INRIA

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

2007

Table of contents

1. Team

Scientific head
Jean-Marc Jézéquel [Professor Université de Rennes 1, HdR]

Administrative assistant
Loïc Lesage [TR Inria]

Research scientist
Olivier Barais [Assistant Professor Université de Rennes 1]
Benoit Baudry [Research scientist Inria]
Yves Le Traon [ENST Bretagne, HdR]
Noël Plouzeau [Assistant Professor Université de Rennes 1]

External collaborators
Pierre-Alain Muller [Assistant Professor Université de Haute Alsace]

Visiting scientist
Robert France [Colorado State University, USA]

Post-doctoral Fellow
Reda Bendraou [INRIA grant from october 2007]
Julien De Antoni [INRIA (project Speeds) from november 2007]
Franck Fleurey [SodiFrance]
Gilles Perrouin [INRIA (project MOPCOM) from november 2007]

Technical staff
Didier Vojtisek [Research engineer Inria]
Franck Chauvel [Université Rennes 1 (project FAROS)]
Cyril Faucher [Inria (project Openembedd)]
François Tanguy [Inria Associated Engineer]
David Touzet [Inria (project Systematic) until September 2007]
Marc Skipper [Inria (project Speeds) until September 2007]

PhD Students
Erwan Brottier [CIFRE grant]
Romain Delamare [Brittanny Council grant]
Marouane Himdi [CIFRE grant until October 2007]
Martin Monperus [DGA grant]
Brice Morin [INRIA grant since October 2007]
Jean-Marie Mottu [MENRT grant]
Freddy Munoz [INRIA grant since October 2007]
Sébastien Saudrais [INRIA grant until December 2007]
Jacques Simonin [FTR&D]
Sagar Sen [INRIA grant from May 2007]
Jim Steel [INRIA grant until February 2007]

Interns Students
Rafael Duarte [Centre de Informatica Universidad Federal de Pernambuco, Brazil]
Claudia Rostagnol [UdelaR (Universidad de la Republica), Uruguay]

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 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 1.0 of its Kermeta software (KERnel META-modelling). 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.

- Prof. Robert France, a leading academic in Model Driven Engineering from Colorado State University, USA, has spend a 6-months sabbatical within Triskell, in the context of the MATT associated team. This has led to several co-publications, as well as the co-edition with J.-M. Jézéquel of a special issue of Transactions on Aspect Oriented Software Development on MDE and Aspects (to be published in 2008).
- Triskell was a major force behind the creation of a new IEEE International Conference on Software Testing Verification and Validation, with Benoit Baudry as vice-chair of the Steering Committee and Yves Le Traon as member of the steering committee. The first edition is a success, ICST can already be considered as the leading conference of the domain with 224 full papers submissions (14% acceptance rate) and 8 satellites workshops.

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 [58], [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 [60].

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 [57]. 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 [63] 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 [59]).

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 [61] 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) [64]. 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 [62], 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.

3.3. Mathematical Foundations for Distributed and Reactive Software

Keywords: *Labeled transition systems*, *event structures*, *partial orders*.

3.3.1. Transition systems

A labeled transition system (or LTS) is a directed graph which edges, called transitions, are labeled by letters from an alphabet of **events**. The vertices of this graph are called **states**. A LTS can be defines as a tuple $M = (Q^M, A, T^M \subset Q^M \times A \times Q^M, q^M_{init})$, in which Q^M is a set of states, q^M_{init} is an initial state, A is a set of events, T^M is a transition relation.

Note that from this definition, the set of states in a LTS s not necessarily finite. Usually, the term **finite state automata** is used to designate a LTS with a finite set of states and events. In fact, automatas are the simplest models that can be proposed. They are often used to model reactive (and usually distributed) systems. Within this framework, events represent the interactions (inputs and outputs) with the environment. The term input/output LTS (IO-LTS) is often used to designate this kind of automata.

Labeled transition systems are obtained from reactive systems specifications in high-level description languages such as UML, or Domain Specific Modeling Languages (such as the Requirement Definition Language, see Section 5.5). The construction of a LTS from a specification is done using an operational semantics for this language, defined for instance with Kermeta (see Section 5.1).

Classical algorithms in language theory build explicitely finite state automatas, that are usually integrally stored in memory. However, for most of the problems we are interested in, exhaustive construction or storage of an LTS is not mandatory. Partial construction of an LTS is enough, and strategies similar to lazy evaluation in functional programs can be used: the only part of LTS computed is the part needed for the algorithm.

3.3.2. Non interleaved models

One of the well known drawbacks of LTSs [56] is that concurrency is represented by means of behaviors interleaving. This is why LTS, automatas and so on are called “interleaved models”. With interleaved models, a lot of memory is lost, and models represented can become very complex. Partial order models partially solve these problems.

A partial order is a tuple $(E, \leq, \Pi, \phi, \Sigma, I)$ in which:

- E represents a set of atomic events, that can be observable or not. Each event is the occurrence of an action or operation. It is usually considered that an event is executed by an unique process in an system.
- $\leq$ is a partial order relation that describes a precedence relation between events. This order relation can be obtained using the hypotheses that:
  1. processes are sequential : two events executed by the same process are causally ordered.
  2. communications are asynchronous and point to point: the emission of a message precedes its reception.
- σ is an alphabet of actions.
- I is a set of process names
- $\Pi : \Sigma \rightarrow I$ is an action placement function.
- $\phi : E \rightarrow \Sigma$ is an event labeling function

A partial order can be used to represent a set of executions of a system in a more “compact” way than interleaved models. Another advantage of partial order models is to represent explicitely concurrency: two events that are not causally dependant can be executed concurrently. In a LTS, such a situation would have been represented by an interleaving.

