IN PARTNERSHIP WITH:
CNRS

Institut national des sciences appliquées de Rennes

Université Rennes 1

Activity Report 2011

Project-Team TRISKELL

Reliable and efficient component based software engineering

IN COLLABORATION WITH: Institut de recherche en informatique et systèmes aléatoires (IRISA)

RESEARCH CENTER
Rennes - Bretagne-Atlantique

THEME
Distributed Systems and Services

Table of contents

Project-Team TRISKELL

Keywords: Model-Driven Engineering, Adaptive Systems, Meta-modeling, Service Oriented Architecture, Software Engineering

1. Members

Research Scientist

Benoit Baudry [Junior Researcher Inria, HdR]

Faculty Members

Jean-Marc Jézéquel [Team Leader, Professor Université de Rennes 1, HdR]
Olivier Barais [Associate Professor Université de Rennes 1]
Arnaurd Blouin [INRIA grant until July 2011, Associate Professor INSA Rennes, from September 2010]
Johann Bourcier [Associate Professor Université de Rennes 1]
Benoit Combemale [Associate Professor Université de Rennes 1]
Naouel Moha [Associate Professor Université de Rennes 1]
Noël Plouzeau [Associate Professor Université de Rennes 1]
Gerson Sunyé [Associate Professor Université de Nantes, INRIA Delegation]

Technical Staff

Didier Vojtisek [Research engineer Inria]
Jean-Emile Dartois [INRIA ADT Daum, from October 2011]
Claire Diehl-Watrin [Inria (project ENTIMID), until July 2011]
Jacques Falcou [Inria (project CHESS)]
Marie Gouyette [Inria (project MOVIDA)]
An Phung-Khac [INRIA (project CESAR), from mid-October 2011]
Hajanirina Johary Rambelontsalama [Inria IJD, until April 2011]

PhD Students

Olivier-Nathanael Ben David [INRIA grant]
Juan-Jose Cadavid Gomez [INRIA grant]
Mickael Clavreul [INRIA grant]
Stephen Creff [CIFRE grant (with THALES)]
Erwan Daubert [INRIA grant]
Maha Driss [INRIA grant, Co-tutelle with U. Tunis]
Joao Bosco Ferreira-Filho [Université de Rennes 1 (project RELATE), from December 2011]
François Fouquet [MESR grant]
Clément Guy [MESR grant]
Aymerick Hervieu [CIFRE grant (with KEREVAL)]
Paul Istoan [Co-tutelle with U. Luxembourg]
Tam Le Nhan [Vietnam Grant]
Jonathan Marchand [ENS Cachan, from October 2011]
Antonio Mattos [Université de Rennes 1 (project RELATE), from December 2011]
Grégory Nain [INRIA grant]
Viet-Hoa Nguyen [INRIA Grant]
Suresh Pillay [Université de Rennes 1 (project RELATE), from October 2011]
Julien Richard-Foy [CIFRE grant (with Zenexity), from October 2011]
Emmanuelle Rouillé [CIFRE Grant (with SODIFRANCE)]
Nicolas Sannier [CIFRE Grant (with EDF)]
Hamza Samih [CIFRE Grant (with All4TECH), from October 2011]

Post-Doctoral Fellows

Cédric Bouhours [INRIA (project OPEES), engineer until August 2011]
Xavier Dolques [INRIA (project OPEES), engineer]

Administrative Assistant

Loic Lesage [TR Inria]

2. Overall Objectives

2.1. Introduction

Components, objects, contracts, aspects, models, meta-models, UML, MDE, software product lines, test, validation, requirements engineering, adaptive systems, services.

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 push towards reducing the duration of developments, and 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 model abstractions and reason on such assemblies of components. This is usually done by building models representing various aspects of a product line, such as functional variations, structural aspects (object paradigm), or dynamic aspects (languages of scenarios), without neglecting of course non-functional aspects like quality of service (performance, reliability, etc.) described in the form of contracts. Model Driven Engineering (MDE) is then a sub-domain of software engineering focusing on reinforcing design, validation and test methodologies based on the automatic processing of multi-dimensional models.

2.1.2. Project-team Presentation Overview

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

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 detailed design. It also involves building a continuum of tools which make use of these models, from model validation and verification, automatic application of design patterns, to test environments and on-line monitors supervising the behavior of the components in Dynamically Adaptable Systems. Since these modeling languages and associated tools appear quite open-ended and very domain specific, there is a growing need for *"tools for building tools for building software"*. Triskell is hence developing 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 developing *"tools for building tools for the sake of it"*, the Triskell project also has the goal of explicitly connecting its research results to industrial problems through collaborations with industry and technology transfer actions. This implies, in particular, taking into account the industrial standards of the field, namely the Eclipse Modeling Framework (EMF), the OMG's Meta-Object Facility (MOF) and Unified Modeling Language (UML), as well as domain specific component models such as OSGi.

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

2.2. Highlights

- Gerson Sunyé, Damien Pollet, Yves Le Traon and Jean-Marc Jézéquel received the Most Influential Paper Award of MODELS 2011, the 14th International Conference on Model Driven Engineering Languages and Systems. Their paper entitled "Refactoring UML Models", being published at the Models conference in 2001, has been selected as one of two papers to receive this award of a most influential paper after ten years [57].
- The Diva European project has been a real success as stated by the assesment of the final review: Excellent progress (the project has fully achieved its objectives and technical goals for the period and has even exceeded expectations).

3. Scientific Foundations

3.1. Model Driven Engineering for Distributed Software

3.1.1. 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.

3.1.2. 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 [52], [50]: 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 [54].

3.1.3. 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 [51]. 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.

3.1.4. Component

The object concept also provides the basis for *software components*, for which Szyperski's definition [58] 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 [53]).

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.

3.1.5. 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 [55] 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) [59]. 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) [49].

3.1.6. 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 [56], 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.

3.1.7. 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.

3.1.8. 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 made quite popular with an OMG initiative called the Model Driven Architecture (MDA). This requires that models are no longer informal, and that the weaving process is itself described as a program (which is as a matter of facts an executable meta-model) manipulating these models to produce a detailled design that can ultimately be transformed to code or at least test suites.

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

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

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

This wider context is now known as Model Driven Engineering.

4. Application Domains

4.1. Application Domains

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

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

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

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

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

5. Software

5.1. Kermeta

Participants: Didier Vojtisek [correspondant], Olivier Barais, Cédric Bouhours, Xavier Dolques, Jacques Falcou, François Fouquet, Marie Gouyette, Jean-Marc Jézéquel, Hajanirina Johary Rambelontsalama.

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. Integrated with the industrial standard Ecore and aligned with the OMG standard EMOF 2.0, the Kermeta language consists in a extension to these meta languages to support behavior definition. The language adds precise action specifications with static type checking and genericity at the meta level. Based on object-orientation and aspect orientation concepts, the Kermeta language adds model specific concepts. It 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 provides an integrated workbench based on Eclipse. It offers services such as : model execution, text editor (with syntax higlighting, code autocompletion), additional views and various import/export transformations.

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 proposes MDKs to work with the metamodels such as Java5, UML2, RDL (requirements), Ecore, Traceability,...

In 2011, Kermeta tooling has been refactored into a version 2.0.x in order to ease the integration of various MOF related languages in the tool chain. This new version also focuses on a fully compiled mode that allows to deploy kermeta programs in production environments.

See also the web page http://www.kermeta.org.

- APP: IDDN.FR.001.420009.000.S.P.2005.000.10400
- Version: 2.0.1
- Programming language: Java, Scala, Kermeta

5.2. Kevoree

Participants: Olivier Barais [correspondant], François Fouquet, Erwan Daubert, Johann Bourcier, Gregory Nain, Noël Plouzeau.

The Kevoree project [1] defines a framework dedicated to distributed systems design, using the *models at runtime* paradigm and a component-based software architecture approach. This framework offers a high-level abstraction for managing components and their interactions. It also provides concepts to describe the underlying infrastructure: resources, logical nodes and their topology.

