INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Project-Team Triskell

Model Driven Engineering for Component Based Software

Rennes - Bretagne Atlantique

THEME COM

Activity Report

2008

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]
Yves Le Traon [ENST Bretagne, HdR]
Noël Plouzeau [Assistant Professor Université de Rennes 1]

External Collaborator
Pierre-Alain Muller [Professor Université de Haute Alsace, HdR]
Régis Fleurquin [Université de Bretagne Sud since September 2008]

Technical Staff
Didier Vojtisek [Research engineer Inria]
Franck Chauvel [Université Rennes 1 (project FAROS)]
Cyril Faucher [Inria (project Openembedd)]
Vincent Mahé [Inria (project Openembedd)]
François Tanguy [Inria Associated Engineer until September 2008]
Mahmoud Ben Hassine [Inria (project Galaxy) since November 2008]

PhD Student
Erwan Brottier [CIFRE grant]
Romain Delamare [Brittanny Council grant]
Marouane Himdi [CIFRE grant until October 2007]
Martin Monperus [DGA grant until November 2008]
Brice Morin [INRIA grant since October 2007]
Jean-Marie Mottu [MENRT grant]
Freddy Munoz [INRIA grant since October 2007]
Jacques Simonin [Orange Labs]
Sagar Sen [INRIA grant]
Tejeddine Mouelhi [ENST grant]
Maha Driss [INRIA grant since March 2008]
Grégory Nain [INRIA grant since October 2008]
Muhammad-Ali Memon [SFERE grant since October 2008]
Mickael Clavreul [INRIA grant since October 2008]

Post-Doctoral Fellow
Reda Bendraou [INRIA grant until october 2008]
Julien De Antoni [INRIA (project Speeds) until october 2008]
Gilles Perrouin [INRIA (project MOPCOM and SPEEDS)]
Naouel Moha [INRIA grant since october 2008]

Administrative Assistant
Loïc Lesage [TR Inria]

2. Overall Objectives

2.1. Introduction

Keywords: *Components*, *MDA*, *MDE*, *UML*, *aspects*, *contracts*, *design patterns*, *frameworks*, *meta-models*, *models*, *objects*, *requirements engineering*, *scenarios*, *software product lines*, *test*, *validation.*

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 reliable and efficient design of software product lines using Model Driven Engineering. Triskell is particularly interested in component based, embedded systems and service oriented architectures with quality of service constraints.

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 a distributed application. 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.

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 has released the version 1.2 of its Kermeta software (KERnel META-modelling), that now comes with a compiler making it possible to deliver efficient standalone tools ready to be integrated in software development environments such as Eclipse. Kermeta is heavily used within Triskell for many collaborative projects, but also beyond Triskell by a growing number of both academic and industrial partners worldwide.

- As promised at the last INRIA evaluation seminar, Triskell has started to investigate a more dynamic use of models (an approach dubbed *models at runtime* [31], [29], [41], [39], [23]), and its application to Software Oriented Architectures in the context of Building Automation [22], [40]. Meanwhile we have made several breakthroughs in tackling the difficult problems of testing models transformations [44], [34] and Aspect Oriented Software [21], [38].
- Triskell had 4 papers accepted [27], [31], [33], [36] at MODELS'2008, the main conference in our field (274 papers submitted, 57 accepted, including 9 French papers). 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

Keywords: *Objects*, *UML*, *aspects*, *contracts*, *design patterns*, *models*, *product lines*, *software components*.

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 [57], [55]: 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 [59].

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 [56]. 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 [58]).

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 [60] 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) [54].

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

Keywords: *Embedded Systems*, *SOA*, *UML*, *distributed systems*, *software engineering*, *telecommunication*, *test*.

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

Keywords: *MDA*, *MOF*, *UML*, *model transformation*.

Participants: Olivier Barais, Franck Chauvel, Cyril Faucher, Jean-Marc Jézéquel, Jean-Marie Mottu, Pierre-Alain Muller, Fraçois Tanguy, 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.

5.2. Sintaks : Textual syntaxes for models

Keywords: *MDA*, *MOF*, *UML*, *model transformation*, *syntax*.

Participants: Erwan Brottier, Pierre-Alain Muller, 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

Keywords: *AOM*, *MDA*, *MOF*, *model composition*.

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 [42]. Kompose has been developed in collaboration with CSU in the context of the MATT équipe associée.

6. New Results

6.1. Contract-based and Aspect Oriented Design

6.1.1. Design of QoS-aware Service Oriented Architectures

Participants: Franck Chauvel, Olivier Barais, Noël Plouzeau, Jacques Simonin, Jean-Marc Jézéquel.

We investigated several aspects of Service Oriented Architectures, from methodological issues of handling their developemnt [45], to tool support and QoS evaluation.

In the context of the Faros French national project, which aims at building a whole design process for QoS-aware Service Oriented Architectures, we designed a transformation framework to handle QoS contract models in two stages: a first stage transform a business model into a central model, which is business and target platform independent.A second stage transform this central model into a platform model, which is platform dependant. The technical delivrable on business to central model transformation is available on the Faros web site.

Further, in highly dynamic environments, software systems requires a capacity of self-adaptation to fit the environment and the user needs evolution, which increases the software architecture complexity. Despite most current execution platforms include some facilities for handling dynamic adaptation, current design methodologies do not address this issue. One of the requirement for such a design process is to describe adaptation policies in a composable and qualitative fashion in order to cope with complexity. This work introduces an approach for describing adaptation policies in a qualitative way while keeping the compositionality of adaptation policies. This work uses the Fuzzy Logic Theory as a fundation for the adaptation policies language. This approach has been tooled as an extension of the Fractal component model [49], [20], [19], [11].