Partial order models are thus the underlying semantics framework of many behavioral modeling languages, including the Message Sequence Charts in UML.

4. Application Domains

4.1. Software for Telecommunication and Real-Time Systems

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

In large scaled distributed systems such as developed for telecommunications, building a new application from scratch is no longer possible. 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 reducing the time to market (such as derivation and validation tools).

Triskell has gained experience in model engineering, and finds here a propitious domain. The increasing software complexity and the reliability and reusability requirements fully justify the methods developed by our project. The main themes studied are reliable software components composition, UML-based developments validation, and test generation from UML models, either at requirement level or at design level.

The research activity in Triskell focuses at the same time on development efficiency and reliability. Our main applications mainly concern reliable construction of large scale communicating software, and object oriented (and aspect oriented) systems testing.

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 an approach where they 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 mainly consist in ensuring that a system implementation conforms to its specifications. Whatever the efforts spent for development, this phase is of real importance to ensure 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, Franck Fleurey, Cyril Faucher, Jean-Marc Jézéquel, Jean-Marie Mottu, Pierre-Alain Muller, Jim Steel, Fraçois Tanguy, David Touzet, 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 that allows 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. UMLAUT NG : Extendible model transformation tool and framework

Keywords: *MDA*, *MOF*, *UML*, *component*, *model transformation*, *patterns*, *validation*.

Participants: Franck Chauvel, Franck Fleurey, Jean-Marc Jézéquel, Vincent Mahé, Didier Vojtisek [correspondant].

UMLAUT is a model tranformation framework for UML (*Unified Modelling Language*) developped since 1998 by Triskell. It allows to apply complex transformations to UML models. UMLAUT NG is the evolution of this framework in order completly take into account the latest versions of UML metamodel and fit to the global approach of the team. It is now distributed as a Kermeta MDK. Thanks to the operationnal semantic expressed using Kermeta, this allows to improve the correctness of the model by applying one or many of the available methods. This includes the simulation of the user model, the use of validation tools like the one from the CADP toolbox, the application of complex transformations, the generation code, etc.

5.3. Sintaks : Textual syntaxes for models

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

Participants: Erwan Brottier, Pierre-Alain Muller, David Touzet, 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.

5.4. Kompose : Generic Model Composition Tool

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

Participants: Franck Fleurey [correspondant], Benoit Baudry.

Kompose is a generic framework to support model composition. It is implemented in Kermeta and it can be specialized for any metamodel in order to easily define composition operators. The framework is made of a generic model element merge algorithms and a basic composition 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 and database schemas.

5.5. Requested : a toolbox for requirement simulation and testing

Keywords: *Test*, *requirement simulation*, *requirement testing*, *textual requirements*, *use cases*.

Participants: Benoit Baudry [correspondant], Yves Le Traon, Franck Fleurey, David Touzet.

The objective of the Requested toolbox is to offer a MDA transformation from textual requirements to simulable requirements within the UML (use cases + scenarios). It allows the simulation of requirements and the automated generation of test objectives. Two tools are under development:

1. The transformation of natural language requirements expressed in the RDL language (Requirement Description Language) into a use case model, enhanced with contracts.
2. The UCTS system allows the simulation of the use case model, enhanced with contracts, and the automated generation of test objective.

More precisely, UCTSystem is a prototype designed to perform automatic test generation from UML requirements. It uses UML use cases enhenced with contracts (*i.e. precondition and postconditions*) to build an execution model allowing all valid sequences of use cases. Using this execution model and several test criteria, it generates test objectives as sequence of use cases to exerce. It includes both criteria for functional testing and a criterion for robusness testing. Those test objectives are then mapped into test cases using test templates.

6. New Results

6.1. Contract-based and Aspect Oriented Design

6.1.1. Behavioral Aspects and Weaving

Participants: Jacques Klein, Jean-Marc Jézéquel, Franck Fleurey.

We proposed [32], [20], [33] a semantically well founded weaving process for scenarios: first, a detection phase searches parts in a base model of scenarios, second the composition phase builds a woven model of scenarios by composing the advice into the base model for each detected part. The entire weaving process is automated and is implemented as model transformations within the Kermeta environment.

6.1.2. Aspect Oriented Modeling

Participants: Brice Morin, Rodrigo Ramos, Olivier Barais, Jean-Marc Jézéquel.

In [37], we present our Generic Aspect-Oriented Modeling Framework and show how it can automatically be specialized for any specific domain. Thus, we can leverage the notion of Aspect, for any domain metamodel, and propose a better modularity. Connected to this work, we point out [34] the needs for variability in the AOM approaches and introduce seamless variability mechanisms in an existing AOM approach to improve reusability. Such mechanisms turn standard aspects into highly reusable aspects.

6.1.3. Vigilant Usage of Aspects

Participants: Freddy Munoz, Olivier Barais, Benoit Baudry.