Kevoree also provides a set of tools to manipulate model abstraction easily, relying in part on a Domain Specific Language (DSL) called KevScript. This DSL makes the architecture model modifications easier. Our DSL can also be used in a reasoning engine to dynamically adapt the running system by applying some changes at different level (SaaS, PaaS and IaaS). Kevoree has several runtime platform implementations, allowing execution of applications on various devices (*e.g.* JavaSeE, Android, μController such as Arduino, and cloud virtual nodes).

See also the web page http://www.kevoree.org.

- Version: 1.0
- Programming language: Java, Scala, Kermeta

5.3. Pramana

Participants: Benoit Baudry, Juan-Jose Cadavid Gomez, Benoit Combemale, Xavier Dolques, Hajanirina Johary Rambelontsalama, Didier Vojtisek [correspondant].

Pramana is an open-source tool, which automatically generates valid instances of a metamodel. These instances can then be used for analysis, verification, simulation or validation of the metamodel. The core mechanism for model generation relies on the bounded constraint-solver of Alloy, a lightweight model checker developed at the MIT. Alloy is integrated in Kermeta to allow the generation of instances of Ecore or Kermeta metamodels. Pramana implements this integration through a series of transformations and analysis, all implemented in Kermeta.
Metamodel instances can be used as input data for model transformation testing, and in particular for the testing of Kermeta code. For this purpose Pramana includes the K-Yeti module implementing a binding between the Kermeta language and the generic testing framework Yeti [2]. This module allows a Kermeta user to automatically run test cases.

See also the web page https://www.irisa.fr/triskell/Softwares/pramana.

- Version: 1.0
- Programming language: Java, Alloy, Kermeta

6. New Results

6.1. Model Driven and Aspect Oriented Design

6.1.1. Requirements Engineering

Participants: Olivier Barais, Benoit Baudry, Benoit Combemale, Maha Driss, Jean-Marc Jézéquel, Emmanuelle Rouillé, Nicolas Sannier, Didier Vojtisek.

[1] http://kevoree.org
[2] York Extensible Testing Infrastructure, Cf. http://www.yetitest.org/

Model-driven engineering can have a huge impact on the early design and analysis of complex systems. We have investigated modeling for requirements engineering in three areas:

- We use executable metamodeling techniques developed in the team to capture formal relationships between regulatory requirements and accepted practices in systems engineering [47], [42].
- We propose an approach for facilitating Web service selection according to user requirements. These requirements specify the needed functionality and expected QoS, as well as the composability between each pair of services. The originality of our approach is embodied in the use of Formal Concept Analysis (FCA) and its extension Relational Concept Analysis (RCA) [33] [25].
- We have analyzed a real industrial software process to illustrate the need for bridging the gap between software processes and software development tools to automate the development tools configuration, deployment, integration and adaptation [46].

6.1.2. Dynamically adaptive interactive systems

Participants: Arnaud Blouin, Jean-Marc Jézéquel, Grégory Nain.

Combining Aspect-Oriented Modeling with Property-Based Reasoning to Improve User Interface Adaptation: in this work we combined aspect-oriented modeling with an interactive system architecture to support dynamic adaptation of interactions and user interfaces [28].

6.1.3. Dynamically adaptive component-based systems

Participants: François Fouquet, Olivier Barais, Viet-Hoa Nguyen, Noël Plouzeau.

Continuous Design to Achieve Intelligent Reflection in Distributed Systems: we defined an intelligent reflection model to support fast adaptation of distributed systems by architecture modification without stopping the system. This adaptation mechanism is well suited to rapidly changing needs (continuous design of eternal systems) or fast paced modifications of the context of the running system (for instance for Internet of Things distributed systems) [34].

6.1.4. Architecture for Services-based applications

Participants: Olivier Barais, Johann Bourcier, Erwan Daubert, Jean-Marc Jézéquel.

The architecture of service-based applications can have a huge impact on their dynamic adaptability. We have investigated various framework for architecting service-based applications:

- Designing SAFDIS: a self adaptive framework for distributed applications based on services. SAFDIS includes facilities to support the coordination of distributed reconfigurations [24]. SAFDIS also takes benefit of the Infrastructure As A Service to dynamically reconfigure Software As A Service [44].
- Analyzing and improving consistency between functional and business view of telecom services architecture. This work is based on the definition of a strategic alignment of the target functional view with the target business view. Alignment is validated with a real case study implemented and deployed at Orange–France Telecom on their messaging service [22].
- Designing AutoHome : a service oriented framework to simplify the development and runtime adaptive support of autonomic pervasive applications. This includes the amalgamation of the two computing areas of Autonomics and Service Orientation, to produce a Component-based platform providing facilities. This infrastructure uniquely blends the advantages of distributed autonomic control with global conflict management in a management hierarchy [17].

6.2. Model V&V and Testing

6.2.1. Formal MDE Foundations

Participants: Benoit Baudry, Benoit Combemale.

- Formally Tracing Executions From an Analysis Tool Back to a Domain Specific Modeling Language's Operational Semantics: in this work, we propose a formal and operational framework for tracing results back (e.g., a program crash log, or a counterexample returned by a model checker) from execution and verification tools to an original DSML's syntax and operational semantics [31].
- A Proof Assistant Based Formalization of components in MDE: using the Coq proof assistant we propose a formalization of some operators for model fragment extraction and composition, as defined in the ReuseWare toolset [39].
- We have developed a methodology to explicitly model the context in which a temporal property must be verified. This contextual information is expressed in the requirements, and an explicit model allows to reduce the complexity of automated verification [41].

6.2.2. Pairwise testing for highly variable systems

Participants: Benoit Baudry, Aymeric Hervieu.

Variability management is a major concern for the development of software intensive systems. In particular, the explosion of variants is an issue for testing and analysis. Feature models allow to explicitly capture the variability in a formal model and get a complete view on all possible variants of the system. We have investigated pair-wise generation from feature models in order to test software product lines [36], and to evaluate QoS contracts in variable web service compositions [38].

6.2.3. Testing aspect-oriented programs

Participant: Benoit Baudry.

Aspect-oriented mechanisms introduce new risks for reliability that must be tackled by specific testing techniques in order to fully benefit from the use of this paradigm. We have investigated a monitoring mechanism of advices in an aspect-oriented program and use this information to build test cases that target faults in pointcut descriptors [18].

6.2.4. Modeling model quality metrics

Participants: Benoit Baudry, Jean-Marc Jézéquel.

We have developed a model-driven measurement approach to measure models of a domain specific modeling language. The approach uses models as unique and consistent metric specifications for the automated generation of a metric tool. The benefit from applying the approach is evaluated by four case studies [20]. In particular, we have evaluated the ability of the approach to build a tool for the measurement of requirements documents [21].

6.3. Meta-Modeling

6.3.1. Model Driven Language Engineering

Participants: Benoit Baudry, Arnaud Blouin, Juan-Jose Cadavid Gomez, Benoit Combemale, Clément Guy, Jean-Marc Jézéquel, Didier Vojtisek.

- Model-Driven Engineering and Optimizing Compilers: A bridge too far? In this work, we report and analyze an experience about the use of MDE technologies to build and evolve compiler infrastructures in the optimizing compiler domain. From this study, we highlight challenges and propose a roadmap for the cross-fertilization of the MDE and compiler domains [35], [45].

- Modeling Model Slicers: model slicing is a model operation that consists in extracting a subset of a model. Because the creation of a new DSL implies the creation from scratch of a new model slicer, we proposed the Kompren language that models and generates model slicers for any DSL [27].
- Empirical Evaluation of the Conjunct Use of MOF and OCL: we evaluate in this work the conjunct usage of MOF (Meta-Object Facility) and OCL (Object Constraint Language) in the development of Domain-Specific Modeling Languages. We observe the state of practice to understand how experts use them and find patterns on its usage, in order to provide techniques to improve the experience [29].

6.3.2. Model Transformation and Composition

Participants: Olivier Barais, Benoit Baudry, Arnaud Blouin, Mickaël Clavreul, Benoit Combemale, Xavier Dolques, Jean-Marc Jézéquel.

