JOnES projects (see sections 8.2 and 8.3), 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 involve in several projects such as the ANR TLog FlexeWare and the ANR ARA REVE (see section 8.2). Several challenges must be address 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 (see section 8.2). 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, the CPER MOSAIQUES project and the INRETS collaboration (see sections 8.1 and 8.2). 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 service, can be adapted to support there 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-paradigms platforms. We list our actual software that we intend to continue and to extend in the ADAM project-team.

5.2. AOKell

Keywords: *AOP*, *Fractal*.

Participants: Nicolas Pessemier, Lionel Seinturier [correspondant].

AOKell is a reflective and open implementation of the Fractal component model. So far, software components are used in various application domains (from embedded systems to e-commerce web sites). However the model of a component is different in each of these domains with solutions such as EJB, CCM, .NET, Accord/UML, ArchJava, OpenCOM, K-Component, etc. This variety hinders the broad adoption of component based techniques. The challenge we are addressing with AOKell is to design and implement a component model adaptable to various application domains. To achieve this goal, AOKell is a reflective component model with two dimensions: the business dimension and the control dimension. The former is concerned with the programming of application functionalities, while the latter is concerned with control and non-functional services needed by the application. The control dimension is fully programmable in AOKell with the same artifacts (component, binding, components assembly) that are available for the business dimension. The integration of these two dimensions is achieved with AOP. The work on AOKell is conducted in the context of a France Telecom R&D research grant (see section 7.1).

AOKell is LGPL open source software available at http://fractal.objectweb.org/aokell. AOKell is being developed since 2004 and is an open-source project which is distributed in the context of the Fractal project hosted by the OW2 (previously ObjectWeb) consortium. Known usages of AOKell by external people include the development of the OSA platform for discrete event simulation (INRIA MASCOTTE project), and the development of the Dynaco platform for adaptive grid computing (INRIA PARIS project). AOKell is written in Java and is composed of 20,000 lines of code.

5.3. FAC

Keywords: *AOP*, *Fractal*.

Participants: Nicolas Pessemier, Lionel Seinturier [correspondant].

The work on FAC (Fractal Aspect Component) is conducted in the context of a France Telecom R&D research grant (see section 7.1) and was the subject of Nicolas Pessemier's Ph.D. thesis [15]. FAC is an extension of the Fractal component model for AOP. The purpose of FAC is to provide a programming model where components and aspects are first-class entities which can be assembled and composed seamlessly by designers and developers. FAC is innovating in the sense that, on the international stage, aspects have been studied so far only at the object level. We believe that the issues of code tangling and scattering exist at a higher level of granularity that of components. With FAC, we wish to provide one of the first models and its corresponding implementation for modularizing crosscutting concerns in component-based applications. The longer term goal of FAC is also to provide a unified framework where crosscutting concerns can be modularized at different levels of granularity: object, component and architecture.

5.4. Fractal

Keywords: *Component model*.

Participants: Yann Davin, Philippe Merle [correspondant], Nicolas Pessemier, 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 to middleware platforms and to graphical user interfaces [17]. Fractal has been designed by both INRIA and France Telecom R&D. Fractal is also an open source software project hosted by the OW2 international consortium and is available at http://fractal.objectweb.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 (see section 5.2 for more details).
- **Fraclet** is an annotation-based programming model for Fractal (see section 5.5 for more details).
- **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.

For year 2007, our new contributions were **Juliac**, **Koch**, **Fractal Distribution**, **Fractal Documentation**, and **Fractal Eclipse Plugin** modules.

5.5. Fraclet

Keywords: *Attribute-Oriented Programming*, *Component-Based Programming*, *Fractal*.

Participants: Philippe Merle [correspondant], Nicolas Pessemier.

Fraclet is an attribute-oriented programming model for developing reliable Fractal components. Fraclet is both an implementation of the Fractal specifications conformed to the level 0.1, and an annotation framework for the Fractal component model. Fraclet is composed of eight annotations and five plugins to generate automatically various artifacts required by the Fractal component model. Annotations provide an abstract way to describe the component meta-information directly in the source code of the content class. Fraclet plugins generate either Fractal component glue, Fractal ADL definitions or Monolog configurations.

Two implementations of the Fraclet annotation framework exist: Fraclet-XDoc and Fraclet-Annotation.

- Fraclet-XDoc uses the XDoclet generation engine and Javadoc-like annotations to produce the various artifacts required by the Fractal component model.
- Fraclet-Annotation uses the Spoon transformation tool and Java 5 annotations to enhance the handwritten program code with the non-functional properties of the Fractal component model.

Fraclet is used into several software projects based on the Fractal component model, i.e. the ADAM project-team Fractal Deployment Framework and GoTM projects, the GET/INT's COSMOS project, the INRIA SARDES's DREAM project (see at http://dream.objectweb.org), and the OW2 PEtALS project (see at http://petals.objectweb.org).

Fraclet is LGPL open source software available at http://fractal.objectweb.org/fraclet.

5.6. Fractal Deployment Framework

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

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

The work on Fractal Deployment Framework (FDF) is the subject of Jérémy Dubus's Ph.D. thesis and is a component-based software framework to facilitate the deployment of distributed and heterogenous applications on networked systems. FDF implements the virtual machine of the DeployWare framework (see section 6.3). FDF supports any kind of deployment activities like uploading, installing, configuring, starting, stopping and uninstalling any software. 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 Apache Tuscany runtime, JBI components and OW2 PEtALS container), 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 [56], FDF has been successfully experimented for the deployment of OpenCCM middleware and CORBA component-based applications on one thousand nodes of Grid'5000, the french grid infrastructure dedicated to computer science research.

FDF is designed with the OW2 Fractal component model, and is implemented on top of its Java-based tools (Julia or AOKell, Fraclet, Fractal Explorer).

FDF is LGPL open source software available at http://gforge.inria.fr/projects/fdf. FDF is being developed since 2006 and is an open-source project hosted by the INRIA Forge (http://fdf.gforge.inria.fr). 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.objectweb.org) and the JASMINe SOA platform manangement tool (http://jasmine.objectweb.org) developed by Bull SA, and the PEtALS JBI implementation developed by EBM WebSourcing (http://petals.objectweb.org), both are open-source projects hosted by the OW2 (previously ObjectWeb) consortium. FDF is composed of 25.000 lines of Java code and 21.000 lines of Fractal ADL code.

5.7. FIESTA

Keywords: *components*, *separation of concerns*, *software architecture*.

Participants: Anne-Françoise Le Meur, Guillaume Waignier [correspondant].

This work around FIESTA, TranSAT and SafArchie Studio is the subject of Olivier Barais and Guillaume Waignier's Ph.D. theses.

FIESTA aims to be a software architecture-centric IDE that implements a safe and adaptable architecture model. It will be based on some modules that come from SafArchie Studio and TranSAT and will be able to handle and check the adaptability of a software architecture. This software will be developed in the context of Guillaume Waignier's thesis. The long term goal of FIESTA is to provide a unified IDE to design, transform and control the dynamic evolution of component-based applications and to manage the co-evolution between models and execution platform. FIESTA is being developed since 2006. The current version is available on INRIA Gforge.

SafeArchie Studio is composed of three parts. The first one is a set of extensions for ArgoUML to transform the UML case tool into a software architecture case tool. It allows the architect to graphically design a component-based software architecture. The second part analyses the software architecture consistency. This consists of checking the structural compatibility between bound ports and the behavioral compatibility between bound components with respect to the SafArchie model. The third part is a skeleton generator for the ArchJava language or the Fractal component model. This generator provides a first level connection between the design and the implementation of a component-based system.

TranSAT tools are designed as extensions of SafArchie Studio. They aim at providing IDE support to allow the architect to stepwise design a software architecture. TranSAT is associated with three main tools:

- new diagram editors to specify new software architecture patterns
- a compiler for the software architecture pattern. It performs static analyses to guarantee that the pattern will not compromise the consistency of the software architectures that will be modified.
- a transformation processor. It integrates a new pattern into a software architecture.