Aspect Oriented Software Development presents two problems that impact directly on its confidence, and represents a major drawback for its mainstream adoption. These problems are referred as "AOSD evolution paradox" and "Invasive aspects". In [41] we present a survey of the existing approaches to address these problems. Then, we present the first lines of an approach based on contract for aspects to solve these problem.

6.1.4. Component Based Software Architecture Evolution with Aspects

Participant: Olivier Barais.

We provided an overview, comparison and detailed treatment of the various state-of-the-art approaches to evolving software architectures [16]. Furthermore, we discuss one particular framework for software architecture evolution in more detail. In [15], we compared aspect oriented approaches, view oriented approaches, and subject oriented approaches to compose and reuse pieces of software in information systems.

This work has been carried out in collaboration with the ADAM project-team at INRIA Lille (Anne-Françoise Le Meur, Julia Lawall, and Laurence Duchien)

6.1.5. Time Quality of Service

Participants: Sébastien Saudrais, Olivier Barais, Noël Plouzeau, Jean-Marc Jézéquel.

In [43], [44], [13], we proposed an approach for the introduction of time properties into a component-based application. The time properties are introduced in the component's behaviour and contracts. The introduction is made by using patterns, each pattern representing a time property like time response or delay.

In [45], [13], [23], we presented the monitoring of time quality of service properties. The approach, from the specification to the implementation, is applied on a Lego Mindstorms robot.

This work has been carried out in collaboration with Laurence Duchien (INRIA Lille).

6.1.6. Requirements Modelling

Participants: Erwan Brottier, Benoit Baudry, Yves Le Traon, David Touzet.

In [26] we propose a model-driven mechanism for generating requirements models from textual specifications. In [27], we detail how several requirement documents can be merged into a single model. This model can then serve for inconsistencies detection in the requirements.

6.2. Model-Based Testing

6.2.1. Test and Aspect Oriented Modeling

Participants: Jacques Klein, Romain Delamare, Olivier Barais, Benoit Baudry.

Aspect-Oriented Modelling (AOM) complements Model-Driven Engineering (MDE) by extending the decomposition capabilities with separation of concerns at the same level of abstraction. Using transformations in software development has many benefits, but when an error is introduced into a model, it is propagated to later refinements. Such propagation makes it harder to trace errors to their source and consequently more difficult to correct. Design models with a good separation of concerns reduce the scope for error propagation as errors are localised. However, to ensure that errors are not propagated between concern models at different levels of abstraction, a technique for detecting errors in design models is required. In [33], we present KerTheme, an approach to supporting error detection in AO models through testing. In KerTheme, testing is supported by defining two views of a concern: an executable model and a model of its expected behaviour. The testing process consists of checking the consistency between a trace from executing the concern model and the expected behaviour of the concern.

6.2.2. Security Testing

Participants: Yves Le Traon, Benoit Baudry.

In [38], [39], we introduce security policy testing as a specific target for testing. We propose two strategies for producing security policy test cases, depending if they are built in complement of existing functional test cases or independently from them. We propose test selection criteria to produce tests from a security policy. To quantify the effectiveness of a set of test cases to detect security policy flaws, we adapt mutation analysis and define security policy mutation operators.

6.2.3. Automatic Model Synthesis

Participants: Sagar Sen, Benoit Baudry.

In collaboration with Doina Precup and Hans Vangheluwe from McGill University (Canada), we studied constraint-solving techniques in order to automatically synthesize models that conform to structural constraints defined in a metamodel as well as to additional constraints that are defined by testers or as the pre-condition to a model transformation [46], [47]. This work aims at automating the generation of test data for model transformations.

6.2.4. Qualifying Input Test Data for Model Transformations

Participants: Franck Fleurey, Jean-Marie Mottu, Benoit Baudry, Pierre-Alain Muller, Yves Le Traon.

Testing a model transformation is typically performed by checking the results of the transformation applied to a set of input models. While it is fairly easy to provide some input models, it is difficult to qualify the relevance of these models for testing. In [17], we propose a set of rules and a framework to assess the quality of given input models for testing a given transformation. Furthermore, the framework identifies missing model elements in input models and assists the user in improving these models.

6.3. Model-Driven Engineering

6.3.1. Kermeta

Participants: Pierre-Alain Muller, Franck Fleurey, Olivier Barais, Jean-Marie Mottu, Didier Vojtisek, Mark Skipper, Jean-Marc Jézéquel.

In [53], [51], [50], we present new Kermeta features such as the direct support of OCL language, the aspect merge operator of Kermeta and the Syntaks tool.

6.3.2. Model Typing

Participants: Jim Steel, Jean-Marc Jézéquel.

We proposed a simple extension to object-oriented typing to better cater for a model-oriented context, including a simple strategy for typing models as a collection of interconnected objects [24], [14]. We suggest extensions to existing type system formalisms to support these concepts and their manipulation. We show using this system and motivating examples how this extended approach permits more flexible reuse, while preserving type safety.

6.3.3. Model Measurement

Participants: Martin Monperrus, Jean-Marc Jézéquel.

