INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

# *Project-Team Triskell*

# *Model Driven Engineering for Component Based Software*

*Rennes - Bretagne-Atlantique*

Theme : Distributed Systems and Services

Activity Report

2009

# Table of contents

# 1. Team

**Research Scientist**
Benoit Baudry [ Research scientist Inria ]

**Faculty Member**
Jean-Marc Jézéquel [ Team Leader, Professor Université de Rennes 1, HdR ]
Olivier Barais [ Assistant Professor Université de Rennes 1 ]
Noël Plouzeau [ Assistant Professor Université de Rennes 1 ]
Benoit Combemale [ Assistant Professor Université de Rennes 1 ]
Naouel Moha [ Assistant Professor Université de Rennes 1 ]

**External Collaborator**
Régis Fleurquin [ On leave from Université de Bretagne Sud ]

**Technical Staff**
Didier Vojtisek [ Research engineer Inria ]
Cyril Faucher [ Inria (project Openembedd) until April 2009 ]
Vincent Mahé [ Inria (project Openembedd) until May 2009 ]
Jacques Falcou [ Inria (project Speeds) since October 2009 ]
Mahmoud Ben Hassine [ Inria (project Galaxy) ]
Hajanirina Johary Rambelontsalama [ Inria IJD ]

**PhD Student**
Erwan Brottier [ CIFRE grant ]
Romain Delamare [ Brittanny Council grant ]
Brice Morin [ INRIA grant ]
François Fouquet [ MESR grant ]
Freddy Munoz [ INRIA grant ]
Jacques Simonin [ Orange Labs ]
Sagar Sen [ INRIA grant ]
Tejeddine Mouelhi [ ENST grant ]
Maha Driss [ INRIA grant, Co-tutelle with U. Tunis ]
Paul Istoan [ Co-tutelle with U. Luxembourg ]
Grégory Nain [ INRIA grant ]
Muhammad-Ali Memon [ SFERE grant ]
Mickael Clavreul [ INRIA grant ]
Juan Cadavid [ INRIA grant ]
Erwan Daubert [ INRIA grant ]

**Post-Doctoral Fellow**
Gilles Perrouin [ INRIA (project MOPCOM and SPEEDS) until June 2009 ]
Kathia Marcal de Oliveira [ INRIA (project Faros) until May 2009 ]

**Visiting Scientist**
Francisca Rosique Contreras [ Universidad de Cartegena ]
Robert France [ Colorado State University ]

**Administrative Assistant**
Loic Lesage [ TR Inria ]

# 2. Overall Objectives

## 2.1. Introduction

### *2.1.1. Research fields*

In its broad acceptation, Software Engineering consists in proposing practical solutions, founded on scientific knowledge, in order to produce and maintain software with constraints on costs, quality and deadlines. In this field, it is admitted that the complexity of a software increases exponentially with its size. However on the one hand, the size itself of the software is on average multiplied by ten every ten years, and on the other hand, economic pressures pushed towards reducing the duration of developments, and in increasing the rates of modifications made to the software.

To face these problems, today's mainstream approaches build on the concept of component based software. The assembly of these components makes it possible to build families of products (a.k.a. *product lines*) made of many common parts, while remaining opened to new evolutions. As component based systems grow more complex and mission-critical, there is an increased need to be able to represent and reason on such assemblies of components. This is usually done by building models representing various aspects of a product line, such as functional variations, structural aspects (object paradigm), or dynamic aspects (languages of scenarios), without neglecting of course non-functional aspects like quality of service (performance, reliability, etc.) described in the form of contracts. Model Driven Engineering (MDE) is then a sub-domain of software engineering focusing on reinforcing design, validation and test methodologies based on the automatic processing of multi-dimensional models.

### *2.1.2. Project-team Presentation Overview*

The research domain of the Triskell project is the model driven development of software product lines. Triskell is particularly interested in component based reactive and large scale distributed systems with quality of service constraints, including reliability, performance, timeliness etc.

Triskell's main objective is to develop model-based methods and tools to help the software designer to efficiently obtain a certain degree of confidence in the reliability of component assemblies that may include third-party components. This involves, in particular, investigating modeling languages allowing specification of both functional and non-functional aspects for software engineering activities ranging from requirements to detailled design. It also involves building a continuum of tools which make use of these models, from model validation and verification, automatic application of design patterns, to test environments and on-line monitors supervising the behavior of the components in Dynamically Adaptable Systems. Since these modeling languages and associated tools appear quite open-ended and very domain specific, there is a growing need for *"tools for building tools for building software"*. Triskell is hence developping KerMeta as an original meta modeling approach allowing the user to fully define his modeling languages (including dynamic semantics) and associated environments (including interpreters, compilers, importers/exporters, etc.) within Eclipse.

To avoid the pitfall of developping *"tools for building tools for the sake of it"*, the Triskell project also has the goal of explicitly connecting its research results to industrial problems through collaborations with industry and technology transfer actions. This implies, in particular, taking into account the industrial standards of the field, namely the Eclipse Modeling Framework (EMF), the OMG's Meta-Object Facility (MOF) and Unified Modeling Language (UML), as well as domain specific component models such as OSGi.

Triskell is at the frontier of two fields of software: the field of specification and formal proof, and that of design which, though informal, is organized around best practices (e.g.; separation of concerns with aspects, models, design patterns, or the use of off-the-shelf components). We believe that the use of our techniques will make it possible to improve the transition between these two worlds, and will contribute to the fluidity of the processes of design, implementation and testing of software.

## 2.2. Highlights of the year

- Triskell developed a first proof of concept of the *models at runtime* idea [33], [21], [35] to tame the complexity of dynamically adaptive systems by combining model-driven and aspect-oriented techniques and offering a high degree of automation and validation. Triskell has applied it in particular to Software Oriented Architectures in the context of Building Automation [30], [33], [47] for AAL (Ambient Assisted Living). A demo of this application has been presented at several venues, including the "Fete de la Science" in Nov. 2009.
- Model transformations constitute a class of programs with unique characteristics that make testing them challenging. We have identified these characteristics [14] and identified promising solutions to overcome these barriers. This work is part of the research developed in the MoCAA Equipe associée in collaboration with Sudipto Ghosh, Robert France from CSU, Franck Fleurey from SINTEF and Jean-Marie Mottu.
- Triskell had 6 papers accepted [29], [32], [36], [31], [38], [42] at MODELS'2009, the main conference in our field (248 papers submitted, 45 accepted), including one ACM Best Paper Award [38]. That consolidates our position as the world leading team in the MDE field.

# 3. Scientific Foundations

## 3.1. Overview

The Triskell project studies new techniques for the reliable construction of software product lines, especially for distributed and reactive software. The key problems are components modeling and the development of formal manipulation tools to refine the design, code generation and test activities. The validation techniques used are based on complex simulations of models building on the standards in the considered domain.

## 3.2. Model Driven Engineering for Distributed Software

### *3.2.1. Software Product Lines*

It is seldom the case nowadays that we can any longer deliver software systems with the assumption that one-size-fits-all. We have to handle many variants accounting not only for differences in product functionalities (range of products to be marketed at different prices), but also for differences in hardware (e.g.; graphic cards, display capacities, input devices), operating systems, localization, user preferences for GUI ("skins"). Obvioulsy, we do not want to develop from scratch and independantly all of the variants the marketing department wants. Furthermore, all of these variant may have many successive versions, leading to a two-dimensional vision of product-lines.

### *3.2.2. Object-Oriented Software Engineering*

The object-oriented approach is now widespread for the analysis, the design, and the implementation of software systems. Rooted in the idea of modeling (through its origin in Simula), object-oriented analysis, design and implementation takes into account the incremental, iterative and evolutive nature of software development [56], [54]: large software system are seldom developed from scratch, and maintenance activities represent a large share of the overall development effort.

In the object-oriented standard approach, objects are instances of classes. A class encapsulates a single abstraction in a modular way. A class is both *closed*, in the sense that it can be readily instanciated and used by clients objects, and *open*, that is subject to extensions through inheritance [58].

### 3.2.3. Design Pattern

Since by definition objects are simple to design and understand, complexity in an object-oriented system is well known to be in the *collaboration* between objects, and large systems cannot be understood at the level of classes and objects. Still these complex collaborations are made of recurring patterns, called design patterns. The idea of systematically identifying and documenting design patterns as autonomous entities was born in the late 80's. It was brought into the mainstream by such people as Beck, Ward, Coplien, Booch, Kerth, Johnson, etc. (known as the Hillside Group). However the main event in this emerging field was the publication, in 1995, of the book *Design Patterns: Elements of Reusable Object Oriented Software* by the so-called Gang of Four (GoF), that is E. Gamma, R. Helm, R. Johnson and J. Vlissides [55]. Today, design patterns are widely accepted as useful tools for guiding and documenting the design of object-oriented software systems. Design patterns play many roles in the development process. They provide a common vocabulary for design, they reduce system complexity by naming and defining abstractions, they constitute a base of experience for building reusable software, and they act as building blocks from which more complex designs can be built. Design patterns can be considered reusable micro-architectures that contribute to an overall system architecture. Ideally, they capture the intent behind a design by identifying the component objects, their collaborations, and the distribution of responsibilities. One of the challenges addressed in the Triskell project is to develop concepts and tools to allow their formal description and their automatic application.

### 3.2.4. Component

The object concept also provides the bases needed to develop *software components*, for which Szyperski's definition [62] is now generally accepted, at least in the industry:

> *A software component is a unit of composition with contractually specified interfaces and explicit context dependencies only. A software component can be deployed independently and is subject to composition by third party.*

Component based software relies on assemblies of components. Such assemblies rely in turn on fundamental mechanisms such as precise definitions of the mutual responsability of partner components, interaction means between components and their non-component environment and runtime support (e.g. .Net, EJB, Corba Component Model CCM, OSGI or Fractal).

Components help reducing costs by allowing reuse of application frameworks and components instead of redeveloping applications from scratch (product line approach). But more important, components offer the possibility to radically change the behaviors and services offered by an application by substitution or addition of new components, even a long time after deployment. This has a major impact of software lifecycle, which should now handle activities such as the design of component frameworks, the design of reusable components as deployment units, the validation of component compositions coming from various origins and the component life-cycle management.

Empirical methods without real component composition models have appeared during the emergence of a real component industry (at least in the Windows world). These methods are now clearly the cause of untractable validation and of integration problems that can not be transposed to more critical systems (see for example the accidental destruction of Ariane 501 [57]).

Providing solutions for formal component composition models and for verifiable quality (notion of *trusted components*) are especially relevant challenges. Also the methodological impact of component-based development (for example within the maturity model defined by the SEI) is also worth attention.