6.1.2. Entimid: a Model Based SOA for Building Automation

Participants: Grégory Nain, Olivier Barais, Jean-Marc Jézéquel.

In the personal or corporate spheres, the home/office of tomorrow is soon to be the home/office of today, with a plethora of networked devices embedded in appliances, such as mobile phones, televisions, thermostats, and lamps, making it possible to automate and remotely control many basic household functions with a high degree of accuracy. In this domain, technological standardization is still in its infancy, or remains fragmented. The different functionalities of the various appliances, as well as market factors, imply that the devices that control them communicate via a multitude of different protocols (KNX, LonWorks, InOne). Building a high level middleware to support all the appliances seems to be a reasonable approach. However, market factors has shown that the emergence of a unique and universal middleware is a dream. To solve this issue, we have built a new generation of schizophrenic middleware in which service access can be generated from an abstract services description. EntiMid, our implementation of schizophrenic middleware, supports various services access models (several personalities): SOAP (Simple Object Access Protocol), UPnP and DPWS (Device Profile for WebServices). These personalities are generated using a Model Driven Engineering approach and this middleware will be deployed in the context of new services definition at the Rennes city level [40].

6.1.3. Managing Software Product Line

Participants: Gilles Perrouin, Jean-Marc Jézéquel.

Product derivation, i.e. reusing core assets to build products, did not receive sufficient attention from the product-line community, yielding a frustrating situation. On the one hand, automated product derivation approaches are inflexible; they do not allow products meeting unforeseen, customer-specific, requirements. On the other hand, approaches that consider this issue do not provide adequate methodological guidelines nor automated support. We proposed an integrated product derivation approach reconciling the two views to offer both flexibility and automation. First, we perform a pre-configuration of the product by selecting desired features in a generic feature model and automatically composing their related product-line core assets. Then, we adapt the pre-configured product to its customer-specific requirements via derivation primitives combined by product engineers and controlled by constraints that flexibly set product line boundaries. Our process is supported by the Kermeta metamodeling environment [42], [41], [16].

6.2. Model-Based Testing

6.2.1. Automatic model synthesis

Participants: Sagar Sen, Benoit Baudry.

The input domain of a complex software is increasingly shifting from primitive data types like integers, booleans, reals, and strings and data structures such as lists, trees, and graphs to models. Models are complex graph structures specified and constrained by a meta-model. Exploring the space of models in the modelling domain, specified by such a meta-model, to effectively attain certain objectives is the theme of our research. In [44], we present a tool, Cartier, to transform a meta-model to a first-order relational logic language with quantifiers, Alloy, in order to perform constraint satisfaction and model synthesis. We apply the building of effective modelling domains to generate models for model transformation testing.

6.2.2. Validating Aspect Oriented Programs

Participants: Romain Delamare, Freddy Munoz, Benoit Baudry, Olivier Barais, Yves Le Traon, Jean-Marc Jézéquel.

The maintenance of aspect-oriented programs is challenging for several reasons because aspects crosscut several program modules making it difficult to reason about their interactions with the program. If test cases exist, the introduction of aspects and / or the evolution of the system may make them fail. In order to address these problems we propose different approaches. To address interaction issues, we propose ABIS [], [38] an aspect-base program interaction specification framework. To address the impact on test cases we propose Vidock [21], a test case impact analysis for aspect-oriented programs. Vidock identifies the test cases which behavior is impacted by aspect weaving.

6.2.3. Model-based security testing

Participants: Tejedinne Mouelhi, Benoit Baudry, Yves Le Traon.

Our work on security testing consisted in three studies related to test generation and the implication of security test cases to evaluate the adaptability of a system to a change in sthe security policy.

We proposed a meta-model for access control policies [37], [36], [35] which is integrated to a framework for specifying, deploying and testing access control policies. We specify the security policy using our metamodel, then we instantiate the policy in a specific language (RBAC,OrBAC, MAC or DAC) using model transformation. The policy is deployed to the XACML architecture and the security calls are weaved in the business logic using AOP. A tool for mutation analysis injects faults in the system using fault mdoels defined at the generic level and is used to qualify security test cases.

We studied combinatorial testing [43] to generate test data for security policies. We compared pair-wise testing to several random test generations. In [25] we studied how to use security tests to detect hidden security mechanisms in legacy systems. If access control policy decision points are not neatly separated from the business logic of a system, the evolution of a security policy likely leads to the necessity of changing the system's code base. We analyzed the notion of flexibility which is related to the presence of hidden and implicit security mechanisms in the business logic.

6.2.4. P2P testing framework

Participant: Yves Le Traon.

Peer-to-peer (P2P) is becoming a key technology for software development, but still lacks integrated solutions to validate the final software. Testing P2P systems is difficult because of the high numbers of peers which can be volatile. In this collaboration with Eduardo Cunha de Almeida, Gerson Sunyé, and Patrick Valduriez of the LINA lab [46], [47], we proposed an integrated solution for testing large-scale P2P systems. The solution is based on a framework with two original aspects: (i) the individual control of peers volatility and (ii) a distributed testing architecture to cope with large numbers of peers. The framework has been validated through implementation and experimentation on two open-source P2P systems. Through experimentation, we analyze the behavior of both systems on different conditions of volatility and show how the framework is able to detect problems.

6.3. Model-Driven Engineering

6.3.1. Model Measurement

Participants: Martin Monperrus, Naouel Moha, Jean-Marc Jézéquel.