Model-driven engineering (MDE) supports the development of software by means of a set of languages, processes, methods and tools. Domain specific languages (DSL) are specified and built so as to raise the level of abstraction and increase the productivity. Assessing the quality of a product built in a given DSL, i.e. a model, involves measurement. Time-to-market and cost constraints do not allow engineers to build ad hoc metric environments for each DSL. In [36], we specified a framework to concisely define and automatically implement an open-ended family of metrics. This framework makes it possible to concentrate only on the conceptual level of the development of metrics [21].

6.3.4. Matching Model Snippet

Participants: Rodrigo Ramos, Olivier Barais, Jean-Marc Jézéquel.

An important demand in Model-Driven Development is the simple and efficient expression of model patterns. Current approaches tend to distinguish the language they use to express patterns from the one for modelling. Consequently, productivity is reduced by dealing with a distinct new language, and new intermediate steps are introduced in order to support pattern-matching. In [42], we propose a framework for expressing patterns as *model-snippets*. We present how model-snippets are specified upon concepts in a given domain (meta-model), and how we perform pattern-matching with model-snippets, whatever the meta-model.

6.3.5. Model Composition

Participants: Franck Fleurey, Robert France, Benoit Baudry.

In [28], [18], we study the composition of models that is a basic mechanim for aspect-oriented modelling. We propose an algorithm and a generic tool that can add composition capabilities to any metamodel. In [35] we apply the proposed technique to the modelling of fault-tolerant systems.

6.3.6. Models at Runtime

Participants: Pierre-Alain Muller, Olivier Barais.

Models at runtime are considered a key enabling technology for systems that control themselves as they operate. The automatic-control community has developed extensive theories and experiences in qualifying the properties of controller and systems; including stability, observability and commandability. In [40], we propose to use control-theory for describing self-adaptive model-driven systems.

6.3.7. Model-Driven Engineering for Software Migration

Participants: Franck Fleurey, Benoit Baudry, Jean-Marc Jézéquel.

In [29], we report on the use of Model-Driven Engineering as an efficient, flexible and reliable approach for software migration. We discuss how MDE is economically profitable and is cost-effective over the migration through out-sourced manual re-development. This work has been done in collaboration with Sodifrance and is illustrated with the migration of a large-scale banking system.

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, Mark Skipper, 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 aims at substantially improving the competitiveness of the European industry in this critical economic sector by marrying design competence with deep technical insights and theoretical foundations. 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 SP.2 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. AOSD-Europe (Network of Excellence)

Keywords: *Aspect Oriented Design*.

Participants: Jean-Marc Jézéquel, Noël Plouzeau, Yves Le Traon, Olivier Barais, Sébastien Saudrais, 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

7.3. 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 has a double core, consisting of leading-edge research in embedded systems design issues (described later in this document) in the Joint Programme of Research Activities (JPRA), and complementary activities around shared platforms and staff mobility in the Joint Programme of Integration Activities (JPIA).

The JPRA activities are pure research, and the JPIA are complementary efforts for integration. Both work towards deep integration between the participating research teams.

The JPRA and JPIA are structured into clusters - one for each of the selected topics in embedded systems design (in red). Teams may be involved in one or several clusters.

Around this double core is the Joint Programme of Activities for Spreading Excellence (JPASE). These are complementary activiites for disseminating excellence across all available channels, targetting industry, students, and other European and international research teams.

Building the embedded systems design scientific community is an ambitious programme. To succeed, ARTIST2 will build on the achievements and experience from the ARTIST1 FP5 Accompanying Measure (http://www.artist-embedded.org/) on Advanced Real-Time Systems. ARTIST1 provided the opportunity to test the concept of a two-level integration (within and between clusters) four clusters in ARTIST2 originated as "actions" in ARTIST1. Building the ARTIST2 consortium and associated structure is the culmination of discussions and ambitions elaborated within ARTIST1.

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/

7.4. Mopcom Hard (RNTL)

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

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

Mopcom hard 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 embedded system typically based on system-on-chip (SOC). The project will produce a complete methodology and development environment dedicated to the domain.

In 2007, Triskell participated to the development process and the specification of precise metamodels (using Kermeta) for each steps of the process. Triskell also studied the MARTE UML profile as the main candidate metamodel for several of these steps.

Project duration: 2007-2010 years

Triskell budget share: 232 851 euros

Project Coordinator: Thalès (TSA)

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

7.5. KEREVAL

Keywords: *components*, *diagnosis*, *extra functional testing*, *probes*.

Participants: Marouane Himdi, Yves Le Traon.

In addition to detection of errors related to design, coding or deployment of an application, the diagnosis is a well-known technique for understanding the behaviour of a software system and an absolute requirement for its improvement. Unfortunately, applications become more difficult to diagnosis as functionalities provided become complex. In collaboration with the KEREVAL company, we explore the use of dynamic probes (sensors) that will be injected into running system to collect various information. The innovative aspect of this approach is the use of generic probes to develop diagnosis framework [12].

7.6. Sodifrance / SOFT-MAINT

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

Participants: Franck Fleurey, Benoit Baudry, Jean-Marc Jézéquel.

Regression testing is a crutial issue in software migration projects. In practice, validation ususally respresents as much as half the overall cost of migration. In the context of Franck Fleurey's INRIA industrial post-doc at Sodifrance, we investigate the use of models for generating functional test scenarios. This work is done in the context of the model-driven migration process developed by Sodifrance. Based on models reverse-engineered for migration, we have defined several test criteria. The proposed techniques are evaluated in the context of industrial migration projects.

