IN PARTNERSHIP WITH:
Institut national des sciences appliquées de Rennes

Université Rennes 1

Activity Report 2012

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 programming and Software engineering

Table of contents

Project-Team TRISKELL

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

Creation of the Project-Team: 2001 November 01.

1. Members

Research Scientist
Benoit Baudry [Team Leader, Researcher Inria, HdR]

Faculty Members
Jean-Marc Jézéquel [Professor Université de Rennes 1, HdR]
Mathieu Acher [Associate Professor Université de Rennes 1, from September 2012]
Olivier Barais [Associate Professor Université de Rennes 1]
Arnaud Blouin [Associate Professor INSA Rennes]
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]

Engineers
Didier Vojtisek [Research engineer Inria]
Jean-Émile Dartois [Inria ADT Daum]
Jacques Falcou [Inria (project CHESS), until April 2012]
Marie Gouyette [Inria (project MOVIDA), until January 2012]
An Phung-Khac [Inria (project CESAR)]

PhD Students
Olivier-Nathanael Ben David [Inria grant]
Jacky Bourgeois [Co-tutelle with Open University (UK), from October 2012]
Erwan Bousse [MENRT grant, from October 2012]
Juan-Jose Cadavid Gomez [Inria grant]
Stephen Creff [CIFRE grant (with THALES)]
Erwan Daubert [Inria grant]
Joao Bosco Ferreira-Filho [Université de Rennes 1 (project RELATE)]
François Fouquet [MESR grant]
Inti Gonzalez Herreira [Inria (project InfraJVM), from October 2012]
Clément Guy [MESR grant]
Aymeric Hervieu [CIFRE grant (with KEREVAL)]
Paul Istoan [Co-tutelle with U. Luxembourg]
Tam Le Nhan [Vietnam Grant]
Valéria Lelli Leitão Dantas [Inria (project Connexion), from November 2012]
Jonathan Marchand [ENS Cachan, until August 2012]
Antonio Mattos [Université de Rennes 1 (project RELATE)]
Viet-Hoa Nguyen [Inria Grant]
Suresh Pillay [Université de Rennes 1 (project RELATE)]
Julien Richard-Foy [CIFRE grant (with Zenexity)]
Emmanuelle Rouillé [CIFRE Grant (with SODIFRANCE)]
Nicolas Sannier [CIFRE Grant (with EDF)]
Hamza Samih [CIFRE Grant (with All4TECH)]

Post-Doctoral Fellows
Marco Biazzini [Université de Rennes 1, from May 2012]
Fabio Buiati [Inria, from June 2012]

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 of the Year

The evaluation seminar has been the main event for the team in 2012. We believe it is a highlight, since we have been evaluated on all our results and activities on the 2008 - 2012 period, and the feedback from the reviewers panel is very positive. They have emphasized the high quality of our results and also encouraged to pursue our perspectives of software engineering for open systems.

3. Research Program

3.1. Model Driven Engineering for Distributed Software

Objects, design patterns, software components, contracts, aspects, models, UML, product lines

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 [76], [71]: 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 [79].

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 [75]. 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 [86] 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 [78]).

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 [80] 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) [87]. 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) [69].

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 [84], 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

SOA, telecommunication, distributed systems, Embedded Systems, software engineering, test, UML

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 and Platforms

5.1. Kermeta

Participants: Didier Vojtisek [correspondant], Olivier Barais, Arnaud Blouin, Benoit Combemale, Jacques Falcou, François Fouquet, Marie Gouyette, Clément Guy, Jean-Marc Jézéquel, Jonathan Marchand.

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 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

Main competitors:

- XMF-Mosaic is developed by Ceteva and is now open-source since 2008.
- GME is a large scale Meta-Modeling Environment developed at Vanderbilt University (ISIS project) since 2002.
- MOFLON is a Metamodeling Framework with Graph Transformations, developed by A. Schuerr's group (TU-Darmstadt) since 2008.
- XCore is a recent (2011) Eclipse project supported by Itemis/Macro Modelling that provides a single operational surface syntax for Ecore.
- Many QVT inspired model transformation tools focused on model transformations.