Companies using domain specific languages in a model-driven development process need to measure their models. However, developing and maintaining a measurement software for each domain specific modeling language is costly. Our contribution is a model-driven measurement approach [52], [27]. This measurement approach is model-driven from two viewpoints: 1) it measures models of a model-driven development process; 2) it uses models as unique and consistent metric specifications, w.r.t a metric specification metamodel. This declarative specification of metrics is then used to generate a fully fledged implementation. The benefit derived from using model-driven technologies has been evaluated by several real-size case studies [26], [28], [12]. They indicate that this approach seems to reduce the domain-specific measurement software development cost.

6.3.2. Executable Software Process Modeling

Participants: Reda Bendraou, Jean-Marc Jézéquel.

One of the main objectives of the Model-Driven Engineering vision is to increase software productivity through the extensive use of models since earliest software development phases. The challenge targeted by this initiative is to use models not only for documentation purposes but also for production aims. In the area of software process modeling, software process modeling languages have not yet reached the level required for the specification of executable models. Executable software process models can help in improving coordination between development teams, in automating iterative and no-interactive tasks and in managing the different tools and artifacts used during the software construction. At this aim, we have proposed UML4SPM, a model-driven and executable language for software process modeling, and we have shown how it was implemented using Kermeta [48].

6.3.3. Model transformation testing

Participants: Jean-Marie Mottu, Sagar Sen, Benoit Baudry, Yves Le Traon.

Model transformations can automate specific tasks in the software development. In [13], we contribute to model transformation testing. Testing such model transformations for correctness presents some new challenges. First, we adapt mutation analysis to model transformations in order to qualify fault detecting effectiveness of a set of test models by considering model transformation specific faults. Second, in [34], [50] we propose a set of functions to express test oracles for detecting faults in a transformation. We evaluate them regarding the complexity and reuse of model transformations. Finally, we integrate our techniques in tools that are used to develop reliable model transformation components and to assist in further model transformation testing studies. In particular, we compare different strategies for automatic test model synthesis [44].

7. Contracts and Grants with Industry

7.1. SPEEDS (IST)

Keywords: *COTS*, *SysML*, *UML*, *embedded systems*, *methods*, *system engineering*.

Participants: Jean-Marc Jézéquel, Julien Deantoni, 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 actively participates to the definition of the UML profile for HRC. 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)

Keywords: *AOSD*, *MDE*, *adaptative systems*.

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 catpured 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 develped 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 mechnanisms 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 adpating 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)

Keywords: *Services-Oriented Architecture*.

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.

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)

Keywords: *Aspect Oriented Design*.

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: 2004-2008

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. Artist2 (Network of Excellence)

Keywords: *Real-Time Component Models*.

Participants: Jean-Marc Jézéquel, Noël Plouzeau, Pierre-Alain Muller, Benoit Baudry, Didier Vojtisek.

The strategic objective of the ARTIST2 Network of Excellence is to strengthen European research in Embedded Systems Design, and promote the emergence of this new multi-disciplinary area. Artist2 gathers together the best European teams from the composing disciplines, and will work to forge a scientific community. Integration will be achieved around a Joint Programme of Activities, aiming to create critical mass from the selected European teams.

The ARTIST2 Network of Excellence on Embedded Systems Design is implementing an international and interdisciplinary fusion of effort to create a unique European virtual centre of excellence on Embedded Systems Design. This interdisciplinary effort in research is mandatory to establish Embedded Systems Design as a discipline, combining competencies from electrical engineering, computer science, applied mathematics, and control theory. The ambition is to compete on the same level as equivalent centres in the USA (Berkeley, Stanford, MIT, Carnegie Mellon), for both the production and transfer of knowledge and competencies, and for the impact on industrial innovation.

ARTIST2 addresses the full range of challenges related to Embedded Systems Design, covering all aspects, ranging from theory through to applications. In this way, ARTIST2 is perfectly in line with the IST priority on embedded systems, and in particular with the focus area called "system design".

The Triskell team is taking part in two Artist2 clusters: the *Real Time Components* cluster (led by Albert Benveniste, Irisa, and Bengt Jonsson, at Uppssala university, Sweden) and the Adaptive Real Time Middleware (led by Giorgo Buttazzo, Italy).

The current cooperation topics within the Real TimeComponents cluster are the use of various formalisms for timed behaviour descriptions, the definition of an architecture for interconnecting simulation and verification platforms for these behaviours. The Triskell team has designed a process and a tool chain to support specification, validation and monitoring of time issues in software components. This tool chain was implemented by integrating and extending existing tools from partners of the RTC cluster.

Within the Adaptive Real Time cluster, Triskell is participating in the common definition of quality of service dictionary, in the context of middleware runtimes. The Triskell project has also proposed a new metamodel for expressing quality of service properties of software components. The proposal is being compared and evaluated with respect to other metamodels proposed by Artist partners (including the Marte profile for UML proposed at OMG), in order to build a common Artist2 metamodel for quality of service.

Project duration: 2004-2008

Triskell budget share: 50 keuros

Project Coordinator: Verimag

Participants: see http://www.artist-embedded.org/artist/

7.6. Mopcom Hard (RNTL)

Keywords: *MARTE*, *MDE*, *RT-E*, *UML*, *reconfigurability*, *system on chip*.

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 2008, 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 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

7.7. Mopcom Ingénierie (Competitivity Cluster I&R)

Keywords: *MDE*, *Model based testing*, *UML*.

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 2008, Triskell participated to the case study definition and proposed a model-driven methodology for this domain.

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

7.8. Orange Labs

Keywords: *MDE*, *migration*, *model transformation*, *regression testing*.