Project duration: 2007-2008

Triskell budget share: 16 keuros

7.7. France Telecom R& D

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

Participants: Yves Le Traon, Jacques Simonin, Jean-Marc Jézéquel.

Since March 2006, we have a collaboration with 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 [48]. In this context, Jean-Marc Jézéquel acts as the Ph.D advisor of Jacques Simonin, a senior FTR&D engineer.

Project duration: 2006-2009

Triskell budget share: contract to be finalized

7.8. 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 will be 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. On this structure, the generic and specific engineering components are case by case integrated depend-ing on the needs of the industrial projects case studies. In this context, Triskell has to develop 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: 75 048 euros

Number of person/years: 43.2

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

7.9. 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.

The 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 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

7.10. OpenEmbeDD (RNTL)

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

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

OpenEmbeDD is a project supported by the RNTL program. OpenEmbeDD is an Eclipse open-source platform, extensible, standardized and generic, 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 project takes part in the topic: embedded systems of RNTL. This project contributes to continue the effort of the RNTL for developing open-source software, a main topic of the RNTL. 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. Kermeta is built on top of the Eclipse framework, it provides an interpreter, a compiler and editors, in particular a graphical editor. In this context, Triskell has to develop an Eclipse plugin providing a graphical editor to design executable metamodels, a second version is now available. This editor is generated from the generator developed in the TopCaseD project (Airbus et al.). Triskell is a member of the TopCaseD project, thus Triskell's members participate to the specification of the source generator for building automatically graphical editors. Triskell has to develop a compiler to target an execution based on Java Runtime, a first version of this feature is available. Some cooperations are in progress in the context of the OpenEmbeDD project, like with CS-SI and the France Telecom Research Unit.

Project duration: 2006-2009
Project budget: 7 Meuros
Triskell budget share: 300 keuros
Number of person/years: 185.3
Project Coordinator: INRIA
Participants: Airbus, Anyware Technologies, CEA-List, CS-SI, France Telecom, INRIA, LAAS, THALES (DAE and RT), Verimag

7.11. 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
Project budget: 1 Meuros
Triskell budget share: 80 keuros
Project type: exploratory
Number of person/years: 30.8
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)

7.12. 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)
- GPL proposal - 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 2007, Triskell project participates to normalization action at OMG (http://www.omg.org/):

- Triskell project has presented Kermeta in OMG's workshop on Precise Behavioral Semantics for Domain Specific Modeling Languages.
- Triskell project is involved in to the Executable UML foundation RFP that will standardize a subset of UML2.0 with a more precise execution semantic.
- Triskell project was also involved in other OMG groups which are related to the team interests. For example, it participates to the ORMSC group which formalizes the MDA approach, to the MDA user SIG which represents the end user point of view for MDA. It is also involved in the more general 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 model validation with testing. In this context, Robert France visited Triskell 6 months from January to June 2007, Franck Fleurey spent a month in CSU in October 2007, and from December 2007 Benoit Baudry started a one year sabbatical at CSU.
- Modelling Simulation and Design Lab, Mc Gill University, Montreal Collaboration on model synthesis and constraint solving issues for model transformation testing. Sagar Sen has started a PhD as a co-direction between IRISA and Mc Gill.
- In February 2007 we started 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.

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. *Pierre-Alain Muller*

is an Associate Editor of the Journal on Software and System Modeling: SoSyM.

9.1.1.3. *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”:

- Mohamed Khalgui, February 2007, université de Nancy (referee);
- Jim Steel, April 2007, université de Rennes (adviser);
- Reda Bendraou, September 2007, université Paris VI (referee);
- Gilles Perrouin, September 2007, Uni. Namur & Uni. Luxembourg (referee);
- Julien De Atoni, October 2007, INSA Lyon (referee);
- Sébastien Saudrais, December 2007, université de Rennes (adviser);
- Marouane Himdi, December 2007, université de Rennes (adviser);

9.1.2.2. *Olivier Barais*

has been a member of the program committee of the following special issue journal:

- Ingéniérie des Systèmes d'Information (ISI), RTSI série special issue on MDE.

9.1.2.3. *Yves Le Traon*

was in the examination committee of the following PhD thesis:

- Patricia Mouy, June 2007, Univ. Evry et CEA (referee)
- Sébastien Labbe, October 2007, Paris VI et CEA (referee).
- Laily Hashim, November 2007, Monash University (Australia) (referee)
- Marouane Himdi, December 2007, Rennes 1 (examiner)

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
- GPCE'07 Sixth International Conference on Generative Programming and Component Engineering, Salzburg, Austria, October 1-3, 2007
- SPLC 2007 The 11th International Software Product Line Conference, Kyoto, Japan, 10-14 Sept. 2007
- CBSE 2007 The 10th International Symposium on Component-Based Software Engineering, Boston, USA, July 9 - July 11, 2007
- EA 2007, Early Aspects 2007, Co-located workshop with AOSD 2007, Vancouver, Canada
- AFADL'2007, (Approches Formelles dans l'Assistance au Développement de Logiciels), Namur, Belgique, June 13-15 2007
- LMO 2007, Toulouse, March 27-29, 2007
- VaMoS 2007 First International Workshop on Variability Modelling of Software-intensive Systems Limerick, Ireland, January 16, 2007
- QoSA 2007, Third International Conference on the Quality of Software-Architectures, July 12-13, 2007 Tufts University, Medford (Boston area), Massachusetts, USA