### 3.2.5. Contracts

Central to this trusted component notion is the idea of *contract*. A software contract captures mutual requirements and benefits among stake-holder components, for example between the client of a service and its suppliers (including subcomponents). Contracts strengthen and deepen interface specifications. Along the lines of abstract data type theory, a common way of specifying software contracts is to use boolean assertions called pre- and post-conditions for each service offered, as well as class invariants for defining general consistency properties. Then the contract reads as follows: The client should only ask a supplier for a service in a state where the class invariant and the precondition of the service are respected. In return, the supplier promises that the work specified in the post-condition will be done, and the class invariant is still respected. In this way rights and obligations of both client and supplier are clearly delineated, along with their responsibilities. This idea was first implemented in the Eiffel language [59] under the name *Design by Contract*, and is now available with a range of expressive power into several other programming languages (such as Java) and even in the Unified Modeling Language (UML) with the Object Constraint Language (OCL) [63]. However, the classical predicate based contracts are not enough to describe the requirements of modern applications. Those applications are distributed, interactive and they rely on resources with random quality of service. We have shown that classical contracts can be extended to take care of synchronization and extrafunctional properties of services (such as throughput, delays, etc) [53].

### *3.2.6. Models and Aspects*

As in other sciences, we are increasingly resorting to modelling to master the complexity of modern software development. According to Jeff Rothenberg,

> *Modeling, in the broadest sense, is the cost-effective use of something in place of something else for some cognitive purpose. It allows us to use something that is simpler, safer or cheaper than reality instead of reality for some purpose. A model represents reality for the given purpose; the model is an abstraction of reality in the sense that it cannot represent all aspects of reality. This allows us to deal with the world in a simplified manner, avoiding the complexity, danger and irreversibility of reality.*

So modeling is not just about expressing a solution at a higher abstraction level than code. This has been useful in the past (assembly languages abstracting away from machine code, 3GL abstracting over assembly languages, etc.) and it is still useful today to get a holistic view on a large C++ program. But modeling goes well beyond that.

Modeling is indeed one of the touchstone of any scientific activity (along with validating models with respect to experiments carried out in the real world). Note by the way that the specificity of engineering is that engineers build models of artefacts that usually do not exist yet (with the ultimate goal of building them).

In engineering, one wants to break down a complex system into as many models as needed in order to address all the relevant concerns in such a way that they become understandable enough. These models may be expressed with a general purpose modeling language such as the Unified Modeling Language (UML), or with Domain Specific Languages when it is more appropriate.

Each of these models can be seen as the abstraction of an aspect of reality for handling a given concern. The provision of effective means for handling such concerns makes it possible to establish critical trade-offs early on in the software life cycle, and to effectively manage variation points in the case of product-lines.

Note that in the Aspect Oriented Programming community, the notion of aspect is defined in a sligthly more restricted way as the modularization of a cross-cutting concern. If we indeed have an already existing "main" decomposition paradigm (such as object orientation), there are many classes of concerns for which clear allocation into modules is not possible (hence the name "cross-cutting"). Examples include both allocating responsibility for providing certain kinds of functionality (such as loggin) in a cohesive, loosely coupled fashion, as well as handling many non-functional requirements that are inherently cross-cutting e.g.; security, mobility, availability, distribution, resource management and real-time constraints.

However now that aspects become also popular outside of the mere programming world [61], there is a growing acceptance for a wider definition where an aspect is a concern that can be modularized. The motivation of these efforts is the systematic identification, modularization, representation, and composition of these concerns, with the ultimate goal of improving our ability to reason about the problem domain and the corresponding solution, reducing the size of software model and application code, development costs and maintenance time.

### 3.2.7. Design and Aspect Weaving

So really modeling is the activity of separating concerns in the problem domain, an activity also called *analysis*. If solutions to these concerns can be described as aspects, the design process can then be characterized as a weaving of these aspects into a detailed design model (also called the solution space). This is not new: this is actually what designers have been effectively doing forever. Most often however, the various aspects are not *explicit*, or when there are, it is in the form of informal descriptions. So the task of the designer is to do the weaving in her head more or less at once, and then produce the resulting detailled design as a big tangled program (even if one decomposition paradigm, such as functional or object-oriented, is used). While it works pretty well for small problems, it can become a major headache for bigger ones.

Note that the real challenge here is not on how to design the system to take a particular aspect into account: there is a huge design know-how in industry for that, often captured in the form of Design Patterns (see above). Taking into account more than one aspect as the same time is a little bit more tricky, but many large scale successful projects in industry are there to show us that engineers do ultimately manage to sort it out.

The real challenge in a product-line context is that the engineer wants to be able to change her mind on which version of which variant of any particular aspect she wants in the system. And she wants to do it cheaply, quickly and safely. For that, redoing by hand the tedious weaving of every aspect is not an option.

### 3.2.8. Model Driven Engineering

Usually in science, a model has a different nature that the thing it models ("do not take the map for the reality" as Sun Tse put it many centuries ago). Only in software and in linguistics a model has the same nature as the thing it models. In software at least, this opens the possibility to automatically derive software from its model. This property is well known from any compiler writer (and others), but it was recently be made quite popular with an OMG initiative called the Model Driven Architecture (MDA). This requires that models are no longer informal, and that the weaving process is itself described as a program (which is as a matter of facts an executable meta-model) manipulating these models to produce a detailled design that can ultimately be transformed to code or at least test suites.

The OMG has built a meta-data management framework to support the MDA. It is mainly based on a unique M3 “meta-meta-model” called the Meta-Object Facility (MOF) and a library of M2 meta-models, such as the UML (or SPEM for software process engineering), in which the user can base his M1 model.

The MDA core idea is that it should be possible to capitalize on platform-independent models (PIM), and more or less automatically derive platform-specific models (PSM) –and ultimately code– from PIM through model transformations. But in some business areas involving fault-tolerant, distributed real-time computations, there is a growing concern that the added value of a company not only lies in its know-how of the business domain (the PIM) but also in the design know-how needed to make these systems work in the field (the transformation to go from PIM to PSM). Reasons making it complex to go from a simple and stable business model to a complex implementation include:

- Various modeling languages used beyond UML,
- As many points of views as stakeholders,
- Deliver software for (many) variants of a platform,
- Heterogeneity is the rule,
- Reuse technical solutions across large product lines (e.g. fault tolerance, security, etc.),
- Customize generic transformations,
- Compose reusable transformations,
- Evolve and maintain transformations for 15+ years.

This wider context is now known as Model Driven Engineering.

# 4. Application Domains

## 4.1. From Embedded Systems to Service Oriented Architectures

From small embedded systems such as home automation products or automotive systems to medium sized systems such as medical equipment, office equipment, household appliances, smart phones; up to large Service Oriented Architectures (SOA), building a new application from scratch is no longer possible. Such applications reside in (group of) machines that are expected to run continuously for years without unrecoverable errors. Special care has then to be taken to design and validate embedded software, making the appropriate trade-off between various extra-functional properties such as reliability, timeliness, safety and security but also development and production cost, including resource usage of processor, memory, bandwidth, power, etc.

Leveraging ongoing advances in hardware, embedded software is playing an evermore crucial role in our society, bound to increase even more when embedded systems get interconnected to deliver ubiquitous SOA. For this reason, embedded software has been growing in size and complexity at an exponential rate for the past 20 years, pleading for a component based approach to embedded software development. There is a real need for flexible solutions allowing to deal at the same time with a wide range of needs (product lines modeling and methodologies for managing them), while preserving quality and reducing the time to market (such as derivation and validation tools).

We believe that building flexible, reliable and efficient embedded software will be achieved by reducing the gap between executable programs, their models, and the platform on which they execute, and by developing new composition mechanisms as well as transformation techniques with a sound formal basis for mapping between the different levels.

Reliability is an essential requirement in a context where a huge number of softwares (and sometimes several versions of the same program) may coexist in a large system. On one hand, software should be able to evolve very fast, as new features or services are frequently added to existing ones, but on the other hand, the occurrence of a fault in a system can be very costly, and time consuming. While we think that formal methods may help solving this kind of problems, we develop approaches where they are kept “behind the scene” in a global process taking into account constraints and objectives coming from user requirements.

Software testing is another aspect of reliable development. Testing activities mostly consist in trying to exhibit cases where a system implementation does not conform to its specifications. Whatever the efforts spent for development, this phase is of real importance to raise the confidence level in the fact that a system behaves properly in a complex environment. We also put a particular emphasis on on-line approaches, in which test and observation are dynamically computed during execution.

# 5. Software

## 5.1. Kermeta : Kernel Metamodeling

**Participants:** Olivier Barais, Jacques Falcou, Cyril Faucher, François Fouquet, Jean-Marc Jézéquel, Hajanirina Johary Rambelontsalama, Didier Vojtisek [correspondant].

Nowadays, object-oriented meta-languages such as MOF (Meta-Object Facility) are increasingly used to specify domain-specific languages in the model-driven engineering community. However, these meta-languages focus on structural specifications and have no built-in support for specifications of operational semantics. Triskell has developped the Kermeta language to explore the idea of using aspect-oriented modeling to add precise action specifications with static type checking and genericity at the meta level, and examine related issues and possible solutions.

Kermeta consists of an extension to the Essential Meta-Object Facilities (EMOF) 2.0 to support behavior definition. It provides an action language to specify the body of operations in metamodels. This action language is imperative and object-oriented.

Kermeta is used in several use cases:

- to give a precise semantic of the behavior of a metamodel which then can be simulated.
- to act as a model transformation language.
- to act as a constraint language.

The development environment built for the Kermeta language currently provides the following tools

- an interpreter and a compiler that allow a metamodel to be executed.
- text and graphical editors, fully integrated within Eclipse, with syntax higlighting, code autocompletion.
- an Eclipse outline view, which allows navigation through the whole model and metamodel.
- various import/export transformations such as ecore2kermeta (kermeta text), kermat2ecore, kermeta2xmi (xmi version of your kermeta metamodel), xmi2kermeta, xmi2ecore.

Developped as an open source software under the terms of the EPL (Eclipse Public License), it has been first deposited to the APP (Agence de Protection des Programmes) in October 2005.

Thanks to Kermeta it is possible to build various frameworks dedicated to domain specific metamodels. Those frameworks are organised into MDKs (Model Development Kits). For example, Triskell porposes MDKs to work with the following metamodels: Java5, UML2, RDL (requirements), Ecore, Traceability, ...Some of these MDKs (UML2, RDL) are advanced enough to constitute a complete application.

In 2009, we've release the version 1.3.2 of Kermeta with a strong focus on the new java compiler to improve the overall performances.

## 5.2. Sintaks : Textual syntaxes for models

**Participants:** Erwan Brottier, Didier Vojtisek [correspondant].

The Sintaks tool enables to define bridges between concrete (textual files) and abstract syntax (models). It automates the process to build parser and pretty printer that are typically used by textual editors.

A bridge consists in a Sintaks model that defines the way to:

- parse a text in order to get the corresponding model (with respect to a given metamodel);
- explore a model in order to pretty print its textual representation.

Sintaks is based onto the EMF repository and then is compatible with most of the modeling tools of the MDA community running in Eclipse.