Main innovative features:

Kermeta was one of the first solutions to offer an operational semantics on top of EMOF. It still proposes several unique features that cannot be found in the tools presented above, such as:

- aspect weaving at the metamodel level allows fast prototyping of a wide variety of tools;
- model typing allows a safe reuse of algorithms and transformations accross diffrent metamodels.

5.2. Kevoree

Participants: Olivier Barais [correspondant], François Fouquet, Erwan Daubert, Jean-Émile Dartois, Johann Bourcier, Antonio Mattos, Noël Plouzeau.

Kevoree is an open-source models@runtime platform [1] to properly support the dynamic adaptation of distributed systems. Models@runtime basically pushes the idea of reflection [82] one step further by considering the reflection layer as a real model that can be uncoupled from the running architecture (e.g. for reasoning, validation, and simulation purposes) and later automatically resynchronized with its running instance.

Kevoree has been influenced by previous work that we carried out in the DiVA project [82] and the Entimid project [83]. With Kevoree we push our vision of models@runtime [81] farther. In particular, Kevoree provides a proper support for distributed models@runtime. To this aim we introduced the *Node* concept to model the infrastructure topology and the *Group* concept to model semantics of inter node communication during synchronization of the reflection model among nodes. Kevoree includes a *Channel* concept to allow for multiple communication semantics between remote*Components* deployed on heterogeneous nodes. All Kevoree concepts (Component, Channel, Node, Group) obey the object type design pattern to separate deployment artifacts from running artifacts. Kevoree supports multiple kinds of very different execution node technology (e.g. Java, Android, MiniCloud, FreeBSD, Arduino, ...).

Kevoree is distributed under the terms of the LGPL open source license.

Main competitors:

- the Fractal/Frascati eco-system [2].
- SpringSource Dynamic Module [3]
- GCM-Proactive [4]
- OSGi [5]
- Chef [6]
- Vagran [7]

[1] http://www.kevoree.org
[2] http://frascati.ow2.org
[3] http://www.springsource.org/osgi/
[4] http://proactive.inria.fr/
[5] http://www.osgi.org
[6] http://wiki.opscode.com/display/chef/Deploy+Resource

Main innovative features:

- distributed models@runtime platform (with a distributed reflection model and an extensible models@runtime dissemination set of strategies).
- Support for heterogeneous node type (from Cyber Physical System with few resources until cloud computing infrastructure).
- Fully automated provisioning model to correctly deploy software modules and their dependencies.
- Communication and concurrency access between software modules expressed at the model level (not in the module implementation).

Impact: Several European projects leveraging the Kevoree platform have recently been accepted. Besides we are currently developing a testbed named DAUM. This testbed is developed since mid 2011 to experiment with Kevoree in real life situations. More precisely, DAUM is a highly dynamic pervasive system that mixes wireless smart sensors, user interaction devices such as digital pads, and distributed data servers in a cloud. The current specialization of DAUM is a distributed tactical information and decision system for firefighters. This application includes individual sensors in the personal protective equipment of firefighters, embedded computation nodes that are fully reconfigurable in real time and over the air, distributed monitoring servers in trucks, and personal computers for information access and decision making. The DAUM platform is used internally to try research results on distributed *models@runtime*. DAUM is used externally to prepare and support cooperation activities with other research teams (the Myriads Inria team is a partner of DAUM) and with potential industrial partners.

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

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

6. New Results

6.1. Distributed models at runtime

In the last two years we have developed a new models@runtime approach, named Kevoree. It supports extensive architecture evolution at runtime and enables the design of eternal systems with a continuous design process. The Kevoree type model supports dynamic types redefinition, allowing for complete redesign of specifications and implementations while the system is running. Communication channels between components are themselves first class dynamic entities. By combining our component metamodel and a *models@runtime* approach we have developed implementations of Kevoree for a wide range of computation nodes, ranging from inexpensive embedded microcontrollers to large commercial cloud implementations. We have shown that **applications based on the Kevoree component model are able to reconfigure their architecture completely on the fly** several times per second [40] on computation nodes with very limited resources.

Using the Kevoree platform, we demonstrated the use of *models@runtime* for large-scale distributed systems. We have shown that the *models@runtime* approach is applicable to pervasive distributed systems, even with volatile networks and continuously changing topologies [41]. Using *ad hoc* distributed algorithms, architectural models are propagated reliably in spite of frequent loss of connectivity, and **reconfigurations of a distributed application are managed in a continuous consistent manner**. Using colored Petri nets to describe quantititative properties we are building a toolchain to estimate the time related properties of assemblies at runtime [51].