The transformation processor includes a first module to find all the join points that satisfy the join point mask constraints, a second module to dynamically verify the consistency of the resulting software architecture and finally a transformation engine to perform the transformations on the join points selected by the architect. We have three implementations for the processor, each of which uses a specific tool: AGG based on graph theory, CIAO Prolog based on predicate logic, and DROOLS based on production rules. The longer-term goal of TranSAT tools consists of comparing the different technologies used to implement an efficient transformation operator. Furthermore, we are going to experiment applying TranSAT' s transformation process to other kinds of diagrams such as class diagrams or deployment diagrams. The work on SafArchie Studio and TranSAT are a part of Olivier Barais's Ph.D. thesis [49] and was mainly developed by Missi Tran-Anh and Olivier Barais.

SafArchie Studio and TranSAT are being developed since 2004 and are accessible LGPL open source software available at http://transat.gforge.inria.fr. They are composed of 20.000 lines of code and they are written in XML, Java, Prolog, AGG Transformation and DROOLS rules.

5.8. GoTM

Keywords: *Component-Based Software Framework*, *Middleware Transaction Services*.

Participants: Nicolas Dolet, Philippe Merle [correspondant].

GoTM is a Fractal component-based software framework to build middleware transaction services. This work around GoTM was the subject of Romain Rouvoy's Ph.D. thesis [71]. GoTM is composed of an extensible set of Fractal components providing basic building blocks (Transaction, Resource, Coordination, Concurrency, etc.) to build various transaction models and services (OMG OTS, Sun JTA, etc.) [71]. This framework provides personalities to build Java Transaction Services (JTS) and Object Transaction Services (OTS). To deal with the transaction demarcation concern, GoTM provides the Open Transaction Demarcation Framework (OTDF) as a dedicated personality. The JTS and OTDF personalities are currently integrated into the ObjectWeb JOnAS application server.

The GoTM framework can be used to build heterogeneous transaction services [71]. Heterogeneous services support multiple transaction standards simultaneously without the performance degradation implied by the use of a coordination protocol.

The GoTM framework implements various 2-Phase Commit protocols (e.g., 2PC, 2PCPA, 2PCPC, etc.). It dynamically adapts the active transaction protocol to provide better transaction completion time depending on the commit/abort rate statistics [72].

The GoTM component-based software framework is designed on top of the ObjectWeb Fractal component model and is implemented on top of the ObjectWeb Julia reference implementation. Since recently, GoTM uses the AOKell software supported by the Adam project-team (see section 5.2). AOKell implements the Fractal component model using Aspect-Oriented Programming (AOP). This choice provides to GoTM the complementarity of technologies like aspects and components to build middleware solutions with added values. GoTM also uses the Fractal Explorer software to build its management tool. GoTM provides various explorer plugins (GoTM, JTS, OTS, etc.). Thus, GoTM consoles can manage heterogeneous GoTM transaction services. Finally, the GoTM OTS personality aims to be integrated in the OpenCCM software developed in the Jacquard project.

On the topic of Transactions and Components, the GoTM and AOKell projects are also collaborating to provide transparent transaction support to Fractal components.

GoTM will be integrated in the next Enterprise Service Bus (ESB) solution developed by the ObjectWeb consortium in the context of the ANR TLog JOnES project (see section 8.2).

GoTM is being developed since 2004 and is LGPL open-source project hosted by the OW2 (previously ObjectWeb) consortium and is available at http://gotm.objectweb.org. GoTM is written in Java and is composed of 25.000 lines of code.

5.9. Tinfi

Keywords: *Service Component Architecture*.

Participants: Damien Fournier, Philippe Merle, Lionel Seinturier [correspondant].

Tinfi (Tinfi Is Not a Fractal Implementation) is a runtime platform for the SCA (Service Component Architecture) 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. The originality of Tinfi is to be architectured as a personality of the Fractal component model. Thanks to the openess 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 which own both a Fractal personality and a SCA personality. As far as we know, this result is original and is the first one to concretly demonstrates that Fractal is open and flexible enough to implement different component personalities. Tinfi has been implemented by reusing modules developed in the context of the Fractal project, and among others, the Juliac Fractal compiler. Tinfi is developed in the context of the ANR TLog SCOrWare project (see section 8.2).

5.10. Spoon

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

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

Spoon [68] 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 phases, 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 [68].
- AVal: a work done within the Ph.D. of Carlos Nogera and that consists of a framework for validating annotation sets forming DSLs. This work was published in [67].
- 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 ObjectWeb component model. This work was published in [70].
- AOKell: an alternative implementation of the ObjectWeb Fractal container using AOP and Spoon.
- 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. See 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 started by Carlos Noguera and Johan Fabry (see section 6.5).

Spoon is an 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. Architectural Patterns with Fractal ADL

Participants: Philippe Merle, Rouvoy Romain [University of Oslo].

The intensive use of components in software developments increases the size of architecture descriptions and both associated syntactic and structural errors. Nevertheless, several architectural patterns can be identified in these architectures. The growing complexity of these descriptions comes with an increase of the concerns involved in the architectures. Moreover, these concerns are often tangled in the same description. In [40], we introduce an extension of the Fractal Architecture Description Language (ADL) to describe and verify architectural patterns, which makes more easier and reliable the architecture descriptions. To support the evolution of architecture descriptions and reduce syntactic and semantic errors, we propose to integrate two new operators in Fractal ADL in order to generate and verify architectural patterns. These operators use the FPath language to describe the requests filtering the content of the architecture. The result of these requests are used by our operators to generate some architectural patterns and to detect assembly errors.

6.2. Adaptable Transaction Services

Participants: Philippe Merle, Romain Rouvoy [University of Oslo].

The evolution of existing transaction services is limited because they are tightly coupled to a given transaction standard, implement a dedicated commit protocol, and support a fixed kind of applicative participants. The next challenge for transaction services will be to deal with evolution concerns. This evolution should allow developers to tune the transaction service depending on the transaction standard or the application requirements either at design time or at runtime.

In [22], we introduce the common approach that we have defined to build various evolutionary transaction services. This common approach is based on the use of microcomponents and design patterns, whose flexibility properties allow transaction services to be adapted to various execution contexts. This approach is applied in our GoTM framework (see section 5.8) that supports the construction of transaction services implementing several transaction standards and commit protocols. We argue that using fine-grained components and design patterns to build transaction services is an efficient solution to the evolution problem and our past experiences confirm that this approach does not impact the transaction service efficiency.

6.3. DeployWare

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

One of the most technical and problematic phase of the software lifecycle is the deployment phase. System administrators, in charge of deploying distributed applications, have to adapt themselves to the evolution of both machines and software technologies. These administrators have to face three kinds of heterogeneity. The first facet of this heterogeneity affects support hardware on which applications are deployed, such diversity affecting various properties of machines (computational capacity and/or memory, operating systems, remote access and file transfer protocols). The second facet of this heterogeneity concerns paradigms that are used to develop business applications. These paradigms change over the innovations (*e.g.* object-oriented, component-oriented, aspect-oriented programming or model-based software engineering). Finally the third kind of heterogeneity affects the variety of technologies that are available for each of these paradigms. As an example, we can cite SCA or EJB as two different technologies of the component-based paradigm. The system administrators must then: 1) learn every paradigm involved in the system, 2) master every deployment procedures specific to a given technology (which means deploying libraries, application servers, components business applications as well as their relationship), and finally 3) adapt these procedures to the characteristics of hardware supposed to execute them. In addition, the administrator must cope with the emergence of a new kind of network of machines that we call *open distributed environments*. Typical examples of such environments are Ambient and Grid computing, where machines can appear/disappear at runtime in an unpredictable way.