## 5.3. Kompose : Generic Model Composition Tool

**Participants:** Mickael Clavreul, Olivier Barais, Freddy Munoz, Benoit Baudry [correspondant].

Kompose is a generic framework to support model composition. The core composition mechanism is implemented in Kermeta as a separate metamodel that can be specialized for a specific domain metamodel in order to easily define composition operators for that domain. The framework is made of a generic model element merge algorithms and a directive language. The specialisation for a specific metamodel is done by defining appropriate signatures for the classes of this metamodel. As examples, Kompose currently includes specialisations for class diagrams, database schemas and feature models as in [60]. Kompose has been developed in collaboration with CSU in the context of the MATT équipe associée.

# 6. New Results

## 6.1. Model Driven and Aspect Oriented Design

### 6.1.1. Models at Runtime

**Participants:** Brice Morin, Olivier Barais, Grégory Nain, Noël Plouzeau, Mahmoud Ben Hassine, Jean-Marc Jézéquel.

We have developed a first proof of concept of the *models at runtime* idea [33], [21], [35], [23] to tame the complexity of dynamically adaptive systems by combining model-driven and aspect-oriented techniques and offering a high degree of automation and validation.

Since software systems need to be continuously available under varying conditions, their ability to evolve at runtime is increasingly seen as one key issue. Modern programming frameworks already provide support for dynamic adaptations. However the high-variability of features in Dynamic Adaptive Systems (DAS) introduces an explosion of possible runtime system configurations (often called modes) and mode transitions. Designing these configurations and their transitions is tedious and error-prone, making the system feature evolution difficult. While Aspect-Oriented Modeling (AOM) was introduced to improve the modularity of software, we have shown how an AOM approach can be used to tame the combinatorial explosion of DAS modes [33]. Using AOM techniques, we were able to derive a wide range of modes by weaving aspects into an explicit model reflecting the runtime system. We used these generated modes to automatically adapt the system. We validated our approach on Entimid, a platform for home automation in the context of AAL (Ambient Assisted Living) [35].. A demo of this application has been presented at several venues, including the "Fete de la Science" in Nov. 2009.

We also showed how aspects can be unwoven, based on a precise traceability metamodel dedicated to aspect model weaving. We analyzed traceability models, which describe how aspects were woven into a base, to determine the extent to which an aspect has affected the woven model in order to determine how it can be unwoven. Aspect unweaving is finally performed by applying inverse operations of a sub-sequence of the weaving operations in opposite order [31].

### 6.1.2. Managing Variability with Aspects

**Participants:** Brice Morin, Olivier Barais, Gilles Perrouin, Jean-Marc Jézéquel.

From a modeling point of view, the terms aspect and model can be considered synonymous. This notion of aspect goes beyond the usual meaning found in the Aspect Oriented Programming community where an aspect is often narrowly dened as the modularization of a cross-cutting concern. From this perspective, work on aspect-oriented approaches to modeling is important because it can yield signicant insights into how the MDE vision of software development can be realized. There is thus a growing community interested in the convergence of Aspect-Oriented Software Development (AOSD) and MDE ideas [17].

Domain-Specific Modeling Languages (DSMLs) describe the concepts of a particular domain and their relationships, in a meta-model. From a given DSML, it is possible to describe a wide range of different models. These models often share a common base and vary on some parts. Current approaches tend to distinguish the variability language from the DSMLs themselves, implying greater learning curve for DSMLs stakeholders and a significant overhead in product line engineering of DSLs. We propose [36], [30] to consider variability concepts as an independent aspect to be woven into the DSML to introduce variability capabilities. In particular we detail how variability is woven and how to perform product line derivation. We validate our approach through the weaving of variability into two very different metamodels: Ecore and SmartAdapter, our aspect-oriented modelling weaver, thus adding exibility in the weaving process itself. These results emphasize how new abilities of the language can be provided by this means.

### 6.1.3. Analyzing aspect-oriented programs

**Participants:** Romain Delamare, Freddy Munoz, Benoit Baudry.

We have analyzed 38 aspect-oriented open source projects with respect to the impact of aspects on the projects, and to coverage of the language features [39]. This reveals that AOP is currently used in a cautious way. This work is a first step to built support and development tools dedicated to actual practices for AOP, based on empirical usage profiles. We believe a possible reason for this distrust is the lack of dedicated tools for anlyzing, testing and evolving aspect-oriented programs.

A major issue when testign AspectJ programs is to ensure that the pointcut descriptor (PCD) matches the intended set of joinpoints. We have developed AdviceTracer to monitor the execution of advices in AspectJ programs and define dedicated oracles that can capture errors in the PCD [27]. In order to validate our approach, we also developed a mutation tool that systematically injects faults into PCDs[28]. Using these two tools, we perform experiments to validate that our approach can be applied for specifyi expected joinpoints to detect faults in the PCD. We have also shown that AdviceTracer is more adapted than JUnit to buid test cases that check PCDs [12].

## 6.2. Model-Based Testing

### 6.2.1. Testing model transformations

**Participants:** Sagar Sen, Naouel Moha, Freddy Munoz, Benoit Baudry, Jean-Marc Jézéquel.

Automatic model transformations play a critical role in MDE since they automate complex, tedious, error-prone, and recurrent software development tasks. A fault in a transformation can introduce a fault in the transformed model, which if undetected and not removed, can propagate to other models in successive development steps. Model transformations constitute a class of programs with unique characteristics that make testing them challenging. We have identified these characteristics [14] and identified promising solutions to overcome these barriers. This work is part of the research developed in the MoCAA Equipe associée in collaboration with Sudipto Ghosh, Robert France from CSU, Franck Fleurey from SINTEF and Jean-Marie Mottu.

Following this work about analysis of current barriers, we have developed several propositions for the automatic generation of test data for model transformation and composition. We have presented test criteria to cover the transformation's input domain. Coverage criteria are thus based on an adaptation of category partition on metamodel properties and combinatorial techniques [50]. These criteria serve as the basis for automatic generation of test models or automatic completion of model [22].

Automatic generation of test models requires a precise model of the effective input domain for the transformation. However, the declared source metamodel of a model transformation (e.g., UML) is usually an over approximation of the input domain. We have developed a systematic approach for pruning a large metamodel in order to obtain the effecitve source metamodel for the transformation [42]. Our tool Cartier can then use this to generate models and compare [41] the effectiveness of test criteria. experiments have shown that using partitioning strategies gives mutation scores of up to 87% vs. 72% in the case of unguided/random generation. These criteria and Cartier have then been used to synthesize models to test model composition engines[46].

### 6.2.2. Requirements Modeling for Early Analysis

**Participants:** Gilles Perrouin, Erwan Brottier, Benoit Baudry.

Ever-growing systems' complexity and novel requirements engineering approaches such as reuse or globalization imply that requirements are produced by different stakeholders and written in possibly different languages. In this context, checking consistency so that requirements specifications are amenable to formal analysis is a challenge. We have defined the R2A (which stands for Requirements for Analysis) platform to promote a better integration of techniques for requirements verification and validation within software development processes [11]. The core of the platform is a model composition process working at two modeling levels [40]. At the instance-level, it produces a global specification of requirements from a collection of partial, heterogeneous and potentially inconsistent specifications. At the design level (so called meta-level), it produces the internal formalism of the platform from design components which embedd operational, compositional and deployment semantics. As such, this process promotes the adaptability of the platform to various industrial contexts. This work has been done in collaboration with Yves Le Traon from University of Luxemburg.

### 6.2.3. Artificial Table Testing Dynamically Adaptive Systems

**Participants:** Freddy Munoz, Benoit Baudry.

Dynamically Adaptive Systems (DAS) are systems that modify their behavior and structure in response to changes in their surrounding environment. A major challenge for testing these systems is the combinatorial explosions of variants and environment conditions to which the system must react. Artificial Shaking Table Testing (ASTT) is a strategy inspired by shaking table testing (STT), a technique widely used in civil engineering to evaluate building's structural resistance to seismic events. ASTT makes use of artificial earthquakes that simulate violent changes in the environmental conditions and stresses the system adaptation capability. We model the generation of artificial earthquakes as a search problem in which the goal is to optimize different types of environmental variations and use ASTT as sampling technique to select test vectors for DAS [51].

### 6.2.4. Transforming and selecting functional test cases for security policy testing

**Participants:** Tejeddine Mouelhi, Benoit Baudry.

This work [37] considers typical applications in which the business logic is separated from the access control logic, implemented in an independent component, called the Policy Decision Point (PDP). The execution of functions in the business logic should thus include calls to the PDP, which grants or denies the access to the protected resources/functionalities of the system, depending on the way the PDP has been configured. The task of testing the correctness of the implementation of the security policy is tedious and costly. In this paper, we propose a new approach to reuse and automatically adapt existing functional test cases for specifically testing the security mechanisms. It includes a two step dynamic analysis technique based on mutation applied to security policies (RBAC, XACML, OrBAC). The method is applied to Java programs and provides tools for performing the two steps of the dynamic analyses. Three empirical case studies provide fruitful results and a first proof of concepts for this approach, e.g. by comparing its efficiency to an error-prone manual adaptation task. This work has been done in collaboration with Yves Le Traon from University of Luxemburg.

## 6.3. Meta-Modeling

### 6.3.1. Modeling Modeling

**Participant:** Benoit Baudry.

Model-driven engineering have permeated all branches of software engineering; to the point that it seems that we are using models, as Molière's Monsieur Jourdain was using prose, without knowing it. At the heart of modeling, there is a relation that we establish to represent something by something else. We have reviewed various definitions of models and relations between them [38]. Then, we have defined a canonical set of relations that express various kinds of representation relations and we proposed a graphical concrete syntax to represent these relations. Hence, this work is a contribution towards a theory of modeling. This is in collaboration with Frédéric Fondement from Université de Haute Alsace.

### 6.3.2. Model Driven Software Evolution

**Participants:** Naouel Moha, Régis Fleurquin, Olivier Barais, Jean-Marc Jézéquel.

Code and design smells are recurring design problems in software systems that must be identified to avoid their possible negative consequences on development and maintenance. In collaboration with the University of Montréeal and the ADAM project team in Lille, we introduced an approach to automate the generation of detection algorithms from specifications written using a domain-specific language [19]. This language is defined from a thorough domain analysis. It allows the specification of smells using high-level domain-related abstractions. It allows the adaptation of the specifications of smells to the context of the analysed systems [18].

We also studied the problem of *God Classes* (large classes which know too much or do too much in an OO design): how they arise, how prevalent they are, and whether they remain or they are removed as the systems evolve over time, through a number of versions [43]. We showed how to detect the degree of godliness of classes automatically, distinguishing between those classes that are so by design (good code) from those that occurred by accident (bad code). This methodology can guide software quality teams in their efforts to implement prevention and correction mechanisms.