9.1.3.2. Yves Le Traon

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

- The 18th IEEE International Symposium on Software Reliability Engineering (ISSRE 2007) November 2007 - Trollhättan, Sweden.
- IEEE International Conference on Software Testing, Verification and Validation (ICST 08)
- 3d int. workshop on Model design and Validation (MoDeVVa 2007)
- 4th int. workshop on Model design and Validation (MoDeVVa 2008)
- 3rd int. workshop on Mutation testing (Mutation 2007)

9.1.3.3. Pierre-Alain Muller

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

- MODELS 2007 The 10th International Conference on Model Driven Engineering Languages and Systems (formerly the UML series of conferences) Nashville, USA, 1-6 October 2007

9.1.3.4. 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
- ICST'08 International Conference on Software Testing Verification and Validation
- MoDeVVa'07 workshop on Verification and Validation for MDE, associated to MoDELS'07
- Quality and Modelling workshop associated to MoDELS'07
- M-ADAPT, Model-Driven Software Adaptation workshop associated to ECOOOP'07
- A-MOST'07 workshop on Advances in Model-Based Testing, associated to ISSTA'07
- ADVCOMP'07, International Conference on Advanced Engineering Computing and Applications in Sciences
- Mutation'07 associated to TAIC-PART'07

9.1.3.5. Franck Fleurey

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

- MoDeVVa'07 workshop on Verification and Validation for MDE, associated to MoDELS'07

9.1.4. Workshops

J.-M. Jézéquel gave an invited talk at the Workshop OPEES, Paris, May 2007.

B. Baudry was co-organizer with Alexander Pretschner, Alain Faivre and Sudipto Ghosh of the 4th MoDeVa workshop in conjunction with MoDELS'07. B. Baudry is workshop chair for ICST'08.

O. Barais was the organizer of the *Third Kermeta days*, on December 4-5. He has also been a member of the programme committee of the following workshops:

- 3èmes Journée Francophone sur le Développement de Logiciels Par Aspects JFDLPA 2007), (associated to LMO'07 / IDM'07), Toulouse, France, March 2007.
- 3èmes Journées sur l'ingénierie Dirigée par les Modèles (IDM'07), Toulouse, France, March 2007.
- Model Patterns 2007, (associated to IDM'07), Toulouse, France, March 2007.

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 a member of the Steering Committee of the MODELS/UML Conferences series. He is appointed to the board of the Committee of Projects of INRIA Rennes. He belongs to the evaluation committee of the SIO division of DGA (Direction Générale de l'Armement). He is a member of the scientific council of MIPS (Université de Haute Alsace). 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 VP of the Université de Haute Alsace.
- Benoit Baudry is on the steering committee 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).

Triskell has initiated a collaboration with the EPI Bunraku thanks to the internship of Claudia Rostagnol. This works aims to bring the benefits of MDE for devolopping the autonomous agent used in virtual reality.

Triskell has also been visited by:

- Nelly Bencomo, U. Lancaster, UK (2 weeks)
- Gordon Blair, U. Lancaster, UK (1 week)
- Sudipto Ghosh, Colorado State University, USA (2 weeks)
- Iman Poermono, Kings College, Great Britain (1 week)
- Hans Vangheuwe, Mc Gill university, Quebec (1 week)

10. Bibliography

Major publications by the team in recent years

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

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

[3] J.-M. JÉZÉQUEL, D. DEVEAUX, Y. LE TRAON. *Reliable Objects: a Lightweight Approach Applied to Java*, in "IEEE Software", vol. 18, n^{o} 4, July/August 2001, p. 76–83.

[4] 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.

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

[6] J.-M. JÉZÉQUEL, M. TRAIN, C. MINGINS. *Design Patterns and Contracts*, ISBN 1-201-30959-9, Addison-Wesley, October 1999.

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

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

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

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

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

Year Publications

Doctoral dissertations and Habilitation theses

[12] M. HIMDI. *Performances des systèmes distribués : proposition d'une plateforme de supervision fédératrice de la supervision*, Ph. D. Thesis, Université de Rennes 1, December 2007.

[13] S. SAUDRAIS. *Qualité de service temporelle pour composants logiciels*, Ph. D. Thesis, Université de Rennes 1, December 2007.

[14] J. STEEL. *Typage de modèles*, Ph. D. Thesis, Université de Rennes 1, April 2007.

Articles in refereed journals and book chapters

[15] O. BARAIS, P. LAHIRE, A. MULLER, N. PLOUZEAU, G. VANWORMHOUDT. *Évaluation de l'apport des aspects, des sujets et des vues pour la composition et la réutilisation des modèles*, in "RSTI-L'Objet", to appear, 2007.

[16] O. BARAIS, J. LAWALL, A.-F. LE MEUR, L. DUCHIEN. *Software Architecture Evolution*, in "Software Evolution", to appear, Springer Verlag, december 2007.