We want to streamline the process of deployment, so that it is possible to abstract it, provide tooling for it and automate it as most as possible in the context of an approach conform to *Model Driven Architecture* (MDA) of the OMG. For this we propose DeployWare, a model-based approach for the description of autonomic distributed systems.

In this approach, we propose a generic deployment metamodel cut into three parts. The first part offers concepts used to describe in a generic the process of deploying software for a given technology. The second part provides generic concepts used to compose software instances as described in the first part. Finally, the third part of the metamodel concerns autonomic policies (in accordance with the precepts of Autonomic Computing), which, at runtime, adjust the deployment of the application to the fluctuations in the network, according to high-level instructions given by administrators. The definition of a metamodel also allows to define concepts making possible static validations on deployment procedures, as we have shown in [26]. We have shown that sometimes in a syntactically correct autonomic deployment program with also correct types, some errors could occur at runtime (e.g. side-effects of deployment, software that cannot be undeployed). Some important behavioural validations, resolving that kind of problems, have been made possible using specific metamodel concepts. In addition, it is demonstrated that autonomic policies defined in accordance with the control loop of the Autonomic Computing, can lead to inconsistencies in autonomic behaviour. Indeed chained policies execution, for example, may lead to the emergence of a cycle and then become infinite. The overall system state becomes incoherent, despite the fact that the system was expected to be autonomic. The concepts that we propose in our metamodel for autonomic policies allow a static verification of policies, to ensure that the overall behaviour of the system does not contain a cycle. Further checking on the policies of autonomy are being actually studied.

DeployWare makes the execution of the deployment procedures possible thanks to the FDF execution platform (see section 5.6), which reifies deployment process through component-based architecture, and abstract this procedure from hardware constraints, distribution constraints but also dependencies constraints between software of a system. DeployWare models can, using model transformation techniques, be projected into an executable FDF software architecture, which represents the deployment machine of the whole system.

6.4. Aspect-Oriented Programming and Components

6.4.1. Fractal Aspect Component

Participants: Laurence Duchien, Nicolas Pessemier, Lionel Seinturier.

FAC (Fractal Aspect Component) is a framework for unifying component-based and aspect-oriented software development. This is the result of the Ph.D. thesis of Nicolas Pessemier [15] which has been defended on June 27, 2007. This Ph.D. took place in the context of a research contract (CRE) with France Telecom R&D (see section 7.1). Two main and new results have been produced by this Ph.D. First, the concepts of aspect-oriented programming (AOP) which were so far only available for object-oriented languages, have been adapted for component-based (CB) languages. Hence, the core notions of AOP (join point, pointcut, advice, aspect) have been adapted and transposed to the level of software components. A second result is the reinterpretation of the notions of CB languages and of software architecture languages (ADL) in the light of AOP. This result enables the design of multi-layered software architectures where each layer is a view on the overall architecture focusing on a particular functionality: business core, security, data persistence, transaction management, GUI, etc. This enables a clear separation of concerns and facilitates the design of the overall architecture which can be defined by aggregating layers which have been designed separately.

6.4.2. Complex Aspects

Participants: Johan Fabry, Nicolas Pessemier.

When considering aspects that capture a large and complex cross-cutting concern, we have established that these aspects themselves suffer from a lack of important software engineering properties. Typically, using current-day popular aspect languages, the aspects have a low degree of modularity, and can themselves also be subject to tangled and scattered code. As a result of this, development and maintenance of the code of the aspect is negatively impacted by the low separation of concerns. We have investigated how we can improve the software engineering properties of the aspects, so that they themselves have a clean modularisation. We have determined that to obtain a high degree of separation of concerns of the aspect, more symmetrical aspect languages are required. More concretely, we have experimented with the aspect language FAC, and have been able to obtain a clean separation of concerns within the case study of advanced transaction management.

6.5. Software Engineering for Ambient Intelligence

6.5.1. COSMOS

Participants: Denis Conan [GET/INT], Romain Rouvoy [University of Oslo], Lionel Seinturier.

COSMOS [25] 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 easy design and reuse adaptation policies for context-aware applications.

6.5.2. Ambient-Oriented Programming in Fractal

Participants: Philippe Merle, Ales Plsek, Lionel Seinturier.

Ambient intelligence deals with applications which can react and be adapted to their environment. In [37] we proposed a solution for transparently adapting component frameworks for communication in such environments. These communications pose several challenges, and among others, the fact that they are volative with peers entering and leaving frequently the environments. The programming artefacts must then be able to support frequent connections and disconnections. We have thus proposed a service for handling these communications. This service has been applied to the Fractal component model. We took advantage of the openess of this model which enables to customize the non-functional services which are offered to hosted components in order to insert this service.

6.5.3. Attribute-Oriented Programming

Participants: Johan Fabry, Carlos-Francisco Noguera-Garcia.

Ambient Intelligence applications face the fundamental property of the domain that network connections and disconnections are the norm, and no longer the exception. As devices come in and out of range of each other, connections between these devices will be created and be lost. Connections have therefore become volatile. Supporting connection volatility inside an application is however a difficult task due to the cross-cutting nature of this concern, leading to the code that handles this concern to be tangled and scattered. This low separation of concern makes these applications hard to develop and maintain. We have developed a new abstraction mechanism for connection volatility, called tagged futures, that untangles this concern from the base code of the application. As a result, support for connection volatility can now be added to a distributed application without requiring this application to be intrusively modified in an extensive way. Our experiments have been realised using the Spoon source code processing toolkit. Spoon Graffiti [29], the additional tools we developed, is composed of the different templates, source code processors, and additional infrastructure required. It implements an extension of the well-known distributed systems construct, futures. This in order to address the issue of connection volatility in AmI applications. The second one, CARBO [36], leverages attributes to generate context aware applications parting from a model of the environment in which the applications reside.

During the first half of 2006 we have developed an approach to validate annotation frameworks called AVal. A paper [21] describing this work was published in the Journal of Software Maintenance and Evolution in July of 2007.

6.5.4. Context-Aware Modelling

Participants: Maja D'Hondt, Johan Fabry, Carlos-Francisco Noguera-Garcia, Carlos Andres Parra Acevedo, Ellen Van Paesschen.

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.

Our goal is to bridge the gap between the computational level and the domain level in Ambient Computing. In order to achieve this, several challenges are addressed. First of all, the knowledge of this domain is represented in models and they have to be synchronised with the computational level. The models represent concepts of the application domain, but also general concepts related to context. Secondly, we provide a rule language in order to specify how the behaviour of the application has to adapt to changes in the context. The rules are defined in terms of the models. This approach and a preliminary implementation are presented in [36] and is called CARBO.

An extension to the aforementioned approach in order to be able to handle distributed contexts is based on the concept of [30]. The CARBO orchestration engine, intended for services orchestration in ubiquitous computing, suffers from the assumption that context is centralised in the device performing the orchestration. 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 [35].

6.6. Software Evolution

6.6.1. Integration of New Functionalities into Software Architectures

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

In the context of Guillaume Waignier's Master thesis, we investigated the possibility of having a more generic approach, which led to the development of FIESTA, a Framework for Incremental Evolution of SofTware Architectures. To generalize the TranSAt approach, proposed by Olivier Barais during his Ph.D. [49], in order to be independent of any specific ADL, we performed a domain analysis to understand the integration process in architectures described in various ADLs. This analysis allowed us to identify the common architecture elements that are involved in integration, leading to the definition of a generic ADL model. This analysis also enabled the definition of more abstract expressions to specify a join point mask and the transformation rules. Furthermore, we built our generic framework so that adding support for new ADLs is easy as the generic engine can be configured through the specification of a limited number of well-identified ADL-specific functions [43], [23]. FIESTA is still under development in the context of Guillaume Waignier's Ph.D. and is available on INRIA Gforge (see section 5.7).