- Model operations such as transformation and composition declare source metamodels that are usually larger than the set of concepts and relations actually used by the operation. We have proposed and validated a static operation analyzer to retrieve the metamodel footprint of the operation [37]
- Service-Oriented Architecture Modeling: Bridging the Gap between Structure and Behavior: In this approach, we propose to detect divergences among structural and behavioral models to support a semi-automatic process of synchronization between class diagrams and workflow models [30].
- The paper propose a technique for discovering matchings between two model elements modeling the same system, but being instances of different metamodels. This is achieved by using property names and models structure thanks to the adaptation of a schema matching techique named Anchor-PROMPT [32].
- Specifying and implementing UI Data Bindings with Active Operations: based on the concept of active operations, this work proposes a framework to bind models at runtime and more precisely to bind data and their possible representations [26].
- We propose a requirement-centric approach for Web service composition which allows: (i) modeling users' requirements with the MAP formalism and specifying required services using an Intentional Service Model (ISM); (ii) discovering and selecting relevant Web services and high QoS services; and (iv) generating automatically BPEL coordination processes by applying the model transformation technique [19].

7. Contracts and Grants with Industry

7.1. Mopcom Ingénierie (Competitivity Cluster I&R)

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

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

In 2010, Triskell evaluates the proposed solution to easily integrate legacy systems with MDE in order to address the Thomson case study.

Project duration: 2008-2011 years

Triskell budget share: 150 keuros

Project Coordinator: Thales (TSA)

Participants: Thals Systmes Aroports, Thomson, Sodifrance, ENSIETA, INRIA, ENST Bretagne, Valoria, Orange Labs

7.2. ANR Movida

Participants: Olivier Barais, Jean-Marc Jézéquel.

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

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

Triskell mainly works this year on a tool to support software product line derivation for viewpoint modeling. We also works on the CVL standard. We present a tutorial on MOVIDA studio at the french conference IDM this year.

Project duration: 2009-2011

Triskell budget share: 184 keuros

Number of person/years: 1,2

Project Coordinator: Thales

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

7.3. Orange Labs

Participants: Jacques Simonin, Jean-Marc Jézéquel.

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

Project duration: 2006-2011

Triskell budget share: 25 keuros

7.4. EDF

Participants: Nicolas Sannier, Benoit Baudry.

Since October 2010, we have a collaboration with EDF R& D, Chatou. This project aims at investigating the application of metamodeling and model-driven engineering for modeling and analyzing requirement documents of control-command systems. The purpose of this modeling activity is to improve the global understanding of dependencies between requirements and their context and to use this knowledge for impact analysis in case of evolution. In this context, Benoit Baudry acts as Ph.D advisor for Nicolas Sannier.

Project duration: 2010-2013

Triskell budget share: 30 keuros

7.5. Kereval

Participants: Aymeric Hervieu, Benoit Baudry.

Since October 2010, we have a collaboration with Kereval, an SME specialized in software testing. In this project we investigate the selection and reuse of test cases for software product lines in the automotive domain. In this context, Benoit Baudry acts as Ph.D advisor for Aymeric Hervieu. Arnaud Gotlieb from the Celtique EPI acts as a co-advisor for the PhD, as well as Olivier Philippot from Kereval.

Project duration: 2010-2013

Triskell budget share: 15 keuros

7.6. Sodifrance

Participants: Emmanuelle Rouillé, Benoit Combemale, Olivier Barais, Jean-Marc Jézéquel.

Since October 2010, we have a collaboration with Sodifrance, Rennes. In this project we investigate the support (capitalization, definition, execution, and adaptation) of software processes in the context of model driven development (MDD). The purpose of this work is twofold:

- automate the tool configuration and the dynamic adaptation of MDD CASE tools.
- support an automated verification of models, according to the requirements for each activity of the process.

In this context, Jean-Marc Jézéquel acts as Ph.D advisor for Emmanuelle Rouillé, also supervised by Benoit Combemale and Olivier Barais.

Project duration: 2010-2013

Triskell budget share: 25 keuros

7.7. All4Tec

Participants: Hamza Sahmi, Benoit Baudry.

In this project with the All4Tec company we investigate the support of variability modelling for model-based test generation with Matelo (a tool developed by All4Tec).

In this context, Benoit Baudry acts as Ph.D advisor for Hamza Samih.

Project duration: 2011-2014

Triskell budget share: 20 keuros

7.8. All4Tec

Participants: Julien Richard-FOY, Olivier Barais, Jean-Marc Jezequel.

In this project with the Zenexity company we investigate the new architecture model for efficient web development on top of the play framework (a web framework developed by Zenexity).

In this context, Jean-Marc Jézéquel and Olivier Barais act as Ph.D advisor for Julien Richard Foy.

Project duration: 2011-2014

Triskell budget share: 20 keuros

8. Partnerships and Cooperations

8.1. Technology Development Actions (ADT)

8.1.1. DAUM

Participants: Didier Vojtisek, Jean-émile Dartois, François Fouquet, Erwan Daubert, Noël Plouzeau.

DAUM is a a Technology Development Action (ADT) by INRIA aiming at providing an integrated platform for distributed dynamically adaptable component based applications. DAUM unites and integrates results and software from the Triskell EPI and the Myriads EPI. More precisely, DAUM extends the Kevoree component framework designed by Triskell with adaptation mechanisms from the SAFDIS framework designed by Myriads.

DAUM will evaluate this integration by designing a full scale system for a tactical assistant for firefighter officers, in collaboration with the firefighters organization of Ille et Vilaine department (2800 firefighters).

Project duration: October 2011 - September 2012

Triskell budget share: One associated engineer shared with the Myriads EPI

Project Coordinator: Noël Plouzeau, Triskell INRIA Project.

Participants: Myriads, Triskell.

8.1.2. KerGekoz

Participants: Didier Vojtisek, Benoit Combemale, Olivier Barais, Clément Guy.

KerGekoz is a A Technology Development Action (ADT) by INRIA which goal is to improve the Gecos platform of Cairn EPI by applying MDE technologies from Triskell EPI.

Gecos platform is a compiler infrastructure for the conception *System on Chip*. Gecos integrates ASIP flow synthesis, automatic parallelisation and hardware synthesis (C to hardware).

This ADT focuses on

- consolidation of existing work,
- improvement of the reusability and maintainability by applying Kermeta MDE technologies.

Triskell EPI mainly works in collaboration with CAIRN to integrate Kermeta to the Gecos platform.

Project duration: 2010-2012

Triskell budget share: One associated engineer shared with CAIRN EPI

Project Coordinator: Steven Derrien, CAIRN INRIA Project.

Participants: CAIRN, TRISKELL.

8.2. Labex

8.2.1. Participation to Comin Labs

Participants: Johann Bourcier, Jean-Marc Jézéquel.

The Triskell project is involved in the Laboratory of excellence Comin Labs (Digital Communications and Informatics for the Future Internet) which involves various academics in French Brittany. The triskell team is mainly involves in the first challenge of Comin Labs : Digital Environment for the Citizen. Johann Bourcier has made an invited presentation about Software Engineering for Smart Cities at the first seminar of this challenge.

Project duration: 2011 - 2021

Participants: CNRS, Inserm, Université de Rennes 1, Université Rennes 2, Université de Bretagne Occidentale, Université de Bretagne Sud, Université de Nantes, Ecoles des Mines de Nantes, INSA de Rennes, ENS Cachan - antenne de Bretagne, Télécom Bretagne, Supelec, INRIA Rennes - Bretagne Atlantique.

8.3. National Initiatives

8.3.1. CNRS GDRs

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

- GDR GPL: Génie de la Programmation et du Logiciel (http://www-lsr.imag.fr/GPL), where Jean-Marc Jézéquel is a member of the scientific committee.
- Action IDM (on Model Driven Engineering) (http://www.actionidm.org), a transversal action (GDRs GPL, ASR and I3S).