Participants: Yves Le Traon, 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. More specifically, in this area, we are working on measuring alignment between Buiseness and IT levels of the IS for Telecom Service Development. In this context, Jean-Marc Jézéquel acts as Ph.D advisor for Jacques Simonin and Mariano Belaunde, both being senior Orange Labs engineers.

Project duration: 2006-2009

Triskell budget share: 10 keuros

7.9. OpenDevFactory

Keywords: *MDE*, *UML*, *metamodel*, *requirements engineering*, *traceability*.

Participants: Benoit Baudry, David Touzet, Erwan Brottier, Didier Vojtisek.

OpenDevFactory is a sub project of the project Usine Logicielle (labelled by the System@tic Competitivity Cluster). Its objective is to supply a standard platform for integrating technological developments for modelling software tools. This sub project produces technological components on top of which domain tools (automobile, security, telecommunication, aeronautical) can be derived at a lesser effort. That platform is built as an interoperable federation of tools which limited parts could be deployed to make specialised IDEs meeting the particular needs of different kinds of users. The technological bricks are organized as follows:

- Technological infrastructure bricks for MDE such as providing support for model transformation, behaviour modelling as well as orchestration of engineering activities.
- Domain extension bricks supporting fault tolerance modelling, Real time embedded systems modelling, platforms modelling, requirements modelling or UML simulations.
- Integration technologies of MDE design environments with other engineering environments such as design environments for design of automatisms or critical embedded software.

The integration structure of OpenDevFactory is build on top of the Eclipse framework. In this context, Triskell has developped an Eclipse plugin providing a requirements engineering integrated environment. This environment includes the following features:

- Requirements specification by means of a controled natural language (requirement description language).
- Definition of a requirements metamodel, and automated transformation from textual to model-based specifications.
- Definition of a usecase based metamodel encoding the dynamic semantics of the defined requirements.

- Parametrized interpretation (using interpretation patterns) of a requirements model in order to build its corresponding usecase model.
- Simultaion facilities enabled over the obtained usecase model.

Project duration: 2005-2008

Triskell budget share: 75 keuros

Project Coordinator: Thales R&T

Participants: CEA, CS, Dassault Aviation, EADS, EDF, Esterel Technologies, Hispano Suiza, IFP, INRIA, LIP6, LRI, MBDA,Ecole Polytechnique, Softeam, Supelec, Thales, Trialog

7.10. DOMINO (RNTL)

Keywords: *domain specific languages*, *model transformation*, *model-driven engineering*, *reliability*, *validation*, *verification*.

Participants: Benoit Baudry, Jean-Marc Jézéquel, Jean-Marie Mottu, Yves Le Traon, 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.

Project duration: 2006-2008

Triskell budget share: 79 keuros

Project Coordinator: IRIT

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

7.11. OpenEmbeDD (RNTL)

Keywords: *MARTE*, *MDE*, *RT/E requirements engineering*, *RT/E system*, *UML*, *formal proof*, *model transformation*, *model-checking*.

Participants: Jean-Marc Jézéquel, Didier Vojtisek, Cyril Faucher, Vincent Mahé, François Tanguy.

OpenEmbeDD is a RNTL project which buid 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 intregrates 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

7.12. Faros (RNTL)

Keywords: *MDE*, *Web services*, *model transformation*, *quality of service*.

Participants: Noel Plouzeau, Jean-Marc Jézéquel, Franck Chauvel.

Faros is a project supported by the RNTL program. The Faros project has started in march, 2006. This project will last 36 months. The general objective of the project is 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 will be able to accept as input domain specific, platform independant components. The tool will 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 will use the 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 will be 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)

7.13. TopCaseD (Aerospace Valley Competitivity Cluster)

Keywords: *MDE*, *RT/E requirements engineering*, *RT/E system*, *model checking*, *model transformation*.

Participants: Jean-Marc Jézéquel, Didier Vojtisek, Cyril Faucher, François Tanguy.

TopCaseD is a project of the Aerospace Valley Competitivity Cluster aiming at developing an open source CASE environment for critical applications and systems development. 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, TopCaseD relies on the Eclipse Modelling Project platform (EMF, GEF, GMF, OCL, UML2, ...) and on many available tools such as the AMMA tools, MDDi model bus, Kermeta executable models ...

The participation of Triskell into the TopCaseD project aims to the integration of Kermeta as the simulation engine of Topcased. Triskell is also participating to the development of the code generators to generate graphical editors.

Project duration: 2006-2009

Project Coordinator: Airbus

Participants: Airbus, CNES, EADS-Astrium, Rockwell Collins, Siemens VDO Automotive, Thales Avionics, TurboMeca, AdaCore, AnyWare Technologies, ATOS Origin, C-S, Ellidiss Technologies, Micoin Consulting, SodiFrance, Sogeti-HiTech, SopraGroup, Tectosages, TNI-Software, ENSIETA, ESEO, FERIAIRIT/ LAAS/ ONERA, INRIA (ATLAS/EXPRESSO/TRISKELL), MIPS, SEI, UFSC, ENSEEIHT, INSAT, UPS

8. Other Grants and Activities

8.1. International working groups

8.1.1. 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

8.1.2. 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.

8.1.3. Standardization at OMG

In 2008, Triskell project participates to normalization actions at OMG (http://www.omg.org/). It is involved in the MARTE FTF (Finalization Task Force), in the Concrete Syntax for a UML Action Language RFP (Request For Proposal) and in the Analysis and Design group which promotes standard modelling techniques including UML and MOF.