6.6.2. Detection of Design Defects

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

The Ph.D Thesis of Naouel Moha 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 proposed a language to specify detection rules and generate automatically design defect detection algorithms. The paper appeared in the international ECOOP workshop on Object-Oriented Reengineering (WOOR 2007) presents and discusses the results of applying these algorithms on 11 open-source object-oriented systems [34].

6.6.3. Metamodel for Software Tracability

Participants: Dolores Diaz, Laurence Duchien, Lionel Seinturier.

In [18], we proposed a metamodel for software tracability. The idea is to be able to follow changes during time and across different successive versions of an application or a system. This metamodel helps in understanding how a piece of software evolves to understand the cause of errors and malfunctions. This metamodel has been put into practise on case studies from the NorSys company in the context of the Ph.D. thesis of Dolores Diaz (see section 7.2 and [14]).

6.6.4. A Model-Driven Approach for Soft Real-Time Component Based Applications

Participants: Olivier Barais [IRISA], Laurence Duchien, Sébastien Saudrais [IRISA], Noël Plouzeau [IRISA].

In component-based applications for real-time software, Quality of Service must be introduced into component specifications. However, this information is complex to manipulate when components are composed. The difficulty with time properties comes from the fact that time is not a ressource and can not be stopped. First, our approach proposes a formalism for representing and adding QoS informations of time in a component assembly. The QoS represents the time and will be used to describe components behaviour and compose components together. Secondly, we introduce a set of patterns that defines time properties in the component behavior. This approach is based on a separation of concerns in order to facilitate the composition and also the architect's design [41] [42].

6.7. CALA Results

This section presents results obtained in the context of the INRIA associate team CALA (see section 8.3).

6.7.1. Early Aspects and Model-Driven Software Engineering

Participants: Maja D'Hondt, María Agustina Cibrán [VUB].

We propose an approach that combines Model-Driven Engineering and Aspect-Oriented Software Development in order to automatically translate high-level business rules to aspects and integrate them with existing object-oriented applications. The separation of rule-based knowledge from the core application as explicit business rules has been the focus of many existing approaches. However, they fail at supporting rules that are both high-level, defined in domain terms, and operational, i.e. automatically executable from the core application. We propose high-level languages for expressing business rules at the domain level, as well as their connections to the core application. We provide support for automatically translating high-level rules to object-oriented programs and their connections to aspects, since these crosscut the core application. Separation of concerns is preserved at the domain and implementation levels, facilitating traceability, reusability and adaptability. A prototype implementation is provided. The approach is extensively evaluated in the Web Services Management Layer, which contains implicit rules for configuring, managing and selecting Web Services. This work is presented in the Ph.D. thesis of María Agustina Cibrán at the Vrije Universiteit Brussel (Belgium) in June 2007 and was co-supervised by Maja D'Hondt [50].

6.7.2. Change-Oriented Model-Driven Software Engineering

Participants: Peter Ebraert [VUB], Ellen Van Paesschen.

We propose an approach that combines Model-Driven Engineering and Change-Oriented Software Development and which centralises changes as the main entity in the development process. The subject of the change refers to the building block of a programming language in which the program is being developed. We build on object-oriented programming and take the Famix meta-model as a model for describing the programs that are to be changed. The four problems with respect to the model of first-class changes we aim to tackle are the restricted level of granularity in the different types of changes, the overloading of change types, the lack of high-level changes and the lack of program exploration facilities hinder good software evolution support. Four extensions to the model of first-class changes are presented: fine-grained, composable, dependable and intensional changes. These extensions overcome the problems that are identified in the existing model of first-class changes. ChEOPS, a preliminary Smalltalk implementation is based on the existing implementation of first-class changes, but extended with solutions for the four extra requirements that were identified. Experiments in ChEOPS made a validation of the extensions to the model possible.

Intermediate results are presented in [27] and [28]. This work will be presented in the Ph.D. thesis of Peter Ebraert at the Vrije Universiteit Brussel (Belgium) in 2008 and was co-supervised by Ellen Van Paesschen.

6.7.3. Separation of Concerns for Supporting UI Evolution and Adaptation

Participants: Sofie Goderis [VUB], Ellen Van Paesschen.

Evolving a software system not only affects the source code responsible for the core application, but also the user interface. A problem with maintaining user interface (UI) logic is that it is entangled with the underlying application logic. The fact that the UI logic is scattered throughout the application logic makes adapting the UI logic and evolving the application logic cumbersome for the programmer.

In this approach we aid a programmer to cope with the complexity of UI development. In order to do so the UI and application code should be separated as much as possible. This idea has been applied to many software engineering areas. With respect to user interfaces, it has been applied, amongst others, by the Model-View-Controller metaphor and 3-tiered systems. Both approaches however, when put to practice, still result in entangled and scattered code and don't help the developer to cope with several UI concerns separately. A framework to support our approach, called DEUCE, is currently being implemented. The concerns are described declaratively by means of facts and rules. A reasoning mechanism (i.e. backward and forward chainer) puts the concerns together. One of the case studies on which we are working deals with how UIs can adapt to a changing context, e.g. the UI screen of a mobile phone or a calculator is automatically rotated when the user turns the device upside down.

Intermediate results are presented in [32] and [19]. This work will be presented in the Ph.D. thesis of Sofie Goderis at the Vrije Universiteit Brussel (Belgium) in 2008 and was co-supervised by Ellen Van Paesschen.

6.7.4. Static Analysis

Participants: Johan Brichau, Coen De Roover [VUB].

With the participation of Dr. Johan Brichau and Coen De Roover from the associated CALA team, we implemented a tool that allowed to query a code-base by means of a behavioural template. This template is translated to queries on a dynamic representation of the code-base, obtained by performing static analysis. The results of this work were accepted to the workshop on Partial Evaluation and Program Manipulation (PEPM) [39]. This work was performed as a part of the CALA team.

7. Contracts and Grants with Industry

7.1. France Telecom

Participants: Laurence Duchien, Nicolas Pessemier, Lionel Seinturier.

This contract is a CRE (“Contrat de Recherche Externe”) that takes place in the context of the “accord-cadre” between INRIA and France Telecom R&D. This is a 3-years contract that begun in October 2004. The scientific teams involved in the project are for INRIA, the Jacquard project-team, and for France Telecom R&D, the ASR/Polair department. The contract goal is to study and construct component and aspect based software architectures. The Fractal component model from France Telecom R&D and the JAC AOP framework from Jacquard form the background of this work. The expected result is a model (FAC for Fractal Aspect Component) that merges and unifies aspects and components. The Ph.D. thesis (defended at the University of Lille on June 27, 2007) of Nicolas Pessemier [15] is directly related to this contract [14].

7.2. NorSys

Participants: Dolorès Diaz, Laurence Duchien, Lionel Seinturier.

This contract is associated to a CIFRE Ph.D. thesis between the Jacquard project-team and the NorSys service company. The goal of the contract is to study aspect-orientation in the early stages of software development. AOP emerged as a programming technique but the question is now open in the international research community to tell whether it can also bring to innovations into the early stages of requirement engineering, analysis and design. This contract begun in January 2004. The Ph.D. thesis work of Dolores Diaz is directly related to this contract [14].

8. Other Grants and Activities

8.1. Regional Initiatives

8.1.1. CPER MOSAIQUES

The MOSAIQUES Project (“MOdeles et InfraStructures pour Applications ubIQUitairES” or Models and middleware for ubiquitous applications) defines a design and programming framework for adaptive application definitions that run in an ubiquitous environment. The CPER (Contrat Plan Etat Region) project includes the University of Lille 1 with LIFL Laboratory (STC and SMAC teams) and INRIA projects Jacquard and POPS, TRIGONE laboratory, INRETS, Ecole des Mines de Douai and the University of Valenciennes and of Hainaut-Cambresis. Application domains are transportation and e-learning systems. Laurence Duchien is in charge of this project. The project has started in February 2005 and is scheduled for a 24-months period. The end was in June 2007.