The Triskell team also led an "Action Spécifique 2011 du GDR GPL" about software engineering for software intensive heterogeneous systems. Both the AOSTE and Triskell INRIA teams evolved in this project led by Benoit Combemale, and sharing a 5 keuros budget.

8.4. European Initiatives

8.4.1. 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

8.5. International Initiatives

8.5.1. Standardization in Eclipse projects

In 2011, Triskell project participates to the creation of Polarsys (A New Industry Collaboration to Build Open Source Tools for Safety-Critical Software Development) at the Eclipse Foundation to focus on building and maintaining tools for safety critical and embedded system development.

8.5.2. Standardization at OMG

In 2011, Triskell project participates to normalization actions at OMG (http://www.omg.org/). It was involved in the CVL Common Variability Language Response to RFP and was interested in the Analysis and Design group which promotes standard modeling techniques including UML and MOF.

8.5.3. Collaboration with foreign research groups

- University of Zürich Since 2010, Triskell has been working with the Requirements Engineering group on static analysis for model operations. Our work on metamodel footprint recovery has been published at ICSE [37]. We have started an empirical validation of this work with groups of students from Rennes and Zürich.
- University of Luxembourg. Since 2009 Triskell is involved in a collaborative project called SPLIT: Combine Software Product Line and Aspect-Oriented Software Development (with Nicolas Guelfi and Jacques Klein), that is funded by both the PICS program of CNRS and the FNR of Luxembourg. This project is providing the background and the funding for Paul Istoan's PhD thesis, done in co-tutelle between University of Rennes and University of Luxembourg. As an initial research result, we showed how aspects can be unwoven, based on a precise traceability metamodel dedicated to aspect model weaving.

8.6. European Initiatives

8.6.1. FP7 Projects

8.6.1.1. DIVA

Participants: Jean-Marc Jézéquel, Benoit Baudry, Olivier Barais, Didier Vojtisek, Johann Bourcier, Arnaud Blouin.

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

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

The DiVA project has ended during the year 2011, with the final review in May. This project has been a real success as stated by the assesment of the final review: Excellent progress (the project has fully achieved its objectives and technical goals for the period and has even exceeded expectations).

Project duration: 2007-2011

Triskell budget share: 400 keuros

Project Coordinator: SINTEF

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

8.6.1.2. S-CUBE

Title: S-CUBE

Type: COOPERATION (ICT)

Defi: Service & SW architectures, infrastructures and engineering

Instrument: Network of Excellence (NoE)

Duration: October 2008 - March 2012

Coordinator: University of Duisburg-Essen (Germany), Tilburg University (The Netherlands)

Others partners: Tilburg University (The Netherlands), City University London (UK), Consiglio Nazionale delle Ricerche (Italy), Center for Scientific and Technological Research, The French National Institute for Research in Computer Science and Control, Lero - The Irish Software Engineering Research Centre (Ireland), Politecnico di Milano (Italy), MTA SZTAKI - Computer and Automation Research Institute, Vienna University of Technology (Austria), Université Claude Bernard Lyon (France), University of Crete,Universidad Politécnica de Madrid (Spain), University of Stuttgart(Germany)

See also: http://www.s-cube-network.eu/

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

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

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

Triskell budget share: 150 keuros

8.6.1.3. NESSoS

Title: NESSoS

Type: COOPERATION (ICT)

Defi: Service & SW architectures, infrastructures and engineering

Instrument: Network of Excellence (NoE)

Duration: October 2010 - October 2014

Coordinator: CNR - Consiglio Nazionale delle Ricerche (Italy)

Others partners: ATOS (Spain), ETH (Switzerland), Katholieke Universiteit Leuven (Belgium), Ludwig-Maximilians-Universitaet Muenchen (Germany), IMDEA (Spain), INRIA (France), University of Duisburg-Essen (Germany), University of Malaga (Spain), University of Trento (Italy), SIEMENS (Germany), SINTEF (Norway)

See also: http://www.nessos-project.eu/

Abstract: The Network of Excellence on Engineering Secure Future Internet Software Services and Systems (NESSoS) aims at constituting and integrating a long lasting research community on engineering secure software-based services and systems. In light of the unique security requirements the Future Internet will expose, new results will be achieved by means of an integrated research, as to improve the necessary assurance level and to address risk and cost during the software development cycle in order to prioritize and manage investments. NESSoS will also impact training and education activities in Europe to grow a new generation of skilled researchers and practitioners in the area. NESSoS will collaborate with industrial stakeholders to improve the industry best practices and support a rapid growth of software-based service systems in the Future Internet.

Three INRIA EPIs are involved in NeSSoS: ARLES, CASSIS and Triskell. Triskell leads the research workpackage on design and architecture for secured future internet applications.

Triskell budget share: 100 keuros

8.6.1.4. CESAR

Title: CESAR

Duration: February 2009 - January 2012

Coordinator: AVL - GmbH (Austria)

See also: http://www.cesarproject.eu/

Abstract: In the context of CESAR, we have participated to the sub-project 3 demonstrator in order to demonstrate the usability of Polychrony as a co-simulation tool within the reference technology platform of the project, to which its open-source release has been integrated. The case-study, implemented in collaborateion with Airbus and IRIT, consists of co-modeling the doors management system of an Airbus A350 by merging its architecture description, specified with AADL, with its behavioral description, specified with Simulink.

Triskell brings its model-driven engineering expertise to compositionally assemble, compile and verify heterogeneous specifications (AADL and Simulink). Our case study will cover code generation for real-time simulation and test as well as formal verification both at system-level and in a GALS framework. Based on that case study, we aim at developing further modular code-generation services, real-time simulation, test and performance evaluation, formal verification as well as the validation of the generated concurrent and distributed code.

8.6.1.5. Artemis CHESS

Participants: Noël Plouzeau, Jean-Marc Jézéquel, Jacques Falcou, Viet-Hoa Nguyen.

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

In 2011 Triskell contributed to the definition and development of the model editor specially built for CHESS on top of Papyrus. Triskell is also a contributor of model transformation tools, by adapting its Kermeta platform to the Chess process, and by contributing to the interconnection of external tools from industrial tool provider partners. Triskell is also the implementor of a set of constraint checkers, which ensure that designers define models compliant with the CHESS metamodel.

Project duration: 2/2009-4/2012

Triskell budget share: 400 keuros

Project budget: 6 M euros

Project Coordinator: INTECS

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

8.6.2. Collaborations in European Programs, except FP7

Program: ITEA2

Project acronym: OPEES

Project title: Open Platform for the Engineering of Embedded Systems

Duration: 2010-2012

Triskell budget share: 150 keuros

Coordinator: OBEO (Gaël Blondelle)

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

Abstract: OPEES is an ITEA2 project which goal is to build a community able to ensure long-term availability of innovative engineering technologies in the domain of software-intensive embedded systems. Its main benefits should be to perpetuate the methods and tools for software development, minimize ownership costs, ensure independence of development platform, integrate, as soon as possible, methodological changes and advances made in academic world, be able to adapt tools to the process instead of the opposite, take into account qualification constraints. In this purpose, OPEES relies on the Eclipse Modeling Project platform (EMF, GEF, GMF, OCL, UML2, ...) and on many available tools such as Kermeta. The participation of Triskell into the OPEES project aims at industrializing both ModMap and Pramana. ModMap is a method and the associated tool to specify and use alignment rules between both homogeneous and heterogeneous languages. Current use is the creation of adapters between aligned languages. Pramana is a model transformation testing framework that makes it possible to synthesize input data (i.e. test models) for model transformations and check that the transformation behaves "correctly" on them.

Program: Marie Curie

Project acronym: Relate

Project title:Trans-European Research Training Network on Engineering and Provisioning of Service-Based Cloud Applications

Duration: February 2011 - January 2015

Triskell budget share: 730 keuros

Coordinator: Karlsruhe Institute of Technology

Other partners: Université de Rennes, IRISA (France); King’s College, (UK); South East European Research Center, SEERC (Greece); Charles University (Czech Republic); CAS Software (Germany); Singular Logic (Greece)

Abstract: The RELATE Initial Training Network aims to establish a network of international academic and industrial partners for a joint research training effort in the area of engineering and provisioning service-based cloud applications. The training is intended to not only shape high-level academic researchers, but also educate next generation experts and innovators in the European software industry. Through an integrative and multidisciplinary research approach, RELATE aims to promote the advancement of the state of the art in the related areas of model-driven engineering and formal methods, service-based mash-ups and application integration, security, performance, and trust in service-based cloud applications, and quality management and business model innovation.