8.1.4. 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 on Model-driven engineering: Aspects, Transformations and Test[1]. We have collaborated on model composition for aspect-oriented modelling, model transformation and testing aspect-oriented programs. In this context, Benoit Baudry visited CSU from December 2007 to September 2008 through the INRIA sabbatic program, Romain Delamare and Freddy Muñoz visited CSU in March 2008, Jean-Marc Jézéquel visited CSU in July 2008, Robert France and Geri Georg visited IRISA in December 2008.

  In October 2008 we submitted a proposal to renew the équipe associée.
- 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 2008, we have consolidated the model-snippet MDK used in Kermeta.
- 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

[1](see http://www.irisa.fr/triskell/matt for details)

9.1.1.2. Yves Le Traon

is a member of the editorial board of the "L'Objet" journal.

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":

- Diego Alonso Cáceres, April 2008, Univ. Cartagena (referee);
- Slavisa Markovic, May 2008, EPFL (referee);
- Francois Mekerke, July 2008, université de Rennes (president);
- Naouel Moha, August 2008, université de Montréal (referee);
- Chantal Kabore, Sept. 2008, université de Rennes (member);
- Franck Chauvel, Sept 2008, université de Bretagne Sud (co-adviser);
- Martin Monperrus, October 2008, université de Rennes (adviser);
- Jean-Marie Mottu, November 2008, université de Rennes (member);
- Carlos Noguera, November 2008, université de Lille (president);
- Xavier le Guillou, November 2008, université de Rennes (president);
- Greg O'Keefe, November 2008, Australian National Univ. (referee);
- Romain Robbes, December 2008, Univ. Lugano (referee);

9.1.2.2. Benoit Baudry

was in the examination committee of the following PhD thesis

- Jean-Marie Mottu, November 2008, université de Rennes (adviser);

9.1.2.3. Olivier Barais

was in the examination committee of the following PhD thesis

- Jérémy Dubus, October 2008, université de Lille (adviser);

9.1.2.4. Yves Le Traon

was in the examination committee of the following PhD thesis:

- Youssef Serristou, December 2008, Institut National Polythechnique de Grenoble (INPG)
- Christophe Grandpierre, July 2008, Univ. de Franche Comté
- Jean-Marie Mottu, November 2008, université de Rennes (adviser);

9.1.3. Conferences

9.1.3.1. Jean-Marc Jézéquel

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

- ICSE 2008 The 30th International Conference on Software Engineering, Leipzig, Germany, 10 - 18 May 2008
- CBSE 2008 The 11th International Symposium on Component-Based Software Engineering, Karlsruhe, Germany, October 14th-17th, 2008
- MODELS 2008 The 11th International Conference on Model Driven Engineering Languages and Systems Toulouse, France 28 september - 3 october 2008
- ERTS 2008 4th European Congress Embedded Real Time Software, Toulouse, France January 30, 31, February 1, 2008

- VaMoS 2008 Second International Workshop on Variability Modelling of Software-intensive Systems Essen, Germany, January 16-18, 2008
- QoSA 2008 4th International Conference on the Quality of Software Architectures, University of Karlsruhe (TH), Germany October 14-17, 2008
- MOMPES 2008 5th International Workshop on Model-based Methodologies for Pervasive and Embedded Software, Budapest, Hungary, April 5, 2008
- AOM AOSD'08 12th Int'l Workshop on Aspect-Oriented Modeling, Brussels, Belgium, April 1, 2008

9.1.3.2. Yves Le Traon

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

- The 19th IEEE International Symposium on Software Reliability Engineering (ISSRE 2008) November 2008 - Denver, USA.
- IEEE ICST'08 International Conference on Software Testing Verification and Validation, Lillehamer, Norway, April 2008
- 5th int. workshop on Model design and Validation (MoDeVVa 2008)
- 1st IEEE Int. Workshop on Security Testing(SecTest 2008), collocated with ICST 2008
- Workshop on Empirical Studies of Model-Driven Engineering ESMDE 08, collocated with MODELS 2008
- Modeling Security Workshop (Modsec 2008), collocated with MODELS '08.
- Intern. Symposium on Quality Engineering for Embedded Systems (QEES'08), collocated with ECMDA 08.

9.1.3.3. Benoit Baudry

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

- MODELS 2008 The 11th International Conference on Model Driven Engineering Languages and Systems Toulouse, France 28 september - 3 october 2008
- IEEE ICST'08 International Conference on Software Testing Verification and Validation, Lillehamer, Norway, April 2008
- SECTEST'08 workshop on security testing at ICST'08
- 5th int. workshop on Model design and Validation (MoDeVVa 2008) at ICST'08
- A-MOST'08 workshop at ICST'08
- L'Objet special issue on verification of component-based systems

O. Barais has been a member of the programme committee of the following workshops:

- 4èmes Journées sur l'ingénierie Dirigée par les Modèles (IDM'08), Toulouse, France, March 2009.
- Model-Driven Web Engineering workshop in conjonction with Models'08.
- 2nd International Workshop on Domain-Specific Program Development (DSPD) in association with GPCE'2008

9.1.4. Workshops, Tutorials and Keynotes

J.-M. Jézéquel gave invited talks at the Colorado State University, the Florida International University, Telecom Bretagne, and gave keynotes at the Product-Line Conference in Nantes, and at the MDE seminar at EDF R&D. He also gave a tutorial on Model Driven Language Engineering with Kermeta at ICSE'08, MODELS'08 and EJCP'08.

B. Baudry was co-organizer with Alexander Pretschner, Alain Faivre and Sudipto Ghosh of the 5th MoDeVa workshop in conjunction with ICST'08.