8.2. National Initiatives

8.2.1. ANR ARA REVE

Participants: Laurence Duchien, Anne-Françoise Le Meur, Frédéric Loiret, Ales Plsek, 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 [63]. Lionel Seinturier is the scientific leader of this project. The project has started in January 2006 and is scheduled for a 36-months period.

8.2.2. 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 Neptune's days [38].

8.2.3. ANR TLog Flex-eWare

Participants: Frédéric Loiret, Philippe Merle, Alex Plsek, 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 which 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 which 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.

8.2.4. ANR TLog JOnES

Participants: Nicolas Dolet, Philippe Merle.

JOnES is a 24-months ANR TLog project started in January 2006 and involving INRIA (ObjectWeb, Sardes and ADAM teams), EBM WebSourcing, ENSTIMAC, France Telecom R&D, Open Wide, and ScalAgent Distributed Technologies.

The goal of this project is to design and develop an open source distributed Enterprise Software Bus (ESB) infrastructure based on the Java Business Integration (JBI) specification and implemented with the Fractal component model. The major innovation of JOnES is to propose a new distributed open framework for ESB based on the interconnection of JBI-compliant containers. The project results are distributed through the ObjectWeb PEtALS open source project (see at http://petals.objectweb.org).

The main ADAM's contributions are: Fraclet (see section 5.5) as the Fractal programming model used to build PEtALS components, Fractal Deployment Framework (FDF) (see section 5.6) to deploy distributed PEtALS infrastructures and applications, and GoTM (see section 5.8) to build the PEtALS distributed transaction service.

8.2.5. ANR TLog SCOrWare

Participants: Damien Fournier, Philippe Merle, Lionel Seinturier.

SCOrWare is a 24-months ANR TLog project started in January 2007 and involving Amadeus, Artenum, EBM WebSourcing, Edifixio, INRIA (ADAM, ObjectWeb, and Sardes teams), INT, IRIT, Obeo, and Open Wide. 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.

The main ADAM's contributions are: Tinfi (see section 5.9) as the micro-kernel for supporting the SCA component model, FDF (see section 5.6) to deploy distributed SCOrWare infrastructures and applications, and GoTM (see section 5.8) to build the SCOrWare distributed transaction service.

8.2.6. Trade cluster CAPPUCINO

Participant: All members of the ADAM project-team.

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 in the cluster ("pôle de compétitivité") of trade business ("industrie du commerce") - http://www.picom.fr/ - that 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 partners of this project are NorSys (leader) which is a SME service company, INT/GET (School of Telecom industry), INRIA, AUCHAN and 3 Suisses (2 trade companies). See at http://cappucino.oqube.com.

8.2.7. INRETS-LEOST

Participants: Aurélien Bocquet, Areski Flissi, Jean-Marc Geib, Christophe Gransart, Philippe Merle.

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 [59]. We are together members of the french ANR ARA REVE and CPER MOSAIQUES funded projects. In the latter, C. Gransart, A. Flissi and P. Merle have collaborated to design and implement a component-based software infrastructure for ubiquitous computing [53], [55], especially for contextual transportation applications [52], providing service discovery and automatic deployment to PDA devices [54]. Moreover, J.-M. Geib and C. Gransart are co-supervisors of the Ph.D. thesis of A. Bocquet around a multiple middleware software infrastructure for accessing services from mobile systems.

8.3. European Initiatives

8.3.1. ERCIM Working Group Software Evolution

Participants: Maja D'Hondt, Stéphane Ducasse, Laurence Duchien, Tom Mens, Ellen Van Paesschen.

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. We have organized the International ERCIM Workshop on Software Evolution in April 2006 at Lille, and we will participate to the organization of the symposium in 2007. Maja D'Hondt and Ellen Van Paesschen had an ERCIM post-doc grant. In 2007, ADAM team accommodates Tom Mens, U. Mons, person in charge of this WG.

8.3.2. IAP MoVES

Participants: Laurence Duchien, Ales Plsek, 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.

8.3.3. INRIA associate team CALA

Participants: Maja D'Hondt, Laurence Duchien, Johan Fabry, Anne-Françoise Le Meur, Carlos Francisco Noguera Garcia, Nicolas Pessemier, Lionel Seinturier.

The INRIA project ADAM 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/index.html) 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.

8.3.4. IST-FP6 NoE AOSD-Europe

Participants: Laurence Duchien, Carlos Francisco Noguera Garcia, Nicolas Pessemier, 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.

8.3.5. ITEA S4ALL

Participants: Nicolas Dolet, James Manley, Philippe Merle.

S4ALL ("Services-for-All") is an ITEA project whose partners are Alcatel CIT, Bull, Capricode, Fraunhofer Fokus, HIIT, INRIA, Instituto de Telecomunicaçoes, INT, mCENTRIC, Nokia, PT Inovaçao, Schneider Electric, Thales, Université Joseph Fourier, Universidad Politécnica de Madrid, University of Kassel, Vodafone, and Xquark/Odonata. The main vision is to build "A world of user-centric services that are easy to create, share and use".

Our contribution in this project is to design and build a component-based framework to automatically deploy any kind of middleware and business components like J2EE components, OSGi services, and Web services (see section 5.6).

8.4. International Initiatives

8.4.1. ICT-Asia

Participant: Philippe Merle.

This is a French-Asian cooperation on open adaptive middleware for ubiquitous environments. Partners are National University of Defense Technology (China), Open Source Software Resource Center (Vietnam), Peking University (China), and INRIA (Sardes, ObjectWeb, and ADAM project-teams).

It is widely anticipated that future ubiquitous computing environments will be highly dynamic, subject to constant changes and of ever-increasing complexity. This in turn motivates the construction of dynamically configurable software infrastructures to provide a consistent, systematic basis for system evolution, control and management.

Targeted research is required into autonomic and complex adaptive approaches to distributed system design, along with non-intrusive middleware virtualization techniques. Such research will provide the essential foundations upon which distributed, adaptive, self-managing, self-healing systems, capable of transparently supporting the growing needs of distributed e-business, e-gov, e-learning applications, will be built.

8.4.2. OW2

Participants: Yann Davin, Nicolas Dolet, Guillaume Dufrêne, Philippe Merle, Nicolas Pessemier, 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 ObjectWeb is that of a set of components which can be assembled to offer high quality middleware. We are member of this consortium, Philippe Merle is the leader of the Fractal project, and our software projects AOKell, Fraclet, Fractal Explorer, GoTM, JAC, and OpenCCM are hosted by the consortium.

8.4.3. 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. These algorithms will reuse some modules of our tool FIESTA (see section 5.7). 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 will be in the ADAM project-team during one year and the Ph.D. defense will take place in December 2008. Several exchanges of one or two weeks for the advisors will allow the coordination of the Ph.D. co-supervision.

9. Dissemination

9.1. Visiting researchers

- T. Mens, University of Mons, Belgium, December 2006-October 2007,
- Y.-G. Gueheneuc, University of Montreal, Canada, March 2007.