However, these kind of model analysis, or other model transformations such as refactorings specified for a given language cannot be readily reused for another language because their related metamodels may be structurally different. To solve this problem, we have discovered [32] an approach allowing the specification of generic model transformations, in particular refactorings, so that they can be applied to different metamodels. Our approach relies on two mechanisms: (1) an adaptation based mainly on the weaving of aspects; (2) the notion of model typing, an extension of object typing in the model-oriented context. We validated our approach by performing some experiments that consisted of specifying three well known refactorings (Encapsulate Field, Move Method, and Pull Up Method) and applying each of them onto three different metamodels (Java, MOF, and UML).

### 6.3.3. Executable Software Process Modeling

**Participants:** Benoit Combemale, Jean-Marc Jézéquel.

Describing and managing activities, resources and constraints of software development processes is a challenging goal for many organizations [45]. We proposed a comparison of UML-based Software Process Modeling Languages (SPMLs) [15].

One major advantage of executable software process models is that once defined, they can be simulated, checked and validated in short incremental and iterative cycles. This also makes them a powerful asset for important process improvement decisions such as resource allocation, deadlock identification and process management. We proposed a framework that combines Aspect and Model-Driven Engineering approaches in order to ensure process modeling, simulation and execution [26]. This framework is based upon UML4SPM, a UML2.0-based language for Software Process Modeling and Kermeta, an executable metaprogramming language. This work has been done in collaboration with Reda Bendraou from the UPMC.

# 7. Contracts and Grants with Industry

## 7.1. SPEEDS (IST)

**Participants:** Jean-Marc Jézéquel, Gilles Perrouin, Olivier Barais.

SPEEDS is an IST Integrated Project defining the new generation of end-to-end methodologies, processes and supporting tools for safety-critical embedded system design. They will enable European systems industry to evolve from model-based design of hardware/software systems, towards integrated component based construction of complete virtual system models.

SPEEDS partners are companies active in the entire supply chain: OEMs, suppliers, and tool vendors, supported by leading European research institutions. The technical pillars of the SPEEDS approach are:

- A semantics-based modeling method to support the construction of complex embedded systems by composing heterogeneous subsystems while enabling sound integration of new and existing tools. This modeling approach defines “rich-component” models to represent both functional and non-functional aspects so that efficient implementations can be derived from abstract models.
- Novel formal analysis tools and techniques to assess precisely properties of the system that will allow to explore architectural alternatives of implementation platforms and enable correct-by-construction designs. Compositionality and abstractions will make this approach scalable for large systems.
- A new tool-supported process, controlled speculative design, minimizing the risk of concurrent design activities by establishing formal “contracts” between inter- and intracompany design groups.

Triskell mainly participates to the SP2 work package named heterogeneous rich components (HRC) to define a semantic-based common meta-model, which forms the foundations for the component based construction of complete virtual system models. In this context, Triskell mainly implements this year the static semantics checker of speeds model. The static semantics checker checks that a speeds model is statically well-formed. This checker is connected to the speeds bus and it can be invoked as a bus services. Triskell also provides supports on MDE tools and MDE techniques that can facilitate the integration of partners's tools.

Project duration: 2006-2009

Triskell budget share: 201 keuros

Project Coordinator: Airbus

Participants: Airbus Deutschland GmbH (A-D), Airbus France S.A.S. (A-F), DaimlerChrysler AG (DC), Israel Aircraft Industries Ltd (IAI), Robert Bosch GmbH, INRIA, Kuratorium OFFIS e.V., PARADES, Universite Joseph Fourier, TNI, I-Logix Israel Ltd, Extessy AG, Knorr Bremse Fekrendszerek Kft, Steyr GmbH & Co KG, SAAB AB, Esterel Technologies SA

## 7.2. DiVA (IST)

**Participants:** Jean-Marc Jézéquel, Benoit Baudry, Brice Morin, Freddy Munoz, Olivier Barais, Didier Vojtisek.

The goal of DiVA is to provide a tool-supported methodology for managing dynamic variability of co-existing, co-dependent configurations in adaptive systems that span system administration and platform boundaries. Examples of such adaptive systems are communication infrastructure in rescue operations and mobile entertainment environments. This will be addressed through a combination of aspect-oriented and model-driven techniques. DiVA will explore how adaptation policies can be captured in the requirements, how aspects can model the variants used to adapt the system, how models can be kept at runtime to drive the adaptation and which validation techniques have to be developed in this context.

The Triskell team participates mainly in the definition of models that can drive the adaptation at runtime. The benefits of keeping models at runtime is to have an abstract view of the adaptation policies and mechanisms on which it is possible to reason (to check invariants, QoS properties, etc.) before actually adapting the running system. One important challenge tackled by Triskell is a mechanism to synchronize the running system with the model that has been adpated according to the changes in the environment. Triskell is also involved in the different validation tasks that occur when building such systems and when adapting these systems at runtime. An important issue for validation at design time is to select a subset of all possible configurations for testing. At design time, it is necessary to validate interactions between variants and to check that invariants on the system are satisfied.

Project duration: 2007-2010

Triskell budget share: 400 keuros

Project Coordinator: SINTEF

Participants: SINTEF, Uni. Lancaster, INRIA, Pure Systems, Thales IS, CAS.

## 7.3. S-Cube (Network of Excellence)

**Participants:** Jean-Marc Jézéquel, Noël Plouzeau, Olivier Barais, Grégory Nain, Sagar Sen, Maha Driss.

S-Cube, the Software Services and Systems Network, will establish an integrated, multidisciplinary, vibrant research community which will enable Europe to lead the software-services revolution, thereby helping shape the software-service based Internet which is the backbone of our future interactive society.

An integration of research expertise and an intense collaboration of researchers in the field of software services and systems are needed to address the following key problems:

- Research fragmentation: Current research activities are fragmented and each research community (e.g., grid computing or software engineering) concentrates mostly on its own specific techniques, mechanisms and methodologies. As a result the proposed solutions are not aligned with or influenced by activities in related research fields.
- Future Challenges: One challenge, as an example, is to build service-based systems in such a way that they can self-adapt while guaranteeing the expected level of service quality. Such an adaptation can be required due to changes in a system's environment or in response to predicted and unpredicted problems.

S-Cube will pursue the following objectives which will have a long-lasting impact on European research:

- Re-aligning, re-shaping and integrating research agendas of key European players from diverse research areas. By synthesizing and integrating diversified knowledge, a long-lasting foundation for steering research and for achieving innovation at the highest level will be achieved.
- Inaugurating a Europe-wide common program of education and training for researchers and industry. This will create a common culture that will have a profound impact on the future of the field.
- Establishing a pro-active mobility plan to enable cross-fertilisation, which will foster the integration of research communities and the establishment of a common software services research culture.
- Establishing trust relationships with industry. Via European Technology Platforms (specifically NESSI) a catalytic effect in shaping European research, strengthening industrial competitiveness and addressing main societal challenges will be accomplished.
- Defining a broader research vision and perspective. This will shape the software-service based Internet of the future and will accelerate economic growth and improve the living conditions of European citizens.

Two INRIA project-teams participate to this NoE. Paris and Triskell. The Work in S-Cube clearly distinguishes between principles and methods for engineering and adapting service-based systems and the technology which is used to realize those systems while taking into account cross-cutting issues like Quality of Service (QoS) and SLA compliance. Consequently two two joint research activities has been designed. Triskell mainly participates to the first one. which is concerned with engineering and adaptation methodologies for Service-based applications. It combines different research efforts form the requirements engineering discipline, the human computer interaction discipline and the software engineering, adaptation and testing disciplines. This year, Triskell mainly works on the adaptation framework for services in unifying runtime adaptation and design evolution for services based systems.

Project duration: 2008-2012

Triskell budget share: 150 keuros

Project Coordinator: Prof. Dr. Klaus Pohl (Project Coordinator), University of Duisburg-Essen, Germany – Prof. Dr. Mike Papazoglou (Scientific Director), Tilburg University, The Netherlands

Participants: University of Duisburg-Essen, Tilburg University, City University London, Consiglio Nazionale delle Ricerche, Center for Scientific and Technological Research, The French National Institute for Research in Computer Science and Control, Lero - The Irish Software Engineering Research Centre, Politecnico di Milano, MTA SZTAKI - Computer and Automation Research Institute, Vienna University of Technology, Université Claude Bernard Lyon, University of Crete, Universidad Politécnica de Madrid, University of Stuttgart

## 7.4. AOSD-Europe (Network of Excellence)

**Participants:** Jean-Marc Jézéquel, Noël Plouzeau, Olivier Barais, Didier Vojtisek.

Aspect-Oriented Software Development (AOSD) supports systematic identification, modularisation, representation and composition of crosscutting concerns such as security, mobility, distribution and resource management. Its potential benefits include improved ability to reason about the problem domain and corresponding solution; reduction in application code size, development costs and maintenance time; improved code reuse; architectural and design level reuse by separating non-functional concerns from key business domain logic; improved ability to engineer product lines; application adaptation in response to context information and better modelling methods across the lifecycle. AOSD-Europe will harmonise and integrate the research, training and dissemination activities of its members in order to address fragmentation of AOSD activities in Europe and strengthen innovation in areas such as aspect-oriented analysis and design, formal methods, languages, empirical studies and applications of AOSD techniques in ambient computing. Through this harmonisation, integration and development of essential competencies, the AOSD-Europe network of excellence aims to establish a premier virtual European research center on AOSD. The virtual research centre will synthesise the collective viewpoints, expertise, research agendas and commercial foci of its member organisations into a vision and pragmatic realisation of the application of AOSD technologies to improve fundamental quality attributes of software systems, especially those critical to the information society. It will also act as an interface and a centralised source of information for other national and international research groups, industrial organisations and governmental bodies to access the members' work and enter collaborative initiatives. The existence of such a premier research base will strengthen existing European excellence in the area, hence establishing Europe as a world leader. (http://www.aosd-europe.net/)

Project duration: 2004-2009

Triskell budget share: 150 keuros

Project Coordinator: University of Lancaster

Participants: University of Lancaster, Technical University of Darmstadt, INRIA, VUB, Trinity College Dublin, University of Malaga, Katholieke Universiteit Leuven, Technion, Siemens, IBM Hursley Development Laboratory

## 7.5. Mopcom Hard (RNTL)

**Participants:** Jean-Marc Jézéquel, Didier Vojtisek, Gilles Perrouin, Cyril Faucher.

Mopcom hard is a RNTL project supported by the Competitivity Cluster "Images & réseaux" of Brittany. The project focuses on the use of model driven engineering for the development of embedded system typically based on system-on-chip (SOC). The project will produce a complete methodology and development environment dedicated to the domain.

In 2009, Triskell participated to the development process and the specification of precise metamodels (using Kermeta) for each steps of the process. Triskell also studied and produced tools using Kermeta Java compiler for the MARTE UML profile as it the main metamodel for several of these steps .