[17] F. FLEUREY, B. BAUDRY, P.-A. MULLER, Y. LE TRAON. *Towards Dependable Model Transformations: Qualifying Input Test Data*, in "Journal of Software and Systems Modeling (SoSyM)", 2007.

[18] R. B. FRANCE. *Model Driven Engineering: Managing Software Complexity*, in "Shaping the Future: 10 years of IrisaTech", IrisaTech, 2007, p. 71–78.

[19] J.-M. JÉZÉQUEL. *New Software Architectures*, in "Shaping the Future: 10 years of IrisaTech", IrisaTech, 2007, p. 51–54.

[20] J. KLEIN, F. FLEUREY, J. M. JÉZÉQUEL. *Weaving Multiple Aspects in Sequence Diagrams*, in "Transactions on Aspect-Oriented Software Development (TAOSD)", vol. LNCS 4620, 2007, p. 167-199.

[21] M. MONPERRUS, J.-M. JÉZÉQUEL, J. CHAMPEAU, B. HOELTZENER. *Measuring models*, in "Model-Driven Software Development: Integrating Quality Assurance", To appear, IDEA Group Inc., Hershey, PA, USA, 2007.

[22] S. PICKIN, C. JARD, T. JÉRON, J.-M. JÉZÉQUEL, Y. LE TRAON. *Test Synthesis from UML Models of Distributed Software*, in "IEEE Transactions on Software Engineering", vol. 33, n° 4, April 2007, p. 252–268.

[23] S. SAUDRAIS, O. BARAIS, L. DUCHIEN, N. PLOUZEAU. *From formal specifications to QoS monitors*, in "Journal of Object Technology, Special Issue on Advances in Quality of Service Management", vol. 6, n° 11, December 2007, p. 7–24.

[24] J. STEEL, J.-M. JÉZÉQUEL. *On Model Typing*, in "Journal of Software and Systems Modeling (SoSyM)", vol. 6, n° 4, December 2007, p. 401–414.

Publications in Conferences and Workshops

[25] Z. ALTAHAT, T. ELRAD, D. VOJTISEK. *Using Aspect Oriented Modeling to localize implementation of executable models*, in "Models and Aspects workshop, at ECOOP 2007", July 2007, http://www.irisa.fr/triskell/publis/2007/Altahat07a.pdf.

[26] B. BAUDRY, C. NEBUT, Y. LE TRAON. *Model-driven Engineering for Requirements Analysis*, in "EDOC'07 (Entreprise Distributed Object Computing Conference), Annapolis, MD, USA", 2007.

[27] E. BROTTIER, B. BAUDRY, Y. LE TRAON, D. TOUZET, B. NICOLAS. *Producing a Global Requirement Model from Multiple Requirement Specifications*, in "EDOC'07 (Entreprise Distributed Object Computing Conference), Annapolis, MD, USA", 2007.

[28] F. FLEUREY, B. BAUDRY, R. FRANCE, S. GHOSH. *A Generic Approach For Automatic Model Composition*, in "Aspect Oriented Modeling (AOM) Workshop, Nashville, USA", October 2007.

[29] F. FLEUREY, E. BRETON, B. BAUDRY, A. NICOLAS, J.-M. JÉZÉQUEL. *Model-Driven Engineering for Software Migration in a Large Industrial Context*, in "MoDELS'07, Nashville, TN, USA", 2007.

[30] R. FRANCE, F. FLEUREY, R. REDDY, B. BAUDRY, S. GHOSH. *Providing Support for Model Composition in Metamodels*, in "EDOC'07 (Entreprise Distributed Object Computing Conference), Annapolis, MD, USA", 2007.

[31] R. FRANCE, B. RUMPE. *Model-driven Development of Complex Software: A Research Roadmap*, in "FOSE '07: 2007 Future of Software Engineering, Washington, DC, USA", IEEE Computer Society, 2007, p. 37–54.

[32] J.-M. JÉZÉQUEL. *Modeling and Aspect Weaving*, in "Methods for Modelling Software Systems (MMOSS)", E. BRINKSMA, D. HAREL, A. MADER, P. STEVENS, R. WIERINGA (editors), Dagstuhl Seminar Proceedings, <http://drops.dagstuhl.de/opus/volltexte/2007/864> [date of citation: 2007-01-01], n^{o} 06351, Internationales Begegnungs- und Forschungszentrum fuer Informatik (IBFI), Schloss Dagstuhl, Germany, 2007.

[33] J. KLEIN, B. BAUDRY, O. BARAIS, A. JACKSON. *Introduction du test dans la modélisation par aspects*, in "Troisième Journées sur l'Ingénierie Dirigée par les Modèles (IDM'2007), Toulouse, France", March 2007.

[34] P. LAHIRE, B. MORIN, G. VANWORMHOUDT, A. GAIGNARD, O. BARAIS, J.-M. JÉZÉQUEL. *Introducing variability into Aspect-Oriented Modeling approaches*, in "In Proceedings of ACM/IEEE 10th International Conference on Model Driven Engineering Languages and Systems (MoDELS 07), Nashville, TN, USA", October 2007.

[35] L. MICHOTTE, R. FRANCE, F. FLEUREY. *Modeling and Integrating Architural Software Fault Tolerance Features*, in "EDOC'07 (Entreprise Distributed Object Computing Conference), Annapolis, MD, USA", 2007.