9.2. Examination Committees

- **Maja D'Hondt** was in the examination committee of the following Ph.D. thesis:
  - María Agustina Cibrán, June 2007, Vrije Universiteit Brussel, Belgium (co-advisor)

- **Laurence Duchien** was in the examination committee of the following Ph.D. thesis:
  - Q. Nguyen-Hai, January 2007, University of Lille 1 (advisor)
  - C. Dinont, March 2007, University of Lille 1 (chair)
  - Y. Vandewoude, March 2007, University of Leuven (KUL), Belgium (member)
  - N. Pessemier, June 2007, University of Lille 1 (co-advisor)
  - P. Sriplakich, September 2007, University of Paris 6 (referee)
  - M. Ben Hmida, November 2007, University of Dauphine (referee)
  - H. Chang, December 2007, University of Nice (referee)
  - D. Diaz, December 2007, University of Lille 1 (co-advisor)
  - S. Saudrais, December 2007, University of Rennes 1 (co-advisor)
  - E. Piel, December 2007, University of Lille 1 (chair)
- **Lionel Seinturier** was in the examination committee of the following Ph.D. thesis:
  - Ali-Erdem Özcan, March 2007, Institut National Polytechnique Grenoble (referee)
  - Mikael Desertot, April 2007, University of Grenoble 1 (referee)
  - Simon Denier, June 2007, Ecole des Mines de Nantes (referee)
  - Pierre-André Caron, June 2007, University of Lille 1 (chair)
  - Julien De Antoni, October 2007, INSA Lyon (referee)
  - Didier Hoareau, December 2007, University Bretagne Sud (referee)
  - Guillaume Bastide, December 2007, University of Nantes (referee)
  - Nicolas Loriant, December 2007, Ecole des Mines de Nantes (referee)

9.3. Journals, Conferences, Workshops

- **Maja D'Hondt** has been
  - member of the following committees:
    * Program committee of the Software Composition symposium (SC2007), satellite event of ETAPS 2007,
    * Program committee of the 15th International Smalltalk Joint Conference 2007 (ESUG 2007), Lugano, Switzerland,
    * Program committee of the 3ème Journée Francophone sur le Développement de Logiciels Par Aspects (JFDLPA07),
    * Program committee of the workshop on Software Engineering Properties of Languages and Aspect Technologies, affiliated with AOSD.07,
    * Program committee of the Doctoral Symposium of the ACM/IEEE 10th International Conference on Model Driven Engineering Languages and Systems 2007,
    * Organizing committee of AOSD.07 as Student Co-Chair with Yvonne Coady (40 participants).
  - reviewer for the following journal:
    * Second special issue on Experimental Software and Toolkits (EST) in the Elsevier Journal of Science of Computer Programming
- **Laurence Duchien** has been
  - member of the following committees:

    * Program committee of special issue IDM, RSTI-L'Objet, Hermès, December 2007,
    * Program committee ECSA 2007, http://kybele.escet.urjc.es/ecsa/
    * Program committee FORTE 2007, http://cs.ttu.ee/FORTE07/
  - reviewer for the following journal:
    * Special Issue on Model Based Development for Secure Information Systems, of the Information and Software Technology Journal.

- **Johan Fabry** has been
  - Demonstrations chair of the AOSD07 conference,
  - Co-organiser of the second workshop on Aspects, Dependencies and Interactions (ADI07) at ECOOP 07. Co-organisers: Lodewijk Berghmans (Universiteit Twente), Johan Fabry (INRIA), Mario Sudholdt (Ecole des Mines de Nantes), Ruzanna Chichyan (Lancaster University), Frans Saenen (Katholieke Universiteit Leuven).

- **Anne-Françoise Le Meur** has been member of the following committees:
  - Organizing committee and program committee of DSAL'07 - workshop on Domain Specific Aspect Languages organized at AOSD'07 (International Conference on Aspect-Oriented Software Development), March 12-16, 2007, Vancouver, British Columbia,
  - Program committee of GPCE'07 - International Conference on Generative Programming and Component Engineering (GPCE), Salzburg, Austria, October 1 - 3, 2007,
  - Program committee of AGPES'07 - Workshop on Automatic Program Generation for Embedded Systems organized at GPCE'07, Salzburg, Austria, October 1 - 3, 2007.

- **Philippe Merle** has been
  - member of the following committees:
    * Editorial board of the journal RSTI-L'Objet, Hermès.
  - reviewer for the following journals:
    * Special issue on Software Components - The Fractal Initiative in Annals of Telecommunications Journal edited by Lavoisier.
    * Special issue on « Architecture de contrôle en robotique : aspects logiciels » in Journal Européen des Systèmes Automatisés (JESA).

- **Lionel Seinturier** has been a member of the following committees:
  - Editorial board of the TSI (Hermes) journal.
  - Program committee, 6th International Symposium on Software Composition (ETAPS-SC 2007), Braga, Portugal, March 2007.
  - Program committee, 12th International Conference on Reliable Software Technologies (Ada-Europe 2007), Geneva, June 2007.
  - Program committee, 13th Conference on Langages et Modèles à Objets (LMO 2007), Toulouse, France, March 2007.
  - Program committee, 3rd Journée Francophone sur le Développement de Logiciels Par Aspects (JFDLPA 2007), Toulouse, France, March 2007.

- **Ellen Van Paesschen** has been
  - member of the following committees:

    * Program committee of the Atelier sur l'évolution et la rétro-ingénierie du logiciel, workshop at the Langages et Modèles à Objets (LMO) conference, in March 2007 in Toulouse (France).
  - co-reviewer for several conferences such as ESUG 2007, JFDLPA07, MoDELS07.
  - reviewer for the following journals:
    * Software and System Modeling (SoSyM) journal (Springer-Verlag), 2007.
    * Transactions on Knowledge and Data Engineering journal (IEEE Computer Society), 2007.

- **Maja D'Hondt**, **Toms Mens**, and **Ellen Van Paesschen**: Organizing committee of International ERCIM workshop on Software Evolution, co-located with the International Conference on Software Maintenance (ICSM), in October 2007 in Paris (France).
- **Johan Fabry** and **Anne Françoise Le Meur**: Co-organiser of the second workshop on Domain-Specific Aspect Languages (DSAL07), held at the AOSD07 conference. Co-organisers: Johan Fabry (INRIA) - Eric Tanter (DCC- Univerisdad de Chile), Thomas Cleenewerk (Vrije Universiteit Brussel), Anne-Françoise Le Meur (INRIA), Jacques Noye (Ecole des Mines de Nantes).

9.4. 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** was member of the UFR board at USTL until March 2007, she is member of the LIFL scientific board, member of CSE 27nd section of Universities of Lille 1, CNAM, and Paris 6. She is member of the ERCIM Software Evolution Group. She is chair of the Scientific Committee of the CR INRIA-Futurs and 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://www-lsr.imag.fr/GPL/). She has also evaluated projects for the ANR Software technologies and University of Montreal. She has reviewed Ph.D. applications for the Cor Baayen Award 2007 for ERCIM organization.
- **Areski Flissi** is member of the "Comité des Utilisateurs des Moyens Informatiques" (CUMI) of CR Futurs and representing ADAM project-team members. He is member of the System/Network technical team at LIFL.
- **Anne-Françoise Le Meur** is member of CSE 27nd section of University of Lille 1. She has served as a jury during the recruitment of INRIA R&D engineers in May-June 2007 at INRIA Futurs.
- **Philippe Merle** is in charge of the European Affairs within the European Partnerships Department (DPE) for INRIA CR Futurs, is member of the Incitative Action Working Group (GTAI) of the Scientific and Technological Orientation Council (COST) of INRIA, and is member of the associate engineer committee for INRIA CR Futurs. He is member of the steering committee of the "Grilles, Système et Parallélisme" (GSP) working group of the CNRS ARS GdR.
- **Lionel Seinturier** is, in the direction board of the LIFL laboratory, in charge of supervising the financial activities of the 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 board of the IEEA faculty (UFR) at the University of Lille 1.

9.5. 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) 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 in Master Research Sciences et Technologies Mention Informatique at USTL, UFR IEEA.
- **Lionel Seinturier** teaches middleware (Master 1 and 2), component-based software engineering (Master 2), aspect-oriented programming (Master 2), and object-oriented design (Master 1) at the University of Lille 1. Lionel Seinturier is in charge of supervising the student projects and the student internships (stages) for the TIIR (Telecom and Distributed System) speciality of the master in computer science at USTL, UFR IEEA.