Project duration: 2007-2010 years

Triskell budget share: 101 keuros

Project Coordinator: Thalès (TSA)

Participants: Thalès Systèmes Aéroportés, Thomson, Sodius, ENSIETA, LESTER, Supelec Rennes, INRIA

## 7.6. Mopcom Ingénierie (Competitivity Cluster I&R)

**Participants:** Jean-Marc Jézéquel, Didier Vojtisek, Olivier Barais, Mickael Clavreul.

Mopcom Ingénierie is a project of the Competitivity Cluster "Images & réseaux" of Brittany. The project focuses on the use of model driven engineering for the development of Software for Image domain. The project will produce a complete methodology and development environment dedicated to the domain.

In 2009, Triskell developed a solution to address the Thomson case study. It provides a solution to easily integrate legacy systems with MDE.

Project duration: 2008-2011 years

Triskell budget share: 150 keuros

Project Coordinator: Thalès (TSA)

Participants: Thalès Systèmes Aéroportés, Thomson, Sodifrance, ENSIETA, INRIA, ENST Bretagne, Valoria, Orange Labs

## 7.7. Orange Labs

**Participants:** Jacques Simonin, Jean-Marc Jézéquel.

Since March 2006, we have a collaboration with Orange Labs (France Télécom R& D), Lannion on applying MDE techniques to telecom operator IS. In this context, Jean-Marc Jézéquel acts as Ph.D advisor for Jacques Simonin and Mariano Belaunde, and Slim Ben Hassine, all being senior Orange Labs engineers.

Project duration: 2006-2011

Triskell budget share: 25 keuros

## 7.8. DOMINO (RNTL)

**Participants:** Benoit Baudry, Jean-Marc Jézéquel, Sagar Sen.

The DOMINO project (Methods and processes for domain specific modelling) is funded by the french agency for research (ANR). It aims at proposing a development process based on a multi-view description of a system, each view being expressed with various domain specific modelling languages. Model-driven engineering is the core technology to define this process and is used to validate and verify the different artefacts produced at different steps of the process. A reliable process is crucial in the context of a multi-formalism approach to modelling. This process encompasses all the techniques needed to design, validate, and improve the software artefacts.

Triskell develops techniques to validate and test model transformations that are used to automate different steps of the process. These techniques are based on model synthesis techniques for automatic test input generation and on contracts to check the results of test cases. We also propose an incremental process to build and improve trust in model transformations that are encapsulated as reusable components.

The project ended in June 2009. The final review by ANR was positive, the DOMINO project reached the objectives stated in 2007 and partners in the project showed an actual collaboration in order to deliver significant results for quality improvement in model-driven development. As a result of DOMINO, we reported on the experiments performed on several industrial embedded system applications from the aviation, space and military areas. In particular we show how the Context Description Language can allow a better integration of formal methods with engineering practices [29].

Project duration: 2006-2009

Triskell budget share: 79 keuros

Project Coordinator: IRIT

Participants: IRIT, Airbus, Sodifrance, CNES, CEA-LIST, ENSIETA, INRIA/Triskell

## 7.9. OpenEmbeDD (RNTL)

**Participants:** Jean-Marc Jézéquel, Didier Vojtisek, Vincent Mahé.

OpenEmbeDD is a RNTL project which build an Eclipse open-source platform based on the MDE approach for developing Real-Time and Embedded systems. OpenEmbeDD integrates the technologies based on formal models from synchronous/asynchronous/mixed paradigms. This platform covers the 2 branches of the V cycle : specification/design/implementation et checking/validation. The building of the platform is in synergy with the Competitivity Clusters "SYSTEM@TIC-Paris Région" (Ile de France), "Aéronautique-Espace, Systèmes Embarqués" (Midi-Pyrénées) and "Images et Réseaux" (Bretagne). The platform is adopted in the research program CARROLL, this program is led for 2 years by the CEA, INRIA and THALES that are at the initiative of the OMG MARTE standard.

The main topics of the project are:

- Formal approach (abstraction, proof, model-checking, transformations).
- Modeling of Real-Time requirements.
- Modeling of Real-Time properties (components, systems,...).
- Process and tools for checking and validating (proof, tests,...).
- Languages and tools for describing and designing architectures.

A part of the core of the platform is the metamodeling language Kermeta that is developed by the Triskell project team. In this context, Triskell has developped tools for metamodelling engineering (a graphical editor for Kermeta, a metamodel compiler). Triskell's members participate to the specification of the source generator for building automatically graphical editors. Triskell is also involved in the integration team who coordinates, tests and integrates the works of all the partners.

Project duration: 2006-2009

Triskell budget share: 300 keuros

Project Coordinator: INRIA

Participants: Airbus, Anyware Technologies, CEA-List, CS-SI, France Telecom, INRIA, LAAS, THALES (DAE and RT), Verimag

## 7.10. Faros (RNTL)

**Participants:** Noël Plouzeau, Kathia Marcal de Oliveira, Jean-Marc Jézéquel.

Faros is a project supported by the RNTL program. The Faros project has started in march, 2006 and has ended on october, 2009. The general objective of the project was the definition and the construction of a software process and tool chain to build reliable Web service based application. The process and its corresponding tool chain are be able to accept as input domain specific, platform independent components. The tool can generate platform specific implementations of these components, interconnected through Web services.

The general strategy of the process is based on model engineering. The project's workpackages are organized as follows:

1. definition of metamodels for managing business specific application description;
2. definition of metamodels for Web services platforms;
3. definition of a general metamodel to describe pivot models, which are business and platform independant;
4. definition of transformations to generate Web services implementation from business specific models, using automated model transformation techniques.

The project uses applications of the industrial partners (France Telecom, Electricité de France and Alicante) as case studies to validate the process and its tool chain.

Within the Faros RNTL project, the Triskell project is responsible for the metamodelisation activity, the supervision of transformation designs and the production of the model transformation engine. More precisely, the core of the tool chain is based on the Kermeta model transformation engine, which is being developed entirely by the Triskell team.

Project duration: 2006-2009

Triskell budget share: 80 keuros

Project type: exploratory

Project Coordinator: France Telecom

Participants: France Telecom R&D, EDF R&D, Alicante (industrial partners), university of Nice (I3S laboratory), university of Rennes 1 (IRISA laboratory), university of Lille (LIFL laboratory)

## 7.11. ITEA2 OPEES

**Participants:** Jean-Marc Jézéquel, Didier Vojtisek, Benoit Baudry, Benoit Combemale.

OPEES is an ITEA2 project which goal is to build a community able to ensure long-term availability of innovative engineering technologies in the domain of software-intensive embedded systems. Its main benefits should be to perpetuate the methods and tools for software development, minimize ownership costs, ensure independence of development platform, integrate, as soon as possible, methodological changes and advances made in academic world, be able to adapt tools to the process instead of the opposite, take into account qualification constraints. In this purpose, OPEES relies on the Eclipse Modelling Project platform (EMF, GEF, GMF, OCL, UML2, ...) and on many available tools such as Kermeta.

The participation of Triskell into the OPEES project aims at industrializing both Kompose and MDE-test. Kompose is a model composition framework based on Aspect Oriented Software Development research results. Kompose allows a better modularization and separation of concerns when building tools around meta-models. It features powerful model-level pointcut specification and matching mechanisms allowing semantic based weaving of model level aspects. Kompose builds on and is nicely integrated with Kermeta, but it can also be used independently on any kind of EMF model or meta-model. MDE-test is a model transformation testing framework that makes it possible to synthesize input data (ie test models) for model transformations and check that the transformation behaves "correctly" on them.

Project duration: 2009-2011

Triskell budget share: 150 keuros

Project Coordinator: Airbus

Participants: AIRBUS, ADACORE, Anyware Technologies, Astrium Satellites, Atos Origin, CEA LIST, CNES, C-S, Dassault, EADS Astrium ST, ENAC, INPT-IRIT, INRIA (AtlanMod/EXPRESSO/TRISKELL), MBDA, OBEO, ONERA, Schneider Electric, Thales, Xipp

## 7.12. Artemis CHESS

**Participants:** Noël Plouzeau, Jean-Marc Jézéquel, Jacques Falcou.

CHESS is an Artemis project that seeks industrial-quality research solutions to problems of property-preserving component assembly in real-time and dependable embedded systems, and supports the description, verification, and preservation of non-functional properties of software components at the abstract level of component design as well as at the execution level. CHESS develops model-driven solutions, integrates them in component-based execution frameworks, assesses their applicability from the perspective of multiple domains (such as space, railways, telecommunications and automotive), and verifies their performance through the elaboration of industrial use cases.

Triskell will contribute to the definition of the component metamodel, to the design of process. Triskell wil be the main contributor of model transformation tools, by adapting its Kermeta platform to the Chess process and by contributing to the interconnection of external tools from industrial tool provider partners.

Project duration: 2009-2011

Triskell budget share: 400 keuros

Project budget: 6 M euros

Project Coordinator: INTECS

Participants: AICAS, Aonix, Atos Origin, ENEA, Ericsonn, Fraunhofer, FZI, GMV, INRIA (Triskell), INTECS, Thales Alenia Space, THALES Communications, UPM, University of Padua, X/Open

## 7.13. ANR Movida

**Participants:** Olivier Barais, Jean-Marc Jézéquel.

Movida is an ANR project which goal is to provide a solution for modelling view in system engineering and to provide decision support for architect. Today, and likely for a long time to come, the complexity of software dominant systems is still growing and the variety of system classes tends to expand. From embedded systems which are required to cope with spare resources, to system of systems for which the evolvability and flexibility is key, requirements classes are expanding. In addition new concerns or more stringent existing concerns bring their extra complexity. They are environmental concerns, maintenance, repair and operation (MRO) concerns, supply management concerns etc. All of them play today an active or even sometime decisive role in the engineering decision process. The difficulty to embrace the whole complexity of the concerns and the difficulty to manage their inter-relations has raised the interest of the engineering community for "concerns driven" engineering. This is addressed today in the model driven engineering research community through the exploration of "viewpoint modelling" technologies. The aim of the MOVIDA project is to provide a support to model-driven viewpoint engineering through:

- Defining and specifying the underlying concepts that must be shared and used when implementing an engineering solution supporting viewpoint management.
- Providing a support to the definition of specific viewpoints, enabling their composition in a consistent whole that fits a specific project needs.
- Managing the consistency of an information bulk made of several views on a system which is accessed, modified and managed by different stakeholders during the system definition process.
- Applying decision-support tools to multi-viewpoint modelling frameworks so as to support architectural trade-offs.

Triskell mainly works this year on the state of the arts and the conceptual meta-model for viewpoint modelling.