8.7. International Initiatives

8.7.1. INRIA Associate Teams

8.7.1.1. MOCAA

Title: Models Composition, Aspects and Analysis

INRIA principal investigator: Benoît Baudry

International Partner:

Institution: Colorado State University (United States)

Laboratory: Colorado State University, Software Assurance Lab

Duration: 2006 - 2011

See also: http://www.irisa.fr/triskell/matt/

Computer-based systems have been growing in complexity at an exponential rate (roughly 10 fold increase every ten years) for more than 40 years. Like in other sciences, people have been relying more and more on modeling to try to master this complexity. Modeling, in the broadest sense, is indeed the cost-effective use of a simplified representation of an aspect of the world for a specific purpose. Because in software a model has the same nature as the thing it models, this opens the possibility to automatically derive software (and other artifacts such as test cases, performance profiles, or documentation) from its model. This property is well known from any compiler writer (and others), but it was recently be made quite popular with initiatives such as Model Integrated Computing (MIC) or OMG's Unified Modeling Language (UML) and Model Driven Architecture (MDA), globally known as Model Driven Development (MDD). In this context, models are formally described and can be automatically manipulated for refinement, composition, test case generation, documentation; All those operations are model transformations. This collaboration aims at better understanding how classical software engineering practices (design patterns, validation, methods, IDEs) can be adapted to develop model transformations. Clément Guy worked in collaboration with Prof. Robert B. France (from the software engineering domain), as well as with Prof. Sanjay Rajopadhye (from the optimizing compiler domain) to cross-fertilize both domains. In particular, he was studying the possibility to extend existing model typing to fit the needs of reusing model transformations.

8.7.2. INRIA-CONFAP

Title: Software Testing for Cloud Computing (TAAS)

International Partner:

Universidade Federal do Paraná.

Principal investigator: Gerson Sunyé

Duration: 2011 - 2012

Cloud computing is consolidating as an important paradigm for information technology to provide resources and Internet-based services. In clouds, a large amount of resources (e.g., memory, CPU, disk) is shared between several storage and processing machines or nodes, providing scalable environments.However, building reliable applications for clouds is a difficult task, because developers must face several non-trivial issues, such as: large-scale distribution, fault tolerance, massive data processing, hardware and software heterogeneity. In general, a cloud involves clusters and grids of nodes distributed over the Internet, where each new node shares its resources with the rest of the system, ensuring the scalability of clouds.
Since cloud applications are becoming ubiquitous in society's critical activities (health, economics, governments, etc.), they must ensure that the eventual failures of nodes do not affect the applications running on it. Large-scale distribution increases risks related to the loss of data because of nodes that fail, delay in computation times because of unreliable distribution strategy, etc. and several algorithms are proposed to increase their tolerance to faults. Thus, quality factors such as: reliability, robustness, availability and performance are essential. The main practice to ensure these factors, as well as the correctness, is the systematic use of testing during the different stages of development.
In this project, we propose to adapt and improve the testing architectures previously developed. More precisely, we propose to adapt the existing architecture for cloud environments, to define a testing language that supports the specification of large-scale tests as a whole and to provide both, a generator of test data and a fault injector, to reproduce real cloud environments.

8.7.3. INRIA International Partners

Following the Diva STREP project, we keep an active collaboration with the SINTEF institute. François Fouquet visited SINTEF for 8 weeks. During this visit, we combined the results of Kevoree and the result of the Moderate from SINTEF project to provide a dynamic component model for a micro-controllers based Internet of Things. Indeed, as the Internet of Things promises new ways for humans to interact with computing systems by seamlessly integrating resource constrained devices and traditional computing environment. These new computing environments are highly volatile and force applications to embed self-adaptive behaviors. The contribution of this collaboration is μ-Kevoree: a plain C implementation of the Kevoree runtime which can be deployed on poor in resources micro-controllers. Evaluation of memory usage, reliability and performance shows that μ-Kevoree is a viable solution with strong benefits over adaptation through dynamic firmware upgrades.

8.7.4. Visits of International Scientists

8.7.4.1. Internships

Hanen HAOUAS (from Mar 2011 until Aug 2011)

Subject: Autonomously Optimizing Service-Based Application Dependability in Smart Building

Institution: Ecole Nationale des Sciences de l'Informatique (Tunisia)

Wuliang Sun

Subject: Discovering the boundaries of a Modelling Space

Institution: Colorado State University (United States)

8.7.5. Participation In International Programs

Thanks to the MoCAA Equipe associée, Clément Guy realized a three-month stay in 2011 at Colorado State University (USA). He worked in collaboration with Prof. Robert B. France (from the software engineering domain), as well as with Prof. Sanjay Rajopadhye (from the optimizing compiler domain) to cross-fertilize both domains. In particular, he was studying the possibility to extend existing model typing to fit the needs of reusing model transformations.

9. Dissemination

9.1. Animation of the scientific community

9.1.1. Journals

9.1.1.1. Jean-Marc Jézéquel

is an Associate Editor of the following journals:

- IEEE Computer
- Journal on Software and System Modeling: SoSyM
- Journal of Systems and Software: JSS
- Journal of Object Technology: JOT

9.1.2. Examination Committees

9.1.2.1. Jean-Marc Jézéquel

was in the examination committee of the following PhD thesis and “Habilitation Diriger les Recherches”:

- Pierre Duquesne, January 2011, Université de Rennes 1 (president);
- Mariano Belaunde, January 2011, Université de Rennes 1 (adviser);
- Benoit Caillaud (HDR), March 2011, Université de Rennes 1 (president);
- Jannik Laval, June 2011, Université de Lille 1 (referee);
- Mathieu Acher, September 2011, Université de Nice (referee);
- Yves Le Gloahec, October 2011, Université de Bretagne Sud (referee);
- Hakim Hannousse, November 2011, Ecole des Mines de Nantes (member);
- Stéphane Lecomte, November 2011, Université de Rennes 1 (president);
- Grégory Nain, December 2011, Université de Rennes 1 (adviser);
- Maha Driss, December 2011, Université de Rennes 1 (adviser), co-tutelle with Tunisia;
- Mickael Clavreul, December 2011, Université de Rennes 1 (adviser);

9.1.2.2. Benoit Baudry

was in the examination committee of the following PhD thesis:

- Xavier Dumas, March 2011, Telecom Bretagne (member)
- Hakim Belhaouari, September 2011, Université de Pau et Pays de l’Adour (referee)
- Bastien Amar, October 2011, Université Paul Sabatier (referee)
- Ha Nguyen, October 2011, Université de Nantes (referee)
- Vincent Aranega, November 2011, Université de Lille 1 (referee)

9.1.2.3. Olivier Barais

was in the examination committee of the following PhD thesis

- Grégory Nain, December 2011, Université de Rennes 1 (co-adviser)
- Mickael Clavreul, December 2011, Université de Rennes 1 (co-adviser)

9.1.3. Conferences

9.1.3.1. Jean-Marc Jézéquel

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

- ECOOP 2011, Lancaster, UK, 25th - 29th July 2011
- SEAMS 2011 Software Engineering for Adaptive and Self-Managing Systems, 23-34 May 2011, Honolulu, Hawaii.
- SEFM 2011 The 9th International Conference on Software Engineering and Formal Methods, 14-18 November 2011, Montevideo, Uruguay.
- VaMoS 2011 Namur, Belgium, 24th - 26th January 2011