[7] http://vagrantup.com/

6.2. Real scale platform for dynamic tactical decision system

Since mid 2011 the Triskell team is designing and implementing the DAUM platform that integrates a large range of technologies, ranging from wireless low cost sensors to clouds made of rugged field miniservers. Our application use case is a tactical decision system designed in cooperation with a large firefighter department of 3,500 firefighters. This platform is being used as a real life testbed for our results on dynamic, continuous design of distributed pervasive systems. It is also used as a concrete cooperation support within the Marie Curie Initial Training Network *Relate*.

By combining *models@runtime* techniques and component-based techniques, we have shown how we can apply model driven engineering to design large-scale, distributed, heterogeneous and adaptive systems [40].

6.3. Software Language Engineering

With the growing interest in MDE, more and more models are used during a software development to capture various aspects (both functional and extra-functional). Therefore, explicitly identifying and analyzing these relationships becomes a real challenge during a model-based software development. To address this challenge, we proposed a **formal language that captures relations between modeled things in order to reason and communicate about modeling activities** [19].

More recently, we started to explore the necessary breakthrough in software languages to support a global software engineering. Consequently, we investigate MDE-based tools and methods in software language engineering (SLE) for the design and implementation of collaborative, interoperable and composable modeling languages [32], [31], [30].

6.4. Model Typing

In recent years, the Triskell team established a formal theory of model typing, considering models as first class entities when modeling in the large [8]. Model typing was initially developed to support the reuse of both metamodels and model transformations [21]. It is now becoming the cornerstone of the various established metamodeling operators to ensure structural and behavioral properties [85][43].

The series of work on model typing was initially developed in the context of Jim Steel's PhD, defended in 2008. Then, it has continuously evolved in the scheme of the Naouel Moha's post doctoral position and the Clément Guy's PhD thesis [43]. Recently, work on model typing had a very strong application to the field of optimizing compilers [18]. This is the result of a close collaboration between Inria and Colorado State University (CSU), involving two teams in MDE (the Triskell team at Inria and the SE group at CSU), and two teams in optimizing compilers (the CAIRN team at Inria and the Mélange group at CSU). This collaboration was partially funded by the Inria associated teams MoCAA and LRS.

6.5. Model Footprint / Pruning / Slicing

During the previous evaluation period, we have established various facilities to ease the metamodeling activity.

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 [46]. Then, we propose a conjunct use of model typing and metamodel pruning to ease the reuse of model transformations on instances of different metamodels [21].

In general, many operators consist into extracting a subset of a model according to a language-based specification. 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 [70][66]. An extended version was recently published in SoSyM [14].

[8] Model typing goes beyond the typing of individual model elements to actually deal with the type of graphs of model elements

6.6. Model Composition

Triskell hence contributed to the software engineering community's effort to propose new ways of composing software from modeling elements, including for cross cutting concerns, that would unify the composition ideas behind Model Driven Engineering, Aspect Oriented Modeling, Software Product Lines etc [77]. Several research prototypes [9] have been built to provide new composition operators. In the Mickael Clavreul PhD [72], we define a framework to unify and classify existing model composition operator and ease the definition of new model composition operators. Theoretical basis to such a framework have been recently based on category theory in [48].

6.7. Model Variability

In the context of Aspects Oriented Modeling (AOM), one of the key challenge is the variability management leading to software product lines. Our work in this area has led to the involvement of the Triskell group in the ANR project MOVIDA, as well as in the OMG standardization process of the *Common Variability Language* where we developed a Kermeta-based implementation conforming to this future standard (called *kCVL*).

6.8. Testing software product lines

Nowadays, many applications are expected to run on a tremendous variety of execution environments. For example, network connection software must deliver the same functionalities on distinct physical platforms, which themselves run several distinct operating systems, with various applications and physical devices. Testing those applications is challenging as it is simply impossible to consider every possible environment configuration. We tackle this issue through the systematic selection of a subset of configurations for testing [45] and through model-based verification [37].