Project duration: 2009-2011

Triskell budget share: 184 keuros

Number of person/years: 1,2

Project Coordinator: Thales

Participants: Thales, OBEO, Université Paris 6, INRIA (Triskell)

## 7.14. IDA

**Participants:** Olivier Barais, Grégory Nain.

IDA is a project which goal is to study how technologies can help people to stay as long as possible at home. Industrials, associations and public institutions of the metropolis of Rennes, are working together on the IDA project which aims to allow dependent people to stay at home as long as possible. Due to the large scale of the project, and the diversity of disabilities that have to be considered, the deployment context will be different for each equipped house. The technologies used will vary, in order to compensate handicaps or because a technology is already installed, and people do not want it to be removed. Moreover, the system installed in these houses will have to provide a remote access to the devices of the house, and transmit all the necessary information from the sensors of the house to a control center where information will be treated. INRIA helps in providing support for integrating devices and services and providing a solution for managing variability.

Project duration: 2008-2010

Triskell budget share: 10 keuros

Project Coordinator: ASSAD Pays de Rennes

Participants: http://www.ida-autonomie.fr/partenaires.php

# 8. Other Grants and Activities

## 8.1. GALAXY

**Participants:** Olivier Barais, Mahmoud Ben Hassine, Jean-Marc Jézéquel, Didier Vojtisek.

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

galaxy aims at providing an open SOA platform, enabling agility using dynamic architectures. galaxy will provide an integrated environment by assembling and leveraging INRIA's technologies. galaxy covers a large scope of functionalities, from design tooling to adaptable and extensible runtime, offering monitoring and management advanced functions.

Galaxy allows to design, deploy, run, monitor systems, following concepts and paradigms inherited from service-oriented, process and dynamic architectures, and offering a set of management functions for agile and dynamic systems. galaxy technologies are most of them compliant with the Eclipse and the SCA standards.

Galaxy, a platform unifying highy standard adaptable, extensible and agile runtime, and agile monitoring and modelling capabilities, built on top of INRIA's technologies:

- component-based models and frameworks Fractal and GCM (Grid Component Model) promoted by the ETSI
- FraSCAti and ProActive Java platforms
- Eclipse STP-IM, SOA Intermediate Model, for enabling interoperability b/w SOA-related standards at design- and run-time, such as BPMN, BPEL, SCA and JBI
- Kermeta MDE technologies
- WildCAT extensible framework for context-aware applications

Triskell EPI mainly work in collaboration with ASCOLA to integrate Wildcat monitoring framework and Kermeta to the Galaxy platform.

Project duration: 2008-2010

Triskell budget share: One associated engineer shared with Ascola EPI

Project Coordinator: Alain Boulze Tuvalu INRIA Project.

Participants: ECOO, ADAM, ASCOLA, TUVALU, SARDES, OASIS, TRISKELL.

## 8.2. National projects

### *8.2.1. CNRS GDRs*

The Triskell project is connected to the national academic community through a lightweight participation to several CNRS GDR (Groupement de Recherche).

- GDR ASR: Action IDM (on Model Driven Engineering) (http://www.actionidm.org)
- GDR GPL: Génie de la Programmation et du Logiciel (http://www-lsr.imag.fr/GPL), where Jean-Marc Jézéquel is a member of the scientific committee.

## 8.3. International working groups

### *8.3.1. ERCIM Working Group on Software Evolution*

Numerous scientific studies of large-scale software systems have shown that the bulk of the total software-development cost is devoted to software maintenance. This is mainly due to the fact that software systems need to evolve continually to cope with ever-changing software requirements. Today, this is more than ever the case. Nevertheless, existing tools that try to provide support for evolution have many limitations. They are (programming) language dependent, not scalable, difficult to integrate with other tools, and they lack formal foundations.

The main goal of the proposed WG (http://w3.umh.ac.be/evol/) 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, we plan to become a Virtual European Research and Training Centre on Software Evolution.

Triskell contributes to this working group on the following points:

- re-engineering and reverse engineering
- model-driven software engineering and model transformation
- impact analysis, effort estimation, cost prediction, evolution metrics
- traceability analysis and change propagation
- family and product-line engineering

### *8.3.2. Standardization in Eclipse projects*

In 2009, Triskell project participates to discussion around the creation of the MXF eclipse project (http://www.eclipse.org/proposals/mxf/). This new Eclipse Modeling Framework Technology (EMFT) sub project proposes ideas similar to those included in Kermeta and can be a good candidate for transferring some Kermeta concepts to an Eclipse project.

### *8.3.3. Standardization at OMG*

In 2009, Triskell project participates to normalization actions at OMG (http://www.omg.org/). It was involved in the MARTE FTF (Finalization Task Force)and was interrested in the Analysis and Design group which promotes standard modelling techniques including UML and MOF.

### *8.3.4. Collaboration with foreign research groups*

- Colorado State University (CSU), USA. In January 2006 we started a "Equipe associée" (a three year program for an associated team) called MATT between CSU and Triskell. In 2009 the Equipe associée has been renewed under the name MoCAA which stands for Model Composition Aspects and Analysis (see http://www.irisa.fr/triskell/matt for details). In this context, Robert France and Benoit Baudry gave lectures at the International School on Model-Driven Design for Distributed, Real time, Embedded Systems MDD4DRES. Romain Delamare, Benoit Baudry, Freddy Munoz and Brice Morin visited CSU in 2009. Minhazur Rahman and David Janovy , currently PhD students in CSU, visited Triskell in summer 2009 as part of the REUSSI program which supports travel for US groups involved in Equipe associées. We have a paper in CACM [14] and one at ICST'09 [27] (selected as one of the best papers).

- University of Luxemburg. Since 2009 Triskell is involved in a collaborative project called SPLIT: Combine Software Product Line and Aspect-Oriented Software Development (with Nicolas Guelfi and Jacques Klein), that is funded by both the PICS program of CNRS and the FNR of Luxemburg. This project is providing the backgroud and the funding for Paul Istoan's PhD thesis, done in co-tutelle between University of Rennes and University of Luxemburg. As an initial research result, we showed how aspects can be unwoven, based on a precise traceability metamodel dedicated to aspect model weaving [31].
- Modelling Simulation and Design Lab, Mc Gill University, Montreal Sagar Sen has started a PhD as a co-direction between IRISA and Mc Gill. This PhD is about automatic model synthesis through constraint solving in the context of model transformation testing and automatic exploration of large design spaces.
- Since February 2007 Triskell participates to a "FACEPE" project (a two year program with the University of Pernambouco, Brazil) called SIntArch (Safe Introduction of Interaction patterns in Component Based Software Architectures) between Pr Augusto Cesar Alvez Sampaio and the Triskell group on Component Based Software Architecture design using : Model-driven engineering and Aspects Oriented Modeling. In 2009, we have consolidated the model-snippet MDK used in Kermeta and based our models@runtime platform on top of this MDK.
- Budapest University of Technology and Economics Since 2008, Benoit Baudry is associate researcher in the IP project SENSORIA (Software Engineering for Service Oriented Overlay Computers). In this context, we collaborate with Daniel Varro from Budapest University of Technology and Economics on testing techniques for model transformations.

# 9. Dissemination

## 9.1. Scientific community animation

### *9.1.1. Journals*

*9.1.1.1. Jean-Marc Jézéquel*

is an Associate Editor of the following journals:

- Journal on Software and System Modeling: SoSyM
- Journal of Object Technology: JOT

### *9.1.2. Examination Committees*

*9.1.2.1. Jean-Marc Jézéquel*

was in the examination committee of the following PhD thesis and "Habilitation à Diriger les Recherches":

- Jacques Simonin, January 2009, université de Rennes (adviser);
- Mireille Blay-Fornarino (HDR), April 2009, Univ. Nice-Sophia Antipolis (referee);
- Marc Léger, May 2009, Ecole des Mines ParisTech (member);
- Jean-Rémy Falleri, October 2009, université de Montpellier 2 (referee);
- Xavier Blanc (HDR), November 2009, université de Paris 6 (member);
- Arnaud Blouin, November 2009, université d'Angers (referee);
- Yann Glouche, December 2009, université de Rennes (president);
- Dhaminda Abeywickrama, December 2009, Monash Univ. (referee);
- Romain Delamare, December 2009, université de Rennes (member);

*9.1.2.2. Benoit Baudry*

was in the examination committee of the following PhD thesis

- Emmad Saadeh, October 2009, University of Pretoria (referee)
- Olivier Le Goaer, October 2009, université de Nantes (member)
- Romain Delamare, December 2009, université de Rennes (member)
- Erwan Brottier, December 2009, université de Rennes (member)

*9.1.2.3. Regis Fleurquin*

was in the examination committee of the following PhD thesis

- Reda Kadri, January 2009, Université de Bretagne Sud (member)

### *9.1.3. Conferences*

*9.1.3.1. Jean-Marc Jézéquel*

has been a member of the program committee of the following conferences:

- ICSE 2009 The 31th International Conference on Software Engineering, Vancouver, Canada, May 16-24, 2009
- AOSD 2009 Eighth International Conference on Aspect-Oriented Software Development, Charlottesville, Virginia, March 2-6, 2009
- AFADL 2009 Toulouse, 26 - 28 janvier 2009

*9.1.3.2. Benoit Baudry*

has been a member of the program committee of the following conferences:

- MODELS 2009 The 12th International Conference on Model Driven Engineering Languages and Systems Denver, USA, october 2009
- IEEE ICST'09 International Conference on Software Testing Verification and Validation, Denver, CO, USA, April 2009
- CAMPUS'09, 2nd Workshop on Context-aware Adaptation Mechanisms for Pervasive and Ubiquitous Services, Lisbon, Portugal, June 2009
- MoDeVVa 2009, 6th international workshop on Model design and Validation at MODELS'09, Denver, CO, USA, October 2009
- MoDSE-MCCM workshop on models and evolution, at MODELS'09, Denver, CO, USA, October 2009
- Mutation'09 workshop at ICST'09, Denver, CO, USA, April 2009

*9.1.3.3. O. Barais*

has been a member of the programme committee of the following conferences:

- 5èmes Journées sur l'ingénierie Dirigée par les Modèles (IDM'09), Nancy, France, March 2009.
- 3ème Conférence Francophone sur les Architectures Logicielles (CAL'09), March 2009
- 15ème Conférence Francophone sur les Langages et Modèles à Objets (LMO'09), mars 2009
- 2nd Workshop on Context-aware Adaptation Mechanisms for Pervasive and Ubiquitous Services, (CAMPUS'09), Lisbon, Portugal, June 2009.

*9.1.3.4. R. Fleurquin*

has been a member of the programme committee of the following conferences:

- Journal for Universal Computer Science, Special Issue on Software Components, Architectures and Reuse 2009
- 5èmes Journées sur l'ingénierie Dirigée par les Modèles (IDM'09), Nancy, France, March 2009.

### *9.1.4. Workshops, Tutorials and Keynotes*

J.-M. Jézéquel gave invited talks at Mc Gill University (Barbados Workshop), at Luxemburg University, at the Technical University Munich, at the Scientific Board meeting of INRIA, the University of Nice-Sophia Antipolis, Telecom Bretagne, and gave a keynote address at the INFORSID-IDM day in Paris. He also gave a tutorial on Model Driven Language Engineering with Kermeta at MODELS'09, GTTSE'09 and EJCP'09.