9.1.3.2. Benoit Baudry

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

- MODELS 2011 The 14th International Conference on Model Driven Engineering Languages and Systems Wellington, NZ, October 2011
- IEEE ICST'11 The 4th International Conference on Software Testing Verification and Validation, Berlin, Germany, March 2011
- SC'11, International Conference on Software Composition, Zurich, CH, June 2011
- TAP'11, 5th International Conference on Tests & Proofs, Zurich, CH, June 2011
- AGTIVE'11, Applications of Graph Transformation With Industrial Relevance, Budapest, Hungary, October 2011
- AST workshop on Automated Software Testing, at ICSE'11, Honolulu, USA, May 2011
- MoDeVVa 2011, 8th international workshop on Model design and Validation at MODELS'10, Wellington, NZ, October 2011
- Mutation'11 workshop at ICST'11, Berlin, Germany, March 2011
- CSTVA'11 workshop at ICST'11, Berlin, Germany, March 2011
- Revvert'11 workshop at ICST'11, Berlin, Germany, March 2011
- SEKE'11, International Conference on Software Engineering and Knowledge Engineering, Miami, USA, July 2011

9.1.3.3. Olivier Barais

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

- The 37th EUROMICRO Conference on Software Engineering and Advanced Applications SEAA2011, August 30 to September 2, 2011, Oulu, Finland
- 4nd Workshop on Context-aware Adaptation Mechanisms for Pervasive and Ubiquitous Services, (CAMPUS'11), 9th June 2011, Reykjavik, Iceland

9.1.3.4. Noël Plouzeau

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

- Component Based Software Engineering (CBSE), july 2011.

9.1.3.5. Benoit Combemale

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

- The Eighteenth Asia-Pacific Software Engineering Conference (APSEC 2011)
- The Industry Track of Software Language Engineering (ITSLE 2011)
- The Seventh European Conference on Modelling Foundations and Applications (ECMFA 2011)
- The Seventh Educators' Symposium @ MODELS 2011
- 7ièmes Journées sur l'Ingénierie Dirigée par les Modèles (IDM 2011)
- The MDE track of INFORSID 2011.

9.1.3.6. Gerson Sunyé

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

- The 23rd International Conference on Software Engineering and Knowledge Engineering(SEKE 2011).

9.1.3.7. Johann Bourcier

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

- The Workshop on Software Product Line Development in Dynamic Adaptive Environments at ECOOP 2011

9.1.4. Workshops, Tutorials and Keynotes

J.-M. Jézéquel gave an invited talk on Hyper-agility of Human-Computer Interactions at IHM 2011 (Conférence Francophone sur l'Interaction Homme-Machine), on Hyper-agility of SOA at the Dagstuhl seminar on Models.at.runtime, and an invited seminar at the University of Luxemburg..

Benoit Combemale gave two invited talks on Model Typing for Model Transformation Reuse and Model Validation & Verification at King's College London. He was invited to participate at the 2011 Bellairs CAMPaM (Computer Automated Multi-Paradigm Modelling) workshop (a workshop taking the Dagstuhl seminar format).

Benoit Baudry gave a keynote on mutation analysis from objects to the cloud at the Mutation workshop associated with ICST'2011. He also gave an invited talk at King's College London on Model Validation & Verification and at Université de Luxembourg on search-based software engineering for adaptive systems.

9.2. Miscellaneous

- J.-M. Jézéquel is Deputy Director of MATISSE Doctoral School. He is head of the Language and Software Engineering Department at Irisa. He is appointed to the board of the Committee of Projects of INRIA Rennes, and to the executive board of the Labex CominLabs. He is a member of the Steering Committee of the AOSD Conference series. He is a member of the Scientific Committee of the GDR GPL of CNRS. He belongs to the evaluation committee of the SIO division of DGA (Direction Générale de l'Armement). He is a Member of the Architecture Board of the MDDi Eclipse project. He participated to the creation of IFIP WG 10.2 on Embedded Systems. He is a member of the Advisory Board of the NSF REMODD Project (Repository for Model Driven Development).
- Benoit Baudry is on the steering committee of the IEEE International Conference on Software Testing Verification and Validation. He has been the local organizing chair for AOSD'10.

9.3. Teaching

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

Each of Jean-Marc Jézéquel, Noël Plouzeau, Olivier Barais, Benoit Combemale, Johann Bourcier and Arnaud Blouin teaches about 200h in these domains, with Benoit Baudry teaching about 50h, for a grand total of about 1400 hours, including several courses at ENSTB, Supelec and ENSAI Rennes.

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

Benoit Combemale has published and presented an article on teaching MDE at the 7th international Educators' Symposium at MODELS 2011 (the international symposium about software modeling in education) [43]. In this work, we report our experience on the dissemination in education of the research results in MDE. Benoit Combemale will be also co-chair of the 8th Educators' Symposium at MODELS 2012.

Jean-Marc Jézéquel, Benoit Combemale, and Didier Vojtisek have disseminated from research to teaching their experience in MDE by writing a book in french published by Ellipses [48].

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

PhD & HdR:

PhD: Mickael Clavreul, *Composition of models and meta-models: Separation of correspondences and interpretations for unifying existing model composition approaches*, Université Rennes1, 12/05/11, J.-M. Jézéquel and Olivier Barais [13]

PhD: Maha Driss, *Approche multi-perspective centrée exigences de composition de services Web*, Université Rennes1 and Université de Sfax, 12/08/11, J.-M. Jézéquel [14]

PhD: Grégory Nain, *EnTiMid: a component model to integrate smart devices in service based applications*, Université Rennes1, 12/07/11, J.-M. Jézéquel and Olivier Barais [15]

PhD: Mariano Belaunde, *Le Développement Agile de Services de Télécommunication Intégrés via des techniques d'ingénierie des modèles*, Université Rennes1, 20/01/11, J.-M. Jézéquel [12]

PhD in progress: Olivier Bendavid, *Security for Future Internet*, October 2010, Benoit Baudry and J.-M. Jézéquel

PhD in progress: Slim Benhassem, *Dynamic adaptation of multimodal interactions in the context of ubiquitous environments*, December 2008, J.-M. Jézéquel and Arnaud Blouin

PhD in progress: Juan-Jose Cadavid Gomez, *Assisting Metamodeling with Search Based Techniques*, 10/02/09, Benoit Baudry and J.-M. Jézéquel

PhD in progress: Stephen Creff, *Une variabilité multidimensionnelle pour une évolution incrémentale des lignes de produits dirigées par les Modèles*, May 2009, J.-M. Jézéquel

PhD in progress: Erwan Daubert, *Adaptation environmentale de services sur des plates-formes distribuées large échelle*, November 2009, Olivier Barais

PhD in progress: Joao Bosco Ferreira-Filho, *Variability Management in Model-driven Software and System Engineering*, October 2011, Benoit Baudry and Olivier Barais

PhD in progress: François Fouquet, *Un modèle pour les systèmes distribués dynamiquement adaptables*, October 2009, J.-M. Jézéquel and Noël Plouzeau

PhD in progress: Clément Guy, *Generic Definition of Domain Specific Analysis using Model-Driven Engineering (MDE)*, October 2010, J.-M. Jézéquel and Benoit Combemale

PhD in progress: Aymeric Hervieu, *Génération de test sur les lignes de produits logicielles*, October 2010, Benoit Baudry

PhD in progress: Paul Istoan, *Methodology for the derivation of product behavior in a Software Product Line*, November 2009, J.-M. Jézéquel

PhD in progress: Tam Le Nhan, *Model-Driven Software Engineering for Cloud Computing*, October 2010, J.-M. Jézéquel and Gerson Sunyé

PhD in progress: Jonathan Marchand, *Engineering Semantics in Modeling Languages*, October 2011, Benoit Baudry and Benoit Combemale

PhD in progress: Antonio Mattos, *Synthesis of Service-oriented Architectures by Model Transformation* , December 2011, Noël Plouzeau and Olivier Barais

PhD in progress: Viet Hoa Nguyen, *Evaluation of Stochastic Properties of Dynamic Component Based Systems*, December 2009, J.-M. Jézéquel and Noël Plouzeau

PhD in progress: Suresh Pillay, *Validating Web Service Composition*, October 2011, Benoit Baudry and Benoit Combemale

PhD in progress: Julien Richard-Foy, *A DSL Factory for Modular Web Components*, October 2011, J.-M. Jézéquel and Olivier Barais

PhD in progress: Emmanuelle Rouillé, *Intentional-Driven Development Process*, October 2010, J.-M. Jézéquel and Benoit Combemale

PhD in progress: Hamza Samih, *Extension du model-based testing pour la prise en compre de la variabilité et la sécurité dans les systèmes complexes*, November 2011, Benoit Baudry

PhD in progress: Nicolas Sannier, *Ingénierie des modèles pour la gestion des référentiels d'exigences de sûreté pour les systèmes de contrôle-commande*, October 2010, Benoit Baudry

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", March 2005, vol. 22, n^{o} 2, p. 76–82.