9.6. Miscellaneous

- SciencesOPark and 40 ans de l'INRIA : We have actively participated in the definition of one of the demos to be given by the team at the SciencesOPark and 40 ans de l'INRIA events. This included participation in the definition of the scenario of the demo, meetings with the coordinators and industrial partners, and the development of the demo application itself.
- The team has organized its internal seminar, Cap-Hornu, France, August 29-31th, 2007.
- The team has organized the two CALA seminars, Vrije Universiteit Brussel, Belgium, Ambient Days, February 13 and 14, 2007, and Junior Days, February 19 and 20, 2007.

10. Bibliography

Major publications by the team in recent years

[1] O. BARAIS, L. DUCHIEN. *SafArchie Studio: An ArgoUML Extension to Build Safe Architectures*, ISBN: 0-387-24589-8, Springer, 2005, p. 85–100.

[2] O. BARAIS, J. LAWALL, A.-F. LE MEUR, L. DUCHIEN. *Safe Integration of New Concerns in a Software Architecture*, in "Proceedings of the 13th International Conference on Engineering of Computer Based Systems (ECBS'06), Potsdam, Germany", IEEE, March 2006, p. 52–64.

[3] C. CONSEL, J. LAWALL, A.-F. LE MEUR. *A Tour of Tempo: A Program Specializer for the C Language*, in "Science of Computer Programming", vol. 52, 2004, p. 341-370.

[4] 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.

[5] 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.

[6] P. MERLE, C. GRANSART, J.-M. GEIB. *CorbaWeb: A Generic Object Navigator*, in "Computer Network ISDN Systems", vol. 28, n^{o} 7-11, 1996, p. 1269–1281.

[7] R. PAWLAK, L. DUCHIEN, G. FLORIN, L. SEINTURIER. *Dynamic Wrappers: Handling the Composition Issue with JAC*, in "TOOLS '01: Proceedings of the 39th International Conference and Exhibition on Technology of Object-Oriented Languages and Systems (TOOLS39), Washington, DC, USA", IEEE Computer Society, 2001, p. 56–65.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

Year Publications

Books and Monographs

[13] T. MENS, M. D'HONDT (editors). *Proceedings of the International ERCIM Workshop on Software Evolution (2006)*, Electronic Notes in Theoretical Computer Science, vol. 166, Science Direct, Elsevier, January 2007.

Doctoral dissertations and Habilitation theses

[14] D. DIAZ. *Réaliser des évolutions logicielles sur des applications d'entreprise en phase de maintenance*, Ph. D. Thesis, Université Lille 1, Laboratoire d'Informatique Fondamentale de Lille, Lille, France, December 2007.

[15] N. PESSEMIER. *Unification des approches par aspects et à composants*, Ph. D. Thesis, Université Lille 1, Laboratoire d'Informatique Fondamentale de Lille, Lille, France, June 2007.

Articles in refereed journals and book chapters

[16] J. BRICHAU, A. KELLENS, K. GYBELS, K. MENS, R. HIRSCHFELD, T. D'HONDT. *Application-Specific Models and Pointcuts Using a Logic Metalanguage*, in "Computer Languages, Systems & Structures", May 2007.

[17] T. COUPAYE, V. QUÉMA, L. SEINTURIER, J.-B. STEFANI. *Le système de composants Fractal*, in "Intergiciel et Construction d'Applications Réparties", http://sardes.inrialpes.fr/ecole/livre/pub/, January 2007, http://hal.inria.fr/inria-00155090.

[18] D. DIAZ, L. SEINTURIER, L. DUCHIEN, P. FLAMENT. *Une aide à la réalisation d'évolutions logicielles avec la notion de traçabilité fonctionnelle*, in "L'Objet", vol. 13, 2007, p. 117–145.

[19] S. GODERIS, D. DERIDDER, E. V. PAESSCHEN. *DEUCE : A Declarative Framework for Extricating User Interface Concerns*, in "Journal of Object Technology (JOT), special issue for the Technology on Object-Oriented Languages and Systems (TOOLS) conference", vol. 6, June 2007, p. 142–156.

[20] G. HERMOSILLO, R. GOMEZ, L. SEINTURIER, L. DUCHIEN. *Using Aspect Programming to Secure Web Applications*, in "Journal of Software", To appear, December 2007.

[21] C. NOGUERA, R. PAWLAK. *AVal: an Extensible Attribute-Oriented Programming Validator for Java*, in "Journal of Software Maintenance and Evolution: Research and Practice", vol. 19, n° 4, July 2007, p. 253-275.

[22] R. ROUVOY, P. MERLE. *Using Microcomponents and Design Patterns to Build Evolutionary Transaction Services*, in "Electronic Notes in Theoretical Computer Science (ENCTS)", vol. 166, January 2007, p. 111–125.

[23] G. WAIGNIER, A.-F. LE MEUR, L. DUCHIEN. *FIESTA : A Generic Framework for Integrating New Functionalities into Software Architectures*, in "International Journal of Cooperative Information Systems (IJCIS)", To appear, 2007.

Publications in Conferences and Workshops

[24] M. BRUNTINK, A. VAN DEURSEN, M. D'HONDT, T. TOURWÉ. *Simple Crosscutting Concerns are not so Simple — Analysing Variability in Large-Scale Idioms-Based Implementations*, in "Proceedings of the Sixth International Conference on Aspect-Oriented Software Development (AOSD'07)", vol. 208, ACM Press, 2007, p. 199–211.

[25] D. CONAN, R. ROUVOY, L. SEINTURIER. *Scalable Processing of Context Information with COSMOS*, in "Proceedings of the 7th IFIP International Conference on Distributed Applications and Interoperable Systems (DAIS'07), Paphos, Cyprus", J. INDULSKA, K. RAYMOND (editors), Lecture Notes in Computer Science, vol. 4531, Springer, June 2007, p. 210-224.

[26] J. DUBUS, P. MERLE. *Towards Model-Driven Validation of Autonomic Software Systems in Open Distributed Environments* , in "Proceedings of the ECOOP Workshop on Model-Driven Adaptation, Berlin, Germany", July 2007, p. 39–48.

[27] P. EBRAERT, E. V. PAESSCHEN, T. D'HONDT. *Change-Oriented Round-Trip Engineering: First-class changes for Agile Software Development*, in "Proceedings of the RIMEL workshop at the Langages et Modèles à Objets conference, Toulouse, France", March 2007, p. 2–8.

[28] P. EBRAERT, J. VALLEJOS, P. COSTANZA, E. V. PAESSCHEN, T. D'HONDT. *Change-Oriented Software Engineering*, in "Proceedings of the 15th International Smalltalk Joint Conference (ESUG07), Lugano, Switzerland", Lecture Notes in Computer Science, Springer-Verlag, August 25-31 2007.

[29] J. FABRY, C. NOGUERA. *Abstracting Connection Volatility through Tagged Futures*, in "Proceedings of the 2nd Ambient Intelligence Developments (AmI.d) Conference", September 2007.

[30] J. FABRY, C. NOGUERA. *AmI: The Future is Now – A position paper*, in "Proceedings of the 3rd Workshop on Object Technology for Ambient Intelligence and Pervasive Systems (OT4AmI) at ECOOP'07, Berlin, Germany", July 2007, p. 13–18.

[31] J. FABRY, É. TANTER, T. D'HONDT. *ReLAx: Implementing KALA over the Reflex AOP Kernel*, in "Proceedings of the second workshop on Domain-Specific Aspect Languages", ACM, 2007.

[32] S. GODERIS, D. DERIDDER, E. V. PAESSCHEN, T. D'HONDT. *DEUCE: Separating Concerns in User Interfaces*, in "Proceedings of the 2nd International Conference on Software Engineering Advances (ICSEA07), Cap Esterel, French Riviera, France", June 5–8 2007.