B. Baudry was workshop chair for ICST'08.

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 and Yves Le Traon teaching about 50h, for a grand total of about 700 hours, including several courses at ENSTB and INSA Rennes.

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

9.3. Miscellaneous

- 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.
- 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.
- Yves Le Traon is on the steering committee of the IEEE International Conference on Software Testing, Verification and Validation (ICST). He has been Fast abstract chair of ISSRE 2007. He is creator with Alexander Pretschner (ETH) of the 1st IEEE Int. Workshop on Security Testing (SecTest 2008).

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° 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° 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° 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° 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° 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° 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° 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° 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° ISBN: 978-3-540-33252-7, Springer Verlag, 2006, p. 557–586.

Year Publications

Doctoral Dissertations and Habilitation Theses

[11] F. CHAUVEL. *Méthodes et outils pour la conception de systèmes logiciels auto-adaptatifs*, Ph. D. Thesis, Université de Bretagne Sud, September 2008, http://www.irisa.fr/triskell/publis/2008/Chauvel08d.pdf.

[12] M. MONPERRUS. *La mesure des modèles par les modèles : une approche générative*, Ph. D. Thesis, Université de Rennes 1, October 2008, http://www.monperrus.net/martin/la-mesure-des-modeles-par-les-modeles-une-approche-generative.pdf.

[13] J.-M. MOTTU. *Oracles et qualification du test de transformations de modèles*, Ph. D. Thesis, Université de Rennes 1, November 2008, http://www.irisa.fr/triskell/publis/2008/Mottu08c.pdf.

Articles in International Peer-Reviewed Journal

[14] J.-M. JÉZÉQUEL. *Model Driven Design and Aspect Weaving*, in "Journal of Software and Systems Modeling (SoSyM)", vol. 7, n° 2, may 2008, p. 209–218, http://www.irisa.fr/triskell/publis/2008/Jezequel08a.pdf.

[15] P.-A. MULLER, F. FONDEMENT, F. FLEUREY, M. HASSENFORDER, R. SCHNEKENBURGER, S. GÉRARD, J.-M. JÉZÉQUEL. *Model Driven analysis and synthesis of textual concrete syntax*, in "Journal of Software and Systems Modeling (SoSyM)", vol. 7, n° 4, oct 2008, p. 423–442, http://www.irisa.fr/triskell/publis/2008/Muller08a.pdf.

Articles in National Peer-Reviewed Journal

[16] J.-M. JÉZÉQUEL, G. PERROUIN. *Vers des Lignes de Produits Flexibles*, in "RSTI-L'Objet", vol. 14, n° 3, oct 2008, p. 33–45, http://www.irisa.fr/triskell/publis/2008/Jezequel08b.pdf.

International Peer-Reviewed Conference/Proceedings

[17] D. ALONSO, C. VICENTE-CHICOTE, O. BARAIS. *V3Studio: A Component-Based Architecture Modeling Language*, in "15th IEEE International Conference on Engineering of Computer-Based Systems (ECBS'O8), Belfast, Northern Ireland", April 2008, http://www.irisa.fr/triskell/publis/2008/Barais08b.pdf.

[18] O. BARAIS, J. KLEIN, B. BAUDRY, A. JACKSON, S. CLARKE. *Composing Multi-View Aspect Models*, in "7th IEEE International Conference on Composition-Based Software Systems (ICCBSS), Madrid, Spain", February 2008, http://www.irisa.fr/triskell/publis/2008/Barais08a.pdf.

[19] F. CHAUVEL, O. BARAIS, I. BORNE, J.-M. JÉZÉQUEL. *Composition of Qualitative Adaptation Policies*, in "23rd IEEE/ACM International Conference on Automated Software Engineering - ASE'08, L'Aquila, Italy", sep 2008.

[20] F. CHAUVEL, I. BORNE, J.-M. JÉZÉQUEL, O. BARAIS. *A Model-Driven Process for Self-Adaptive Software*, in "4th European Congress ERTS Embedded Real-Time Software, Toulouse, France", January 2008, http://www.irisa.fr/triskell/publis/2008/Chauvel08b.pdf.

[21] R. DELAMARE, B. BAUDRY, Y. LE TRAON. *Regression Test Selection when Evolving Software with Aspects*, in "Proceedings of LATE workshop in conjunction with AOSD'08, Brussels, Belgium", April 2008, http://www.irisa.fr/triskell/publis/2008/Delamare08.pdf.

[22] M. DRISS, Y. JAMOUSSI, J.-M. JÉZÉQUEL, H. H. B. GHÉZALA. *A Discrete-Events Simulation Approach for Evaluation of Service-Based Applications*, in "In Proceedings of the 6th IEEE European Conference on Web Services, Dublin, Ireland", November 2008, http://www.irisa.fr/triskell/publis/2008/Driss08a.pdf.

[23] F. FLEUREY, V. DELHEN, N. BENCOMO, B. MORIN, J.-M. JÉZÉQUEL. *Modeling and Validating Dynamic Adaptation*, in "Proceedings of the 3rd International Workshop on Models@Runtime, at MoDELS'08, Toulouse, France", oct 2008, http://www.irisa.fr/triskell/publis/2008/2008.pdf.

[24] A. KOUDRI, D. AULAGNIER, D. VOJTISEK, P. SOULARD, C. MOY, J. CHAMPEAU, J. VIDAL, J.-C. LE LANN. *Using MARTE in a Co-Design Methodology*, in "Modeling and Analysis of Real-Time and Embedded Systems with the MARTE UML profile workshop co-located with DATE'08, Munich, Germany", mar 2008, http://www.irisa.fr/triskell/publis/2008/Koudri08a.pdf.