[2] B. BAUDRY, S. GHOSH, F. FLEUREY, R. FRANCE, Y. LE TRAON, J.-M. MOTTU. *Barriers to Systematic Model Transformation Testing*, in "Communications of the ACM journal", Jun 2010, vol. 53, n^{o} 6, 10 pages, http://hal.inria.fr/inria-00542747_v1.

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

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

[5] J. KLEIN, F. FLEUREY, J.-M. JÉZÉQUEL. *Weaving Multiple Aspects in Sequence Diagrams*, in "Transactions on Aspect-Oriented Software Development (TAOSD)", 2007, vol. LNCS 4620, p. 167-199, http://www.irisa.fr/triskell/publis/2007/Klein07a.pdf.

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

[7] B. MORIN, O. BARAIS, J.-M. JÉZÉQUEL, F. FLEUREY, A. SOLBERG. *Models at Runtime to Support Dynamic Adaptation*, in "IEEE Computer", October 2009, p. 46-53, http://www.irisa.fr/triskell/publis/2009/Morin09f.pdf.

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

[9] J. STEEL, J.-M. JÉZÉQUEL. *On Model Typing*, in "Journal of Software and Systems Modeling (SoSyM)", December 2007, vol. 6, n^{o} 4, p. 401–414, http://www.irisa.fr/triskell/publis/2007/Steel07a.pdf.

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

[11] T. ZIADI, J.-M. JÉZÉQUEL. *Product Line Engineering with the UML: Deriving Products*, in "Software Product Lines - Research Issues in Engineering and Management", T. KÄKÖLÄ, J. C. DUEÑAS (editors), Springer Verlag, 2006, n^{o} ISBN: 978-3-540-33252-7, p. 557–586.

Publications of the year

Doctoral Dissertations and Habilitation Theses

[12] M. BELAUNDE. *Le Développement Agile de Services de Télécommunication Intégrés via des techniques d'ingénierie des modèles.*, Université Rennes 1, January 2011, http://tel.archives-ouvertes.fr/tel-00650682.

[13] M. CLAVREUL. *Composition de modèles et de métamodèles : Séparation des correspondances et des interprétations pour unifier les approches de composition existantes*, Université Rennes 1, December 2011, final draft, http://hal.inria.fr/tel-00646893/en.

[14] M. DRISS. *Approche multi-perspective centrée exigences de composition de services Web*, Université Rennes 1, December 2011, http://hal.inria.fr/tel-00648131/en.

[15] G. NAIN. *EnTiMid : Un modèle de composants pour intégrer des objets communicants dans des applications à base de services*, Université Rennes 1, December 2011, http://hal.inria.fr/tel-00646664/en.

Articles in International Peer-Reviewed Journal

[16] B. BAUDRY, G. FRASER, J. BRADBURY. *Guest Editorial for Special Section on Mutation Testing.*, in "Information and Software Technology", October 2011, http://hal.inria.fr/hal-00641272/en.

[17] J. BOURCIER, A. DIACONESCU, P. LALANDA, M. JULIE. *AutoHome: an Autonomic Management Framework for Pervasive Home Applications*, in "ACM Transactions on Autonomous and Adaptive Systems (TAAS)", February 2011, vol. 6, n^{o} 1, http://hal.inria.fr/inria-00554197/en.

[18] R. DELAMARE, B. BAUDRY, S. GHOSH, S. GUPTA, Y. LE TRAON. *An Approach for Testing Pointcut Descriptors in AspectJ*, in "Software Testing, Verification and Reliability", April 2011, AOP, test oracle, mutation analysis, test framework, http://hal.inria.fr/hal-00641263/en.

[19] M. DRISS, Y. JAMOUSSI, N. MOHA, J.-M. JÉZÉQUEL, H. HAJJAMI BEN GHÉZALA. *Une approche centrée exigences pour la composition de services web*, in "Ingenierie des Systemes d Information", May 2011, vol. 16, n^{o} 2, p. 97-125, http://hal.inria.fr/hal-00648159/en.

[20] M. MONPERRUS, B. BAUDRY, J.-M. JÉZÉQUEL, J. CHAMPEAU, B. HOELTZENER. *Automated Measurement of Models of Requirements*, in "Software Quality Journal", 2011, WOS [*DOI :* 10.1007/s11219-011-9163-6], http://hal.inria.fr/hal-00646876/en.

[21] M. MONPERRUS, J.-M. JÉZÉQUEL, B. BAUDRY, J. CHAMPEAU, B. HOELTZENER. *Model-driven Generative Development of Measurement Software*, in "Software and Systems Modeling (SoSyM)", 2011, vol. 10, n^{o} 4, p. 537-552, WOS, http://hal.inria.fr/inria-00504670/en.

[22] J. SIMONIN, E. BERTIN, Y. LE TRAON, J.-M. JÉZÉQUEL, N. CRESPI. *Analysis and improvement of the alignment between business and information system for telecom services*, in "International Journal On Advances in Software", 2011, vol. 4, n^{o} 1 & 2, p. 117-128, http://hal.inria.fr/hal-00631541/en.

Invited Conferences

[23] X. DOLQUES, M. HUCHARD, C. NEBUT, H. SAADA. *Formal and Relational Concept Analysis approaches in Software Engineering: an overview and an application to learn model transformation patterns in examples*,

in "ICESE'11: First ICESE Virtual Workshop, Search-based Model-Driven Engineering", Qatar, May 2011, http://hal.inria.fr/lirmm-00616272/en.

International Conferences with Proceedings

[24] F. ANDRÉ, E. DAUBERT, G. GAUVRIT. *Distribution and Self-Adaptation of a Framework for Dynamic Adaptation of Services*, in "The Sixth International Conference on Internet and Web Applications and Services (ICIW)", St. Maarten, Netherlands Antilles, IARIA, March 2011, p. 16-21, http://hal.inria.fr/inria-00553539/en.

[25] Z. AZMEH, M. DRISS, F. HAMOUI, M. HUCHARD, N. MOHA, C. TIBERMACINE. *Selection of Composable Web Services Driven by User Requirements*, in "ICWS'11: 9th IEEE International Conference on Web Services - Applications and Experiences Track", Washington DC, United States, IEEE Computer Society, July 2011, 8, http://hal.inria.fr/lirmm-00596346/en.

[26] O. BEAUDOUX, A. BLOUIN, O. BARAIS, J.-M. JÉZÉQUEL. *Specifying and implementing UI Data Bindings with Active Operations*, in "ACM SIGCHI Symposium on Engineering Interactive Computing Systems", Pise, Italy, June 2011, p. 127–136 [*DOI :* 10.1145/1996461.1996506], http://hal.inria.fr/inria-00590896/en.

[27] A. BLOUIN, B. COMBEMALE, B. BAUDRY, O. BEAUDOUX. *Modeling Model Slicers*, in "ACM/IEEE 14th International Conference on Model Driven Engineering Languages and Systems", Wellington, New Zealand, Springer Berlin / Heidelberg, October 2011, vol. 6981, p. 62–76 [*DOI :* 10.1007/978-3-642-24485-8_6], http://hal.inria.fr/inria-00609072/en.