Benoit Baudry gave an invited talk on Aspect-Oriented Modeling for the 'journées IDM'.

Olivier Barais gave an invited talk on Aspect-Oriented Software Development for the SSAIE Summer School 09.

## 9.2. Teaching

The Triskell team bears the bulk of the teaching on Software Engineering at the University of Rennes 1, at the levels M1 (Project Management, OO Analysis and Design with UML, Design Patterns, Component Architectures and Frameworks, V&V) and M2 (Model driven Engineering, Aspect-Oriented Software Development, Software Product Lines, Component Based Software Development, etc.).

Each of Jean-Marc Jézéquel, Noël Plouzeau, Olivier Barais are teaching about 200 h in these domains, with Benoit Baudry teaching about 50h, for a grand total of about 650 hours, including several courses at ENSTB and INSA Rennes.

Olivier Barais is the overall responsible for the Master2 Pro in Computer Science at the University of Rennes.

The Triskell team also receives several Master and summer trainees every year.

## 9.3. Miscellaneous

The Triskell team represented INRIA at the "Fete de la Science", organized in Rennes 20–22 November 2009, with a demo of Kermeta and Entimid applied to Ambiant Assited Living.

- J.-M. Jézéquel is Deputy Director of MATISSE Doctoral School. He is appointed to the board of the Committee of Projects of INRIA Rennes. He is a member of the Steering Committee of the AOSD and the MODELS Conferences series. He is a member of the Scientific Committee of the GDR GPL of CNRS. He belongs to the evaluation committee of the SIO division of DGA (Direction Générale de l'Armement). He is a Member of the Architecture Board of the MDDi Eclipse project. He participated to the creation of IFIP WG 10.2 on Embedded Systems. He is a member of the Advisory Board of the NSF REMODD Project (Repository for Model Driven Development). He is General Chair of the 9th International Conference on Aspect-Oriented Software Development AOSD 2010 to be held in Rennes and Saint-Malo on March 15-19, 2010.
- P.-A. Muller is a member of the Steering Committee of the MODELS/UML Conferences series. He has been nominated as Executive Vice President of the Université de Haute Alsace. He is member of the board of the Cocktail-ERP Open Source Consortium.
- Benoit Baudry is on the steering committee of the IEEE International Conference on Software Testing Verification and Validation. He is co-editor for a special issue of the Information ans Software Technology journal on mutation analysis. He has participated in the evaluation of collborative research projects for the Austrian Research funding agency. He has received, with P.-A. Muler and F. Fondement the ACM distinguished paper award at MODELS'09 for a work entitled 'Modeling Modeling'.

# 10. Bibliography

## Major publications by the team in recent years


[1] B. BAUDRY, F. FLEUREY, J.-M. JÉZÉQUEL, Y. LE TRAON. *Automatic Test Cases Optimization: a Bacteriologic Algorithm*, in "IEEE Software", vol. 22, n^o 2, March 2005, p. 76–82.

[2] A. BEUGNARD, J.-M. JÉZÉQUEL, N. PLOUZEAU, D. WATKINS. *Making Components Contract Aware*, in "IEEE Computer", vol. 13, n^o 7, July 1999.

[3] J.-M. JÉZÉQUEL. *Real Time Components and Contracts*, in "Model Driven Engineering for Distributed Real Time Embedded Systems", S. GÉRARD, J.-P. BABAU (editors), Hermes Science Publishing Ltd, London, 2005.

[4] J.-M. JÉZÉQUEL. *Reifying Variants in Configuration Management*, in "ACM Transaction on Software Engineering and Methodology", vol. 8, n^o 3, July 1999, p. 284–295.

[5] J.-M. JÉZÉQUEL. *Model Driven Design and Aspect Weaving*, in "Journal of Software and Systems Modeling (SoSyM)", vol. 7, n^o 2, may 2008, p. 209–218.

[6] Y. LE TRAON, B. BAUDRY, J.-M. JÉZÉQUEL. *Design by Contract to improve Software Vigilance*, in "IEEE Trans. on Software Engineering", vol. 32, n^o 8, August 2006, p. 571–586.

[7] Y. LE TRAON, T. JÉRON, J.-M. JÉZÉQUEL, P. MOREL. *Efficient OO Integration and Regression Testing*, in "IEEE Trans. on Reliability", vol. 49, n^o 1, March 2000, p. 12–25.

[8] C. NEBUT, F. FLEUREY, Y. LE TRAON, J.-M. JÉZÉQUEL. *Automatic Test Generation: A Use Case Driven Approach*, in "IEEE Trans. on Software Engineering", vol. 32, n^o 3, March 2006, p. 140–155.

[9] G. SUNYÉ, A. LE GUENNEC, J.-M. JÉZÉQUEL. *Using UML Action Semantics for Model Execution and Transformation*, in "Information Systems, Elsevier", vol. 27, n^o 6, July 2002, p. 445–457.

[10] T. ZIADI, J.-M. JÉZÉQUEL. *Product Line Engineering with the UML: Deriving Products*, in "Software Product Lines", n^o ISBN: 978-3-540-33252-7, Springer Verlag, 2006, p. 557–586.


## Year Publications

### Doctoral Dissertations and Habilitation Theses


[11] E. BROTTIER. *Acquisition et analyse des exigences pour le développement logiciel : une approche dirigée par les modèles*, Université de Rennes 1, December 2009, http://www.irisa.fr/triskell/publis/2009/Brottier09.pdf, Ph. D. Thesis.

[12] R. DELAMARE. *Analyses Automatiques pour le Test de Programmes Orientés Aspect*, Université de Rennes 1, December 2009, http://www.irisa.fr/triskell/publis/2009/Delamare09c.pdf, Ph. D. Thesis.

[13] J. SIMONIN. *Conception de l'architecture d'un système dirigée par un modèle d'urbanisme fonctionnel*, Université de Rennes 1, January 2009, http://www.irisa.fr/triskell/publis/2009/Simonin09a.pdf, Ph. D. Thesis.

### Articles in International Peer-Reviewed Journal


[14] B. BAUDRY, S. GHOSH, F. FLEUREY, R. FRANCE, Y. LE TRAON, J.-M. MOTTU. *Barriers to Systematic Model Transformation Testing*, in "Communications of the ACM", 2009.

[15] R. BENDRAOU, J.-M. JÉZÉQUELAND, M.-P. GERVAIS, X. BLANC. *A Comparison of Six UML-Based Languages for Software Process Modeling*, in "IEEE Transactions on Software Engineering", 2009, http://www.irisa.fr/triskell/publis/2009/Bendraou09b.pdf NO .

[16] B. COMBEMALE, X. CRÉGUT, P.-L. GAROCHE, X. THIRIOUX. *Essay on Semantics Definition in MDE. An Instrumented Approach for Model Verification*, in "Journal of Software", vol. 4, n^o^ 6, december 2009.

[17] R. FRANCE, J.-M. JÉZÉQUEL. *Editorial for the Special Issue on Aspects and Model-Driven Engineering*, in "Transactions on Aspect-Oriented Software Development (TAOSD)", vol. LNCS 5560, 2009, p. 16-21, http://www.irisa.fr/triskell/publis/2009/France09a.pdf.

[18] N. MOHA, Y.-G. GUÉHÉNEUC, L. DUCHIEN, A.-F. LE MEUR. *DECOR: A Method for the Specification and Detection of Code and Design Smells*, in "IEEE Transactions on Software Engineering", 2009, http://www.irisa.fr/triskell/publis/2009/Moha09d.pdf NO .

[19] N. MOHA, Y.-G. GUÉHÉNEUC, A.-F. LE MEUR, L. DUCHIEN, A. TIBERGHIEN. *From a Domain Analysis to the Specification and Detection of Code and Design Smells*, in "Formal Aspects of Computing, Springer", May 2009, http://www.irisa.fr/triskell/publis/2009/Moha09a.pdf NO .

[20] M. MONPERRUS, B. LONG, J. CHAMPEAU, B. HOELTZENER, G. MARCHALOT, J.-M. JÉZÉQUEL. *Model-driven Architecture of a Maritime Surveillance System Simulator*, in "Systems Engineering Journal", vol. 13, 2009.

[21] B. MORIN, O. BARAIS, J.-M. JÉZÉQUEL, F. FLEUREY, A. SOLBERG. *Models at Runtime to Support Dynamic Adaptation*, in "IEEE Computer", October 2009, p. 46-53, http://www.irisa.fr/triskell/publis/2009/Morin09f.pdf NO .

[22] S. SEN, B. BAUDRY, H. VANGHELUWE. *Towards Domain-specific Model Editors with Automatic Model Completion*, in "Simulation Journal", vol. 3, n^o^ 12, 2009, http://www.irisa.fr/triskell/publis/2009/Sen09c.pdf.


### Articles in National Peer-Reviewed Journal


[23] F. CHAUVEL, O. BARAIS, I. BORNE, J.-M. JÉZÉQUEL. *Un processus à base de modèles pour les systèmes auto-adaptatifs*, in "Revue de l'Electricité et de l'Electronique (REE)", n^o^ 2, February 2009, p. 38–44, http://www.irisa.fr/triskell/publis/2009/Chauvel09a.pdf.


### Articles in Non Peer-Reviewed Journal


[24] A. KOUDRI, J. CHAMPEAU, D. AULAGNIER, D. VOJTISEK. *Processus MOPCOM pour SoC/SoPC*, in "Génie Logiciel - Ingéniérie dirigée par les modèles", sep 2009, http://www.irisa.fr/triskell/publis/2009/Koudri09a.pdf.

[25] D. VOJTISEK. *OpenEmbeDD : La plate-forme et ses outils d'ingénierie de modèles*, in "Génie Logiciel - Ingéniérie dirigée par les modèles", vol. 89, jun 2009, p. 31-37.

### International Peer-Reviewed Conference/Proceedings


[26] R. BENDRAOU, J.-M. JÉZÉQUEL, F. FLEUREY. *Combining Aspect and Model-Driven Engineering Approaches for Software Process Modeling and Simulation*, in "Proceedings of the International Conference on Software Process (ICSP2009), Vancouver, Canada", Springer Lecture Notes in Computer Science (LNCS), may 2009, http://www.irisa.fr/triskell/publis/2009/Bendraou09a.pdf.

[27] R. DELAMARE, B. BAUDRY, S. GHOSH, Y. LE TRAON. *A Test-Driven Approach to Developing Pointcut Descriptors in AspectJ*, in "ICST '09: Proceedings of the 2nd International Conference on Software Testing, Verification, and Validation", April 2009, http://www.irisa.fr/triskell/publis/2009/Delamare09a.pdf.