[36] M. MONPERRUS, J. CHAMPEAU, B. HOELTZENER. *Counts count*, in "Proceedings of the 2nd Workshop on Model Size Metrics (MSM'07) co-located with MoDELS'2007", 2007.

[37] B. MORIN, O. BARAIS, J.-M. JÉZÉQUEL, R. RAMOS. *Towards a Generic Aspect-Oriented Modeling Framework*, in "Models and Aspects workshop, at ECOOP 2007", July 2007, http://www.irisa.fr/triskell/publis/2007/morin07a.pdf.

[38] T. MOUELHI, Y. LE TRAON, B. BAUDRY. *Mutation analysis for security tests qualification*, in "Mutation'07 workshop associated to TAIC-Part'07, Cumberland Lodge, Windsor, UK", 2007.

[39] T. MOUELHI, Y. LE TRAON, B. BAUDRY. *Testing security policies: going beyond functional testing*, in "ISSRE'07 (Int. Symposium on Software Reliability Engineering), Trollhättan, Sweden", 2007.

[40] P.-A. MULLER, O. BARAIS. *Control-theory and models at runtime*, in "Proceedings of the Models Workshop on Models@Runtime, Nashville, USA", oct 2007.

[41] F. MUNOZ, O. BARAIS, B. BAUDRY. *Vigilant usage of Aspects*, in "in Proceedings of ADI 2007 - Workshop on Aspects, Dependencies, and Interactions at ECOOP 2007, Berlin, Germany", July 2007.

[42] R. RAMOS, O. BARAIS, J.-M. JÉZÉQUEL. *Matching Model-Snippets*, in "In Proceedings of ACM/IEEE 10th International Conference on Model Driven Engineering Languages and Systems (MoDELS 07), Nashville, TN, USA", October 2007.

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

[44] S. SAUDRAIS, O. BARAIS, N. PLOUZEAU. *Integration of time issues into component-based applications*, in "The 10th International ACM SIGSOFT Symposium on Component-Based Software Engineering (CBSE'07), Medford (Boston area), Massachusetts, USA", Springer Lecture Notes in Computer Science (LNCS), July 2007.

[45] S. SAUDRAIS, O. BARAIS, N. PLOUZEAU. *Monitoring your Lego Mindstorms with Giotto*, in "in Proceedings of ARTIST International Workshop on Tool Platforms for Modeling, Analysis and Validation of Embedded Systems, Berlin, Germany", A Workshop affiliated with CAV 2007, July 2007.

[46] S. SEN, B. BAUDRY, D. PRECUP. *Partial Model Completion in Model Driven Engineering using Constraint Logic Programming*, in "INAP'07 (International Conference on Applications of Declarative Programming and Knowledge Management), Würzburg, Germany", 2007.

[47] S. SEN, H. BAUDRY. *Domain-specific Model Editors with Model Completion*, in "Proceedings of MPM Workshop associated to MoDELS'07", 2007.

[48] J. SIMONIN, Y. LE TRAON, J.-M. JÉZÉQUEL. *An Enterprise Architecture Alignment Measure for Telecom Service Development*, in "EDOC'07 (Entreprise Distributed Object Computing Conference), Annapolis, MD, USA", 2007.

[49] T. ZIADI, J.-M. JÉZÉQUEL. *PLiBS: an Eclipse-based tool for Software Product Line Behavior Engineering*, in "Proc. of 3rd Workshop on Managing Variability for Software Product Lines, SPLC 2007, Kyoto, Japan", September 2007.

Miscellaneous

[50] O. BARAIS. *Séparation des préoccupations en phase de méta-modélisation*, March 2007, http://www2.lifl.fr/~mullera/CoMo07.htm, Atelier Composition de modèles (Cosmo) de la Conférence Langage et Modèles à Objets (LMO'07) et de la 3 ème Journée sur l'Ingénierie Dirigée par les Modèles.

[51] O. BARAIS, F. FLEUREY, P.-A. MULLER, D. VOJTISEK, J.-M. JÉZÉQUEL. *Nouvelles fonctionnalités de Kermeta*, March 2007, Session Démonstrations des 3 ème Journées sur l'Ingénierie Dirigée par les Modèles.

[52] C. FAUCHER, J.-Y. LAFAYE. *Model Driven Engineering for implementing the ISO 19100 series of international standards*, in "In Proceedings of the 8th International Symposium on GIS and Computer Mapping for Coastal Zone Management (CoastGIS 07), Santander, Spain", http://www.coastgis07.com, vol. 2, October 2007.

[53] J.-M. MOTTU, O. BARAIS, M. SKIPPER, D. VOJTISEK, J.-M. JÉZÉQUEL. *Intégration du support OCL dans Kermeta. Spécifiez la sémantique statique de vos méta-modèles.*, June 2007, Dixième Anniversaire de la Conférence Francophone sur les Approches Formelles dans l'Assistance au Développement de Logiciels (AFADL'07).

References in notes

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

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

[56] L. CASTELLANO, G. DE MICHELIS, L. POMELLO. *Concurrency versus Interleaving: An Instructive Example*, in "BEATCS: Bulletin of the European Association for Theoretical Computer Science", vol. 31, 1987.

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

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

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

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

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

[62] 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.

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

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