[28] A. BLOUIN, B. MORIN, O. BEAUDOUX, G. NAIN, P. ALBERS, J.-M. JÉZÉQUEL. *Combining Aspect-Oriented Modeling with Property-Based Reasoning to Improve User Interface Adaptation*, in "ACM SIGCHI Symposium on Engineering Interactive Computing Systems", Pise, Italy, June 2011, p. 85–94 [*DOI :* 10.1145/1996461.1996500], http://hal.inria.fr/inria-00590891/en.

[29] J. CADAVID, B. BAUDRY, B. COMBEMALE. *Empirical evaluation of the conjunct use of MOF and OCL*, in "Experiences and Empirical Studies in Software Modelling (EESSMod 2011)", Wellington, New Zealand, M. CHAUDRON, M. GENERO, P. MOHAGHEGHI, L. PARETO (editors), CEUR, October 2011, http://hal.inria.fr/hal-00647347/en.

[30] M. CLAVREUL, S. MOSSER, M. BLAY-FORNARINO, R. B. FRANCE. *Service-Oriented Architecture Modeling: Bridging the Gap between Structure and Behavior*, in "Model Driven Engineering Languages and Systems", Wellington, New Zealand, J. WHITTLE, T. CLARK, T. KÜHNE (editors), Lecture Notes in Computer Science, Springer Berlin / Heidelberg, 2011, vol. 6981, p. 289-303 [*DOI :* 10.1007/978-3-642-24485-8_21], http://hal.inria.fr/inria-00634943/en.

[31] B. COMBEMALE, L. GONNORD, V. RUSU. *A Generic Tool for Tracing Executions Back to a DSML's Operational Semantics*, in "Seventh European Conference on Modelling Foundations and Applications", Birmingham, United Kingdom, Lecture Notes in Computer Science, Springer Verlag, June 2011, vol. 6698, p. 35-51, http://hal.inria.fr/hal-00593425/en.

[32] X. DOLQUES, A. DOGUI, J.-R. FALLERI, M. HUCHARD, C. NEBUT, F. PFISTER. *Easing Model Transformation Learning with Automatically Aligned Examples*, in "ECMFA'11: 7th European Conference Modelling – Foundation and Applications", Birmingham, UK, France, R. B. FRANCE, J. M. KÜSTER, B. BORDBAR, R. F. PAIGE (editors), LNCS: Lecture Notes in Computer Science, Springer, June 2011, vol. 6698, p. 189-204 [*DOI :* 10.1007/978-3-642-21470-7_14], http://hal.inria.fr/lirmm-00616271/en.

[33] M. DRISS, Y. JAMOUSSI, J.-M. JÉZÉQUEL, H. HAJJAMI BEN GHÉZALA. *A Multi-Perspective Approach for Web Service Composition*, in "iiWAS 2011", Ho Chi Minh City, Viet Nam, December 2011, http://hal.inria.fr/hal-00648171/en.

[34] F. FLEUREY, B. MORIN, A. SOLBERG, O. BARAIS. *MDE to Manage Communications with and between Resource-Constrained Systems*, in "Model Driven Engineering Languages and Systems", Wellington, Nouvelle-Zélande, October 2011, vol. 6981/2011, 16, http://hal.archives-ouvertes.fr/hal-00647141.

[35] A. FLOCH, T. YUKI, C. GUY, S. DERRIEN, B. COMBEMALE, S. RAJOPADHYE, R. FRANCE. *Model-Driven Engineering and Optimizing Compilers: A bridge too far?*, in "International Conference on Model Driven Engineering Languages and Systems", Wellington, New Zealand, October 2011, http://hal.inria.fr/inria-00613575/en.

[36] A. HERVIEU, B. BAUDRY, A. GOTLIEB. *PACOGEN : Automatic Generation of Pairwise Test Configurations from Feature Models*, in "ISSRE", Hiroshima, Japan, December 2011, http://hal.inria.fr/hal-00641111/en.

[37] C. JEANNERET, M. GLINZ, B. BAUDRY. *Estimating Footprints of Model Operations*, in "International Conference on Software Engineering", Honolulu, United States, May 2011, http://hal.inria.fr/hal-00641091/en.

[38] A. KATTEPUR, S. SEN, B. BAUDRY, A. BENVENISTE, C. JARD. *Pairwise Testing of Dynamic Composite Services*, in "The 6th international symposium on Software engineering for adaptive and self-managing systems", Waikiki, Honolulu, Hawaii, United States, 1988028, ACM, 2011, p. 138–147 [*DOI :* 10.1145/1988008.1988028], http://hal.inria.fr/hal-00641340/en.

[39] M. KEZADRI, B. COMBEMALE, M. PANTEL, X. THIRIOUX. *A Proof Assistant Based Formalization of components in MDE*, in "8th International Symposium on Formal Aspects of Component Software (FACS 2011)", Oslo, Norway, University of Oslo, Norway, September 2011, http://hal.inria.fr/inria-00614593/en.

[40] T. MOUELHI, Y. LE TRAON, E. ABGRALL, B. BAUDRY, S. GOMBAULT. *Tailored Shielding and Bypass Testing of Web Applications*, in "International Conference on Software Testing Verification and Validation", Berlin, Germany, March 2011, http://hal.inria.fr/hal-00646424/en.

[41] A. RAJI, P. DHAUSSY, B. BAUDRY. *Formal Extension of Use Cases for Context-Aware Verification*, in "MoDeVVA 2011 - Model-Driven Engineering, Verification, and Validation: Integrating Verification and Validation in MDE", Wellington, New Zealand, October 2011, xx, http://hal.inria.fr/hal-00635555/en.

[42] N. SANNIER, B. BAUDRY, T. NGUYEN. *Formalizing standards and regulations variability in longlife projects. A challenge for Model-driven engineering*, in "Model-Driven Requirements Engineering Workshop (MoDRE)", Trento, Italy, August 2011 [*DOI :* 10.1109/MODRE.2011.6045368], http://hal.inria.fr/inria-00636855/en.

Conferences without Proceedings

[43] B. COMBEMALE, X. CRÉGUT, A. DIEUMEGARD, M. PANTEL, F. ZALILA. *Teaching MDE through the Formal Verification of Process Models*, in "7th Educators' Symposium @ MODELS 2011: Software Modeling in Education (EduSymp2011)", Wellington, New Zealand, ECEASST (editor), Marion Brandsteidl; Andreas Winter, July 2011, http://hal.inria.fr/hal-00646426/en.

[44] E. DAUBERT, F. ANDRÉ, O. BARAIS. *Adaptation multi-niveaux : l'infrastructure au service des applications*, in "Conférence Française en Systèmes d'Exploitation (CFSE)", St Malo, France, May 2011, http://hal.inria.fr/inria-00603973/en.

[45] C. GUY, S. DERRIEN, B. COMBEMALE, J.-M. JÉZÉQUEL. *Vers un rapprochement de l'IDM et de la compilation*, in "Journées sur l'Ingénierie Dirigée par les Modèles", Lille, France, June 2011, http://hal.inria.fr/inria-00601670/en.

[46] E. ROUILLÉ, B. COMBEMALE, O. BARAIS, T. DAVID, J.-M. JÉZÉQUEL. *Bridging the Gap Between Software Process and Software Development*, in "IDM", Lille, France, June 2011, http://hal.inria.fr/inria-00605465/en.

[47] N. SANNIER, B. BAUDRY. *Défis pour la variabilité et la traçabilité des exigences en ingénierie système*, in "INFORSID 2011", Lille, France, May 2011, http://hal.inria.fr/inria-00598668/en.

Scientific Books (or Scientific Book chapters)

[48] J.-M. JÉZÉQUEL, B. COMBEMALE, D. VOJTISEK. *Ingénierie Dirigée par les Modèles : des concepts à la pratique...*, Ellipse, 2011, http://hal.inria.fr/hal-00648489/en.

References in notes

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

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

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

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

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

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

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

[56] 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.

[57] G. SUNYÉ, D. POLLET, Y. LE TRAON, J.-M. JÉZÉQUEL. *Refactoring UML Models*, in "Proceedings of UML 2001", LNCS, Springer Verlag, 2001, vol. 2185, p. 134–148.

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

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