[33] G. HERMOSILLO, R. GOMEZ, L. SEINTURIER, L. DUCHIEN. *An Aspect for Programming Secure Web Applications*, in "Proceedings of the 2nd International Conference on Availability, Reliability and Security (ARES'07)", vol. 0, IEEE Computer Society, April 2007, p. 1026–1033.

[34] N. MOHA, Y.-G. GUÉHÉNEUC, L. DUCHIEN, A.-F. LE MEUR. *Discussion on the Results of the Detection of Design Defects*, in "Proceedings of the 10th ECOOP Workshop on Object-Oriented Reengineering (WOOR'07), Berlin, Germany", S. DEMEYER, Y.-G. GUÉHÉNEUC, C. LANGE, K. MENS, R. WUYTS, S. DUCASSE (editors), Springer-Verlag, July–August 2007.

[35] C. NOGUERA, J. FABRY, E. V. PAESSCHEN, C. PARRA. *Generation of Application Composition Support in Ubiquitous Computing Based on Context Models*, in "SAC'08, March 16-20, 2008, Fortaleza, Ceará, Brazil", To Appear, 2008.

[36] C. A. PARRA, M. D'HONDT, C. NOGUERA, E. V. PAESSCHEN. *Introducing Context-Awareness in Applications by Transforming High-Level Rules*, in "Proceedings of the 3rd Workshop on Object Technology for Ambient Intelligence and Pervasive Systems (OT4AmI) at ECOOP'07, Berlin, Germany", July 2007, p. 19–26.

[37] A. PLSEK, L. SEINTURIER, P. MERLE. *Ambient-Oriented Programming in Fractal*, in "Proceedings of the 3rd Workshop on Object Technology for Ambient Intelligence and Pervasive Systems (OT4AmI) at ECOOP'07, Berlin, Germany", July 2007, p. 33–38.

[38] N. RIVIERRE, A.-F. LE MEUR, P. LAHIRE, N. PLOUZEAU, B. TRAVERSON. *Projet FAROS : Un environnement de composition pour la construction fiable d'architectures orientées services.*, in "Journées Neptune, Paris, France", May 2007.

[39] C. D. ROOVER, J. BRICHAU, C. NOGUERA, T. D'HONDT, L. DUCHIEN. *Behavioural Similarity Matching using Concrete Source Code Templates in Logic Queries*, in "Proceedings of the ACM Sigplan Workshop on Partial Evaluation and Program Manipulation (PEPM'07), Nice, France", ACM Press, January 2007, p. 92–101.

[40] R. ROUVOY, P. MERLE. *Description et de vérification de motifs d'architecture avec Fractal ADL*, in "Proceedings of the French Conference on Langages et Modèles à Objets (LMO'07), Toulouse, France", Hermès Science publications, March 2007, p. 49–64.

[41] S. SAUDRAIS, O. BARAIS, L. DUCHIEN, N. PLOUZEAU. *Intégration de propriétés temporelles dans des applications à base de composants*, in "Conférence Francophone sur les Approches Formelles dans l'Assistance au Développement de Logiciels (AFADL'07), Namur, Belgique", June 2007.

[42] S. SAUDRAIS, O. BARAIS, N. PLOUZEAU. *Integration of Time Issues into Component-based Applications*, in "Proceeding of the 10th International ACM SIGSOFT Symposium on Component-Based Software Engineering (CBSE 2007), Medford, MA, USA", Lecture Notes in Computer Science, vol. 4608, Springer, July 2007, p. 173-188.

[43] G. WAIGNIER, A.-F. LE MEUR, L. DUCHIEN. *FIESTA : A Generic Framework for Integrating New Functionalities into Software Architectures*, in "Proceedings of 1st European Conference on Software Architecture (ECSA'07), Aranjuez (Madrid), Spain", Lecture Notes in Computer Science, vol. 4758, Springer-Verlag, September 2007, p. 76–91.

Internal Reports

[44] P. MERLE, ET AL. *SCA Platform Specifications - Version 1.0*, Deliverable, ANR TLog SCOrWare Project, September 2007.

Miscellaneous

[45] L. SCHMIDT. *Micro-noyau de composants logiciels*, Master thesis, Laboratoire d'Informatique Fondamentale de Lille, June 2007.

References in notes

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

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

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

[49] O. BARAIS. *Construire et Maîtriser l'évolution d'une architecture logicielle à base de composants*, Ph. D. Thesis, Laboratoire d'Informatique Fondamentale de Lille, Lille, France, November 2005.

[50] M. CIBRÁN. *Connecting High-Level Business Rules with Object-Oriented Applications: An Approach using Aspect-Oriented Programming and Model-DrivenEngineering*, Ph. D. Thesis, Vrije Universiteit Brussel (VUB), Brussels, June 2007.

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

[52] A. FLISSI, C. GRANSART, P. MERLE. *A Component-Based Software Infrastructure for Contextual Transportation Applications*, in "Proceedings of the 5th International Conference on Intelligent Transportation System - Telecommunication (ITS-T 2005), Brest, France", June 2005.

[53] A. FLISSI, C. GRANSART, P. MERLE. *A Component-based Software Infrastructure for Ubiquitous Computing*, in "4th International Symposium on Parallel and Distributed Computing (ISPDC 2005), Lille, France", July 2005.

[54] A. FLISSI, C. GRANSART, P. MERLE. *A Service Discovery and Automatic Deployment Component-based Software Infrastructure for Ubiquitous Computing*, in "Ubiquitous Mobile Information and Collaboration Systems, CAiSE Workshop (UMICS 2005), Porto, Portugal", June 2005.

[55] A. FLISSI, C. GRANSART, P. MERLE. *Une infrastructure composants pour des applications ubiquitaires*, in "2nde conférence Ubiquité et Mobilité (UbiMob 2005), Grenoble, France", June 2005.

[56] A. FLISSI, P. MERLE. *A Generic Deployment Framework for Grid Computing and Distributed Applications*, in "Proceedings of the 2nd International OTM Symposium on Grid computing, high-performAnce and Distributed Applications (GADA'06), Montpellier, France", Lecture Notes in Computer Science, vol. 4279, Springer-Verlag, November 2006, p. 1402–1411.

[57] 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.

[58] 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.

[59] J.-M. GEIB, C. GRANSART, P. MERLE. *CORBA : des concepts à la pratique*, Masson, 1997.

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

[61] 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.

[62] 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.

[63] 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.

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

[65] 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.

[66] 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.

[67] C. NOGUERA, 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", To appear, IEEE Computer Society, September 2006.

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

[69] 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.

[70] R. ROUVOY, N. PESSEMIER, R. PAWLAK, P. MERLE. *Using Attribute-Oriented Programming to Leverage Fractal-Based Developments*, in "Proceedings of the 5th International ECOOP Workshop on Fractal Component Model (Fractal'06), Nantes, France", July 2006.

[71] R. ROUVOY. *Une démarche à granularité extrêmement fine pour la construction de canevas intergiciels hautement adaptables : application aux services de transactions*, Ph. D. Thesis, Laboratoire d'Informatique Fondamentale de Lille, Lille, France, December 2006, http://www.lifl.fr/~rouvoy/rouvoy-phd-06.pdf.

[72] R. ROUVOY, P. SERRANO-ALVARADO, P. MERLE. *Towards Context-Aware Transaction Services*, in "Proceedings of the 6th International Conference on Distributed Applications and Interoperable Systems (DAIS'06), Bologna, Italy", Lecture Notes in Computer Science, vol. 4025, Springer-Verlag, June 2006, p. 272–288.

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

[74] 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.

[75] 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.

[76] D. THOMAS. *Reflective Software Engineering - From MOPS to AOSD*, in "Journal of Object Technology", vol. 1, n° 4, 2002, p. 17-26.

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