[28] R. DELAMARE, B. BAUDRY, Y. LE TRAON. *AjMutator: A Tool For The Mutation Analysis Of AspectJ Pointcut Descriptors*, in "Mutation'09: Proceedings of the 4th International Workshop on Mutation Analysis", April 2009, http://www.irisa.fr/triskell/publis/2009/Delamare09b.pdf.

[29] P. DHAUSSY, P.-Y. PILLAIN, S. CREFF, A. RAJI, Y. LE TRAON, B. BAUDRY. *Evaluating Context Descriptions and Property Definition Patterns for Software Formal Validation*, in "ACM/IEEE 12th International Conference on Model Driven Engineering Languages and Systems (MODELS'09)", October 2009, p. 438-452, http://www.irisa.fr/triskell/publis/2009/dhaussy09.pdf.

[30] P. ISTOAN, G. NAIN, G. PERROUIN, J.-M. JÉZÉQUEL. *Dynamic Software Product Lines for Service-Based Systems*, in "9th IEEE International Conference on Computer and Information Technology, Xiamen, CHINA", October 2009, http://www.irisa.fr/triskell/publis/2009/Istoan09a.pdf.

[31] J. KLEIN, K. JÖRG, B. MORIN, J.-M. JÉZÉQUEL. *Aspect Model Unweaving*, in "ACM/IEEE 12th International Conference on Model Driven Engineering Languages and Systems (MODELS'09), Denver, Colorado, USA", Oct 2009, http://www.irisa.fr/triskell/publis/2009/Morin09d.pdf LU CA .

[32] N. MOHA, V. MAHÉ, O. BARAIS, J.-M. JÉZÉQUEL. *Generic Model Refactorings*, in "ACM/IEEE 12th International Conference on Model Driven Engineering Languages and Systems (MODELS'09), Denver, Colorado, USA", Oct 2009, http://www.irisa.fr/triskell/publis/2009/Moha09c.pdf US .

[33] B. MORIN, O. BARAIS, G. NAIN, J.-M. JÉZÉQUEL. *Taming Dynamically Adaptive Systems with Models and Aspects*, in "31st International Conference on Software Engineering (ICSE'09), Vancouver, Canada", May 2009, http://www.irisa.fr/triskell/publis/2009/Morin09a.pdf.

[34] B. MORIN, T. LEDOUX, M. BEN HASSINE, F. CHAUVEL, O. BARAIS, J.-M. JÉZÉQUEL. *Unifying Runtime Adaptation and Design Evolution*, in "IEEE 9th International Conference on Computer and Information Technology (CIT'09), Xiamen, China", Oct 2009, http://www.irisa.fr/triskell/publis/2009/Morin09b.pdf.

[35] B. MORIN, G. NAIN, O. BARAIS, J.-M. JÉZÉQUEL. *Leveraging Models From Design-time to Runtime. A Live Demo*, in "4th International Workshop on Models@Run.Time (at MODELS'09), Denver, Colorado, USA", Oct 2009, http://www.irisa.fr/triskell/publis/2009/Morin09e.pdf.

[36] B. MORIN, G. PERROUIN, P. LAHIRE, O. BARAIS, G. VANWORMHOUDT, J.-M. JÉZÉQUEL. *Weaving Variability into Domain Metamodels*, in "ACM/IEEE 12th International Conference on Model Driven Engineering Languages and Systems (MODELS'09), Denver, Colorado, USA", Oct 2009, http://www.irisa.fr/triskell/publis/2009/Morin09c.pdf LU .

[37] T. MOUELHI, Y. LE TRAON, B. BAUDRY. *Transforming and selecting functional test cases for security policy testing*, in "ICST", April 2009, http://www.irisa.fr/triskell/publis/2009/mouelhi09.pdf.

[38] P.-A. MULLER, F. FONDEMENT, B. BAUDRY. *Modeling Modeling*, in "ACM/IEEE 12th International Conference on Model Driven Engineering Languages and Systems (MODELS'09)", October 2009, p. 2-16, http://www.irisa.fr/triskell/publis/2009/muller09.pdf.

[39] F. MUNOZ, B. BAUDRY, R. DELAMARE, Y. LE TRAON. *Inquiring the usage of aspect-oriented programming: an empirical study*, in "25th IEEE International Conference on Software Maintenance (ICSM'09), Edmonton, Alberta, Canada", Sep-Oct 2009, http://www.irisa.fr/triskell/publis/2009/munoz09a.pdf.

[40] G. PERROUIN, E. BROTTIER, B. BAUDRY, Y. LE TRAON. *Composing Models for Detecting Inconsistencies: A Requirements Engineering Perspective*, in "Proceedings of the International Working Conference on Requirements Engineering: Foundation for Software Quality (REFSQ2009), Amsterdam, Netherlands", Springer Lecture Notes in Computer Science (LNCS), june 2009, http://www.irisa.fr/triskell/publis/2009/Perrouin09a.pdf.

[41] S. SEN, B. BAUDRY, J.-M. MOTTU. *Automatic Model Generation Strategies for Model Transformation Testing*, in "ICMT", 2009.

[42] S. SEN, N. MOHA, B. BAUDRY, J.-M. JÉZÉQUEL. *Meta-model Pruning*, in "ACM/IEEE 12th International Conference on Model Driven Engineering Languages and Systems (MODELS'09), Denver, Colorado, USA", Oct 2009, http://www.irisa.fr/triskell/publis/2009/Sen09b.pdf US .

[43] S. VAUCHER, F. KHOMH, N. MOHA, Y.-G. GUÉHÉNEUC. *Tracking Design Smells: Lessons from a Study of God Classes*, in "16th Working Conference on Reverse Engineering (WCRE 2009), IEEE Computer Society Press (WCRE'09), Lille, France", Oct 2009, http://www.irisa.fr/triskell/publis/2009/Moha09b.pdf FR .

### National Peer-Reviewed Conference/Proceedings

[44] F. CHAUVEL, O. BARAIS, N. PLOUZEAU, I. BORNE, J.-M. JÉZÉQUEL. *Composition et expression qualitative de politiques d'adaptation pour les composants Fractal*, in "Actes des Journées nationales du GDR GPL 2009, Toulouse, France", January 2009, http://www.irisa.fr/triskell/publis/2009/Chauvel09b.pdf.

[45] V. LE GLOAHEC, R. FLEURQUIN, S. SALAH. *Formalisation de bonnes pratiques dans les procédés de développement logiciels*, in "5èmes Journées de l'Ingénierie dirigée par les modèles (IDM 2009), Nancy, France", March 2009, http://www.irisa.fr/triskell/publis/2009/legloahec09a.pdf.

[46] F. MUNOZ, B. BAUDRY. *A Framework for Testing Model Composition Engines*, in "SC '09: Proceedings of the 8th International Conference on Software Composition, Berlin, Heidelberg", Springer-Verlag, 2009, p. 125–141, http://www.irisa.fr/triskell/publis/2009/munoz09b.pdf.

[47] G. NAIN, O. BARAIS, R. FLEURQUIN, J.-M. JÉZÉQUEL. *EntiMid : un middleware aux services de la maison*, in "3ème Conférence Francophone sur les Architectures Logicielles (CAL'O9), Nancy, France", March 2009, http://www.irisa.fr/triskell/publis/2009/Nain09a.pdf.

### Workshops without Proceedings

[48] C. ANDRÉ, M. BELAUNDE, B. BERTHOMIEU, C. BRUNETTE, A. CANALS, H. GARAVEL, S. GRAF, F. LANG, V. MAHÉ, M. NAKHLÉ, R. SCHNEKENBURGER, R. DE SIMONE, J.-P. TALPIN, F. VERNADAT.

*Présentation des résultats du projet OpenEmbeDD*, in "Neptune, Paris France", P. BAZEX, A. CANALS, T. MILLAN (editors), Revue Génie Logiciel - AFCET, 2009, http://hal.archives-ouvertes.fr/inria-00381639/en/.

### Scientific Books (or Scientific Book chapters)

[49] D. AULAGNIER, A. KOUDRI, S. LECOMTE, P. SOULARD, J. CHAMPEAU, J. VIDAL, G. PERROUIN, P. LERAY. *SoC/SoPC development using MDD and MARTE profile*, in "Model Driven Engineering for Distributed Real-time Embedded Systems", Hermes, 2009, http://www.irisa.fr/triskell/publis/2009/Aulagnier09a.pdf.

[50] B. BAUDRY. *Testing Model Transformations: A case for Test Generation from Input Domain Models*, in "Model Driven Engineering for Distributed Real-time Embedded Systems", Hermes, 2009, http://www.irisa.fr/triskell/publis/2009/Baudry09b.pdf.

### Research Reports

[51] F. MUNOZ, B. BAUDRY. *Artificial table testing dynamically adaptive systems*, n° inria-00365874, INRIA Bretagne Atlantique, March 2009, http://hal.inria.fr/inria-00365874/en/, Research report.

### Scientific Popularization

[52] D. VOJTISEK. *Introduction à la construction d'un DSL sous Eclipse*, in "Programmez!", vol. 120, june 2009, p. 70-72.

## References in notes

[53] A. BEUGNARD, J.-M. JÉZÉQUEL, N. PLOUZEAU, D. WATKINS. *Making Components Contract Aware*, in "IEEE Computer", vol. 13, n° 7, July 1999.

[54] G. BOOCH. *Object-Oriented Analysis and Design with Applications*, 2nd, Benjamin Cummings, 1994.

[55] E. GAMMA, R. HELM, R. JOHNSON, J. VLISSIDES. *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison Wesley, 1995.

[56] M. JACKSON. *System Development*, Prentice-Hall International, Series in Computer Science, 1985.

[57] J.-M. JÉZÉQUEL, B. MEYER. *Design by Contract: The Lessons of Ariane*, in "Computer", vol. 30, n° 1, January 1997, p. 129–130.

[58] B. MEYER. *Reusability: The Case for Object-Oriented Design*, in "IEEE SOFTWARE", n° 3, March 1987, p. 50–64.

[59] B. MEYER. *Applying "Design by Contract"*, in "IEEE Computer (Special Issue on Inheritance & Classification)", vol. 25, n° 10, October 1992, p. 40–52.

[60] G. PERROUIN, J. KLEIN, N. GUELFI, J.-M. JÉZÉQUEL. *Reconciling Automation and Flexibility in Product Derivation*, in "12th International Software Product Line Conference (SPLC 2008), Limerick, Ireland", IEEE Computer Society, September 2008, p. 339–348, http://www.irisa.fr/triskell/publis/2008/Perrouin08a.pdf.

[61] A. RASHID, J. ARAÚJO. *Modularisation and composition of aspectual requirements*, in "Proceedings of the 2nd international conference on Aspect-oriented software development", 2003, p. 11–20.

[62] C. SZYPERSKI. *Component Software: Beyond Object-Oriented Programming*, ACM Press and Addison-Wesley, New York, N.Y., 1998.

[63] J. WARMER, A. KLEPPE. *The Object Constraint Language*, Addison-Wesley, 1998.