[25] Y. LE TRAON, T. MOUELHI, A. PRETSCHNER, B. BAUDRY. *Test-Driven Assessment of Access Control in Legacy Applications*, in "ICST 2008 : First IEEE International Conference on Software, Testing, Verification and Validation (ICST), April 9-11, Lillehammer, Norway", RSM - Dépt. Réseaux, Sécurité et Multimédia (Institut TELECOM ; TELECOM Bretagne), ETH - Eidgenössische Technische Hochschule Zürich (ETH Zurich), IRISA - Institut de Recherche en Informatique et Systèmes aléatoires (INRIA), 2008.

[26] M. MONPERRUS, F. JAOZAFY, G. MARCHALOT, J. CHAMPEAU, B. HOELTZENER, J.-M. JÉZÉQUEL. *Model-driven Simulation of a Maritime Surveillance System*, in "4th European Conference on Model Driven Architecture Foundations and Applications (ECMDA'2008), Berlin, Germany", June 2008, http://www.irisa.fr/triskell/publis/2008/Monperrus08a.pdf.

[27] M. MONPERRUS, J.-M. JÉZÉQUEL, J. CHAMPEAU, B. HOELTZENER. *A Model-driven Measurement Approach*, in "In Proceedings of ACM/IEEE 11th International Conference on Model Driven Engineering Languages and Systems (MoDELS 08), Toulouse, France", October 2008, http://www.irisa.fr/triskell/publis/2008/Monperrus08d.pdf.

[28] M. MONPERRUS, J.-M. JÉZÉQUEL, J. CHAMPEAU, B. HOELTZENER. *Model-driven Engineering Metrics for Real Time Systems*, in "4th European Congress ERTS Embedded Real-Time Software, Toulouse, France", January 2008, http://www.irisa.fr/triskell/publis/2008/Monperrus08b.pdf.

[29] B. MORIN, O. BARAIS, J.-M. JÉZÉQUEL. *K@RT: An Aspect-Oriented and Model-Oriented Framework for Dynamic Software Product Lines*, in "Proceedings of the 3rd International Workshop on Models@Runtime, at MoDELS'08, Toulouse, France", oct 2008, http://www.irisa.fr/triskell/publis/2008/Morin08e.pdf.

[30] B. MORIN, O. BARAIS, J.-M. JÉZÉQUEL. *Weaving Aspect Configurations for Managing System Variability*, in "2nd International Workshop on Variability Modelling of Software-intensive Systems, Essen, Germany", January 2008, http://www.irisa.fr/triskell/publis/2008/Morin08a.pdf.

[31] B. MORIN, F. FLEUREY, N. BENCOMO, J.-M. JÉZÉQUEL, A. SOLBERG, V. DEHLEN, G. BLAIR. *An Aspect-Oriented and Model-Driven Approach for Managing Dynamic Variability*, in "In Proceedings of ACM/IEEE 11th International Conference on Model Driven Engineering Languages and Systems (MoDELS 08), Toulouse, France", October 2008, http://www.irisa.fr/triskell/publis/2008/Morin08d.pdf.

[32] B. MORIN, J. KLEIN, O. BARAIS, J.-M. JÉZÉQUEL. *A Generic Weaver for Supporting Product Lines*, in "International Workshop on Early Aspects at ICSE'08, Leipzig, Germany", May 2008, http://www.irisa.fr/triskell/publis/2008/Morin08b.pdf.

[33] B. MORIN, G. VANWORMHOUDT, P. LAHIRE, A. GAIGNARD, O. BARAIS, J.-M. JÉZÉQUEL. *Managing Variability Complexity in Aspect-Oriented Modeling*, in "In Proceedings of ACM/IEEE 11th International Conference on Model Driven Engineering Languages and Systems (MoDELS 08), Toulouse, France", October 2008, http://www.irisa.fr/triskell/publis/2008/Morin08c.pdf.

[34] J.-M. MOTTU, B. BAUDRY, Y. LE TRAON. *Model transformation testing : oracle issue*, in "MoDeVVa workshop colocated with ICST'08, Lillehammer, Norway", April 2008, http://www.irisa.fr/triskell/publis/2008/Mottu08a.pdf.

[35] T. MOUELHI, B. BAUDRY, F. FLEUREY. *A Generic Metamodel For Security Policies Mutation*, in "SecTest 08: 1st International ICST workshop on Security Testing, April 9, Lillehammer, Norway", RSM - Dépt. Réseaux, Sécurité et Multimédia (Institut TELECOM ; TELECOM Bretagne), IRISA - Institut de Recherche en Informatique et Systèmes aléatoires (INRIA), SINTEF - The Foundation for Scientific and Industrial Research (SINTEF), 2008.

[36] T. MOUELHI, F. FLEUREY, B. BAUDRY, Y. LE TRAON. *A Model-Based Framework for Security Policy Specification, Deployment and Testing*, in "In Proceedings of ACM/IEEE 11th International Conference on Model Driven Engineering Languages and Systems (MoDELS 08), Toulouse, France", October 2008, http://www.irisa.fr/triskell/publis/2008/Mouelhi08a.pdf.

[37] T. MOUELHI, F. FLEUREY, B. BAUDRY, Y. LE TRAON. *Mutating DAC And MAC Security Policies: A Generic Metamodel Based Approach*, in "Modeling Security Workshop In Association with MODELS '08, 28th September, Toulouse, France", RSM - Dépt. Réseaux, Sécurité et Multimédia (Institut TELECOM ; TELECOM Bretagne), IRISA - Institut de Recherche en Informatique et Systèmes aléatoires (INRIA), SINTEF - The Foundation for Scientific and Industrial Research (SINTEF), 2008.