6.9. Testing service-oriented applications

The changes resulting from the evolution of Service Based Systems (SBSs) may degrade their design and quality of service (QoS) and may often cause the appearance of common poor solutions, called antipatterns. The automatic detection of antipatterns is thus important to assess the design and QoS of SBSs and ease their maintenance and evolution. Using our approach, we specify 10 well-known and common antipatterns, including Multi Service and Tiny Service, and we automatically generate their detection algorithms [50]. This work has received the best paper award at ICSOC 2012.

6.10. Testing aspect oriented programs

Aspect-oriented programming (AOP) promises better software quality through enhanced modularity.

Crosscutting concerns are encapsulated in separate units called aspects and are introduced at specific points in the base program at compile-time or runtime. However, aspect-oriented mechanisms also introduce new risks for reliability that must be tackled by specific testing techniques in order to fully benefit from the use of AOP. During the evaluation period, we proposed a series of work to analyze these new risks, let designers understand the interactions between the base and the aspects and test aspects. The major achievement is a **novel oracle to test the injection of aspects in a base program**. The oracle allows to capture new classes of errors that occur only in aspect-oriented programs. Its ability to capture these errors in a more efficient way than an object-oriented oracle (shorter test cases and written in less time), has been empirically demonstrated and was published in the Journal for Software Testing, Verification and Reliability [74].

[9] http://www.kermeta.org/kompose/,http://www.kermeta.org/mdk/ModMap/

6.11. Testing peer-to-peer systems

Peer-to-peer (P2P) is one of the major distributed platforms for many applications such as large data sharing and collaboration in social networks. However, building trustworthy P2P applications is difficult because they must be deployed on a large number of autonomous, volatile nodes, which may refuse to answer to some requests and even leave the system unexpectedly. This volatility of nodes is a common behavior in P2P systems and may be interpreted as a fault during tests (*i.e.*, failed node). In this context, we have developed a **novel framework and a methodology for testing P2P applications**. The framework is based on the individual control of nodes, allowing test cases to precisely control the volatility of nodes during their execution. We validated this framework through an experimentation on the FreePastry distributed hashtable. The experimentation tests the behavior of the system in different conditions of volatility and shows how the tests were able to detect complex implementation errors. This work, published in the Empirical Software Engineering journal [73], in collaboration with the ATLAS Inria team, is directly related to Triskell's goal to apply software engineering to distributed systems.

6.12. Testing the boundaries of a specific domain

The increasing use of domain-specific modeling to increase efficiency in modeling multiple concerns, increases the need to correctly formalize domain models. Domains are modeled as metamodels, which capture the domain's modeling spaces, *i.e.* the set of all models which structure conforms to the description specified in the metamodel. However, there is currently no systematic method to test that a metamodel captures all the correct models of the domain and no more. Our most recent contribution to testing focuses on the **automatic selection of models in the modeling space captured by a metamodel**. We adapt metaheuristic search to generate a set that covers as many representative situations as possible, while staying as small as possible. This work was published in the International Conference on Software Testing, verification and validation [27].

6.13. Testing interactive systems

While model-based design of interactive systems is moving from pure event-based models of WIMP interactions to stateful models of post-WIMP interactions, model-based test generation techniques for HCI currently consider only WIMP interaction testing. We proposed an original model-based test generation technique, which aims at providing test cases to test post-WIMP behavior (*e.g.* multi-touch). We leverage the Malai architecture to model the system under test to establish two contributions: the definition of novel adequacy criteria to generate test cases that cover Malai models; an algorithm for the automatic generation of test suites that satisfy the adequacy criteria. We applied the novel approach to two open-source interactive systems to validate the ability of generated test cases to reveal bugs. This early work is part of the project Connexion (*cf.* Section 8.1.3) which notably focuses on testing interactive parts of critical systems.

7. Bilateral Contracts and Grants with Industry

7.1. VaryMDE

Participants: Benoit Combemale, Olivier Barais, Mathieu Acher, Jean-Marc Jézéquel, Joao Ferreira Filho, Suresh Pillay.

MDE, Variability Management, Software Language Engineering.

This bilateral collaboration is between the Triskell team and the MDE lab at Thales Research & Technology. This partnership explores variability management both in modeling and metamodeling (i.e., design and implementation of software languages), and enrolls 4 faculty members and 2 PhD students from the Triskell team. This year, we keep working on the CVL usage in the Thales context.

Project duration: 2011-2014

Triskell budget share: 284 keuros

7.2. EDF

Participants: Nicolas Sannier, Benoit Baudry.

model-driven analysis, requirements modelling, evolution

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

7.3. Kereval

Participants: Aymeric Hervieu, Benoit Baudry.

test generation, software product lines, test reuse

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

7.4. Sodifrance

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

Software Process, Intentional-Driven Development, Process Execution

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

7.5. All4Tec

Participants: Hamza Sahmi, Benoit Baudry.

Model-based testing, Software product lines

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

7.6. Zenexity

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

Web engineering, Domain Specific Languages

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. National Initiatives

8.1.1. ANR GEMOC

Participants: Benoit Combemale, Didier Vojtisek, Olivier Barais, Arnaud Blouin, Benoit Baudry.

Heterogeneous modeling, model driven engineering, executable metamodeling, models of computation, simulation.

The ANR project **GEMOC** (French Agency for Research, Program INS 2012) focuses on a generic framework for heterogeneous software model execution and dynamic analysis. This work has the ambition to propose an innovative environment for the design of complex software-intensive systems by providing:

- a formal framework that integrates state-of-the-art in model-driven engineering (MDE) to build domain-specific modeling languages (DSMLs), and models of computation (MoC) to reason over the composition of heterogeneous concerns;
- an open-source design and modeling environment associated to a well-defined method for the definition of DSMLs, MoCs and rigorous composition of all concerns for execution and analysis purposes.

This requires addressing two major scientific issues: the design and verification of a formal framework to combine several different DSMLs relying on distinct MoCs; the design and validation of a methodology for DSMLs and MoC development. GEMOC aims at participating in the development of next generation MDE environments through a rigorous, tool-supported process for the definition of executable DSMLs and the simulation of heterogeneous models.

Project duration: 2012-2016

Triskell budget share: 253 keuros

Number of person/years: 2.2

Project Coordinator: Inria (Triskell)

Participants: ENSTA Bretagne, Inria, IRIT, I3S, Obeo, Thales

8.1.2. ANR INFRA-JVM

Participants: Johann Bourcier, Olivier Barais, Inti Gonzalez.

JVM, Kevoree, Models@Runtime

INFRA-JVM is an ANR project whose goal is to design and provide a new Java Virtual Machine dedicated to pervasive environments. This project focuses on designing a Java Virtual Machine for embedded computing platform offering dynamic reconfiguration capabilities. The project focuses on the three following parts:

- Defining new mechanisms to provide component-based support for provisionning I/O and memory guarantee
- Defining languages and runtime support for efficient process scheduling on multi-core platform
- Optimizing the memory allocation on multi-core platforms.

Triskell mainly works this year on VMkit (the integration platform of the project) and Kevoree (our Component Based platform) to run Kevoree on top of VMkit.

Project duration: 2012-2015

Triskell budget share: 193 keuros

Number of person/years: 2

Project Coordinator: Université Paris 6

Participants: Université Paris 6, Université Bordeaux 1, Université Rennes 1 (Triskell), Ecole des Mines de Nantes

8.1.3. BGLE2 CONNEXION

Participants: Benoit Baudry, Arnaud Blouin, Valéria Lelli, Nicolas Sannier.

requirement, software testing, critical system, HCI, MDE

The cluster CONNEXION (*digital command CONntrol for Nuclear EXport and renovatION*) aims to propose and validate an innovative architecture platforms suitable control systems for nuclear power plants in France and abroad. In this project the Triskell team investigates methods and tools to (i) automatically analyze and compare regulatory requirements evolutions and geographical differences; (ii) automatically generate test cases for critical interactive systems.

Project duration: 2012-2016

Triskell budget share: 515 keuros

Number of person/years: 3

Project Coordinator: EDF

Participants: Atos WorldGrid, Rolls-Royce Civil Nuclear, Corys TESS, Esterel Technologies, All4Tec, Predict, CEA, Inria, CNRS / CRAN, ENS Cachan, LIG, Telecom ParisTech

8.2. European Initiatives

8.2.1. FP7 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

8.2.2. FP7 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

8.2.3. 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.

8.2.4. Artemis CHESS

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

Real-Time Embedded systems, Component-based Development, Model Driven Engineering

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 2012 Triskell contributed to final phase of development of the model editor specially built for CHESS on top of Papyrus. Using its Kermeta platform, Triskell contributed to the design and implementation 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

8.2.5. ITEA2 OPEES

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.

8.2.6. Marie-Curie Relate

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.

8.2.7. MERGE

Program: ITEA2

Project acronym: Merge

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

Duration: December 2012 - December 2015

Triskell budget share: 250 keuros

Coordinator: Thales Research and Technology

Other partners: Thales Global Services, Thales Communications and Security, OBEO, ALL4TEC, Onera, Inria, Université Paris VI, Codenomicon, STUK - Radiation and Nuclear Safety Authority, POHTO nSense Oy, University of Oulu, University of Jyvaskyla, Space Applications Services NV, Melexis, E2S, Katholieke Universiteit Leuven

Abstract: MERgE stands for "Multi-Concerns Interactions System Engineering". Within the "Engineering support" theme of ITEA2 roadmap, the purpose of this project is to develop and demonstrate innovative concepts and design tools addressing in combination the "Safety" and "Security" concerns, targeting the elaboration of effective architectural solutions. MERgE will provide tools and solutions for combining safety and security concerns in systems development in a holistic way. It will provide academically solid and practice proven solutions and models for system developers and system owners to tackle the challenges of designing seamless optimal cost effective safe and secure solutions conformant to the model driven engineering paradigm. This will be done by tightly integrating the following paradigms: requirement engineering, safety, security and risk management in an over-all design process which is supported by adequate tools and methods. MERgE aims to bring a system engineering solution for Combined Safe & Secure system design. The main technical innovation of the project is the application of state of the art design tools tailorisation capabilities and "multi concern engineering" core technologies to the issue of interactions of "Safety" and "Security" concerns as well as other concerns like "Performance" or "Timing" in the design process.

8.3. International Initiatives

8.3.1. Inria International Partners

Following the Diva STREP project, we keep an active collaboration with the SINTEF institute. Fran√ßois 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 Œº-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 Œº-Kevoree is a viable solution with strong benefits over adaptation through dynamic firmware upgrades.

Following the MoCAA Equipe associée, we keep an active collaboration with Colorado State University. Benoit Baudry and Benoit Combemale visited CSU in April 2012 and Philippa Bennett spent a 4-monhts internship in Triskell. We continue the collaboration with Prof. Sanjay Rajopadhye (from the optimizing compiler domain) to cross-fertilize both HPC and MDE. Results of this collaboration were published in the Journal of Software and Systems in October 2012.

8.3.2. Participation In International Programs

8.3.2.1. TAAS

Program: Foundation Araucaria Inria Brazil

Title: Software testing for cloud computing

Inria principal investigator: Gerson SUNYE

International Partner (Institution - Laboratory - Researcher):

Federal University of Parana (Brazil)

Duration: Jul 2011 - Jun 2013

8.3.2.2. SPLIT

Program: PICS International Project of Scientific Cooperation

Title: Combiner les lignes de produits logicielles et le d√©veloppement logiciel orient√© aspects

Inria principal investigator: Jean-Marc JEZEQUEL

International Partner (Institution - Laboratory - Researcher):

University of Luxembourg (Luxembourg)

Duration: Jan 2009 - Dec 2012

8.4. International Research Visitors

8.4.1. Visits of International Scientists

- Joerg Kienzle (http://www.cs.mcgill.ca/~joerg) - Robert France - Barrett Bryant

8.4.1.1. Internships

Phillipa BENNETT (from Apr 2012 until Sep 2012)

Subject: Model Transformation Testing

Institution: Colorado State University (United States)

Martin FAUNES (from Mar 2012 until May 2012)

Subject: Automated discovery of domain invariants

Institution: Carleton University (Canada)

9. Dissemination

9.1. Scientific Animation

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

9.1.1.2. Benoit Baudry

is an Associate Editor of the following journal:

- Journal on Software and System Modeling: SoSyM

9.1.2. Conferences

9.1.2.1. Benoit Baudry

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

- MODELS 2012 The 15th International Conference on Model Driven Engineering Languages and Systems Innsbruck, Austria, October 2012
- IEEE ICST'12 The 5th International Conference on Software