[38] F. MUNOZ, B. BAUDRY, O. BARAIS. *Improving Maintenance in AOP Through an Interaction Specification Framework*, in "ICSM08, 24th International conference on Software Maintentance, Beijing, China", IEEE Computer Society Press, September 2008, http://www.irisa.fr/triskell/publis/2008/munoz2008b.pdf.

[39] F. MUNOZ, B. BAUDRY. *Validation challenges in model composition: The case of adaptive systems*, in "In Proceedings of ChaMDE 2000 - Workshop on Challenges in Model Driven Engineering in conjounction with MODELS'08, Toulouse, France", September 2008, http://www.irisa.fr/triskell/publis/2008/munoz2008c.pdf.

[40] G. NAIN, E. DAUBERT, O. BARAIS, J.-M. JÉZÉQUEL. *Using MDE to Build a Schizofrenic Middleware for Home/Building Automation*, in "In ServiceWave'08: Networked European Software & Services Initiative (NESSI) Conference, Madrid, Spain", December 2008, http://www.irisa.fr/triskell/publis/2008/Nain08a.pdf.

[41] G. PERROUIN, F. CHAUVEL, J. DEANTONI, J.-M. JÉZÉQUEL. *Modeling the Variability Space of Self-Adaptive Applications*, in "2nd Dynamic Software Product Lines Workshop (SPLC 2008, Volume 2), Limerick, Ireland", S. THIEL, K. POHL (editors), IEEE Computer Society, September 2008, p. 15–22, http://www.irisa.fr/triskell/publis/2008/Perrouin08b.pdf.

[42] 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.

[43] A. PRETSCHNER, T. MOUELHI, Y. LE TRAON. *Model-Based Tests for Access Control Policies*, in "ICST 2008 : First IEEE International Conference on Software, Testing, Verification and Validation, April 9-11, Lillehammer, Norway", RSM - Dépt. Réseaux, Sécurité et Multimédia (Institut TELECOM ; TELECOM Bretagne), ETH - Eidgenössische Technische Hochschule Zürich (ETH Zurich), 2008.

[44] S. SEN, B. BAUDRY, J.-M. MOTTU. *On Combining Multi-formalism Knowledge to Select Models for Model Transformation Testing*, in "International Conference on Software Testing, Verification, and Validation, ICST'08., Lillehammer, Norway", April 2008, http://www.irisa.fr/triskell/publis/2008/Sen08a.pdf.

[45] J. SIMONIN, Y. L. TRAON, J.-M. JÉZÉQUEL, B. NICOLAS. *EA4UP: an Enterprise Architecture-Assisted Telecom Service Development Method*, in "EDOC'08 (Entreprise Distributed Object Computing Conference), Munchen, Germany", September 2008, http://www.irisa.fr/triskell/publis/2008/Simonin08a.pdf.

[46] E. DE ALMEIDA, G. SUNYE, Y. LE TRAON, P. VALDURIEZ. *Testing Peers' Volatility*, in "23rd IEEE/ACM International Conference on Automated Software Engineering (ASE 2008), L'Aquila, Italy", September 2008.

[47] E. DE ALMEIDA, G. SUNYE, P. VALDURIEZ, Y. LE TRAON. *A Framework for Testing Peer-to-Peer Systems*, in "19th IEEE International Symposium on Software Reliability Engineering (ISSRE 2008), Seattle, WA, USA", November 2008.

National Peer-Reviewed Conference/Proceedings

[48] R. BENDRAOU, M.-P. GERVAIS, X. BLANC, J.-M. JÉZÉQUEL. *Vers l'Exécutabilité des Modèles de Procédés Logiciels*, in "Langage Modèles et Objets LMO'08, Montréal, Quebec", March 2008, http://www.irisa.fr/triskell/publis/2008/Bendraou08a.pdf.

[49] F. CHAUVEL, O. BARAIS, N. PLOUZEAU, I. BORNE, J.-M. JÉZÉQUEL. *Expression qualitative de politiques d'adaptation pour Fractal*, in "Langage Modèles et Objets LMO'08, Montréal, Quebec", March 2008, http://www.irisa.fr/triskell/publis/2008/Chauvel08a.pdf.

[50] J.-M. MOTTU, B. BAUDRY, Y. LE TRAON. *Test de Transformation de Modèles : Expression d'Oracles*, in "4ièmes Journées sur l'Ingénierie Dirigée par les Modèles, Mulhouse, France", June 2008, http://www.irisa.fr/triskell/publis/2008/Mottu08b.pdf.

Scientific Books (or Scientific Book chapters)

[51] O. BARAIS, J. LAWALL, A.-F. L. MEUR, L. DUCHIEN. *Software Architecture Evolution*, T. MENS, S. D. EDS (editors), Springer Verlag, 2008, p. 233–262.

[52] M. MONPERRUS, J.-M. JÉZÉQUEL, J. CHAMPEAU, B. HOELTZENER. *Measuring Models*, in "Model-Driven Software Development: Integrating Quality Assurance", ISBN: 978-1-60566-006-6, IDEA Group Inc., Hershey, PA, USA, 2008, http://www.irisa.fr/triskell/publis/2008/Monperrus08c.pdf.

Research Reports

[53] F. MUNOZ, B. BAUDRY, O. BARAIS. *A classification of invasive patterns in AOP*, Research report, n^o inria-00266555, IRISA Research Center, http://hal.inria.fr/inria-00266555/en/, March 2008.

References in notes

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

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

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

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

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

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

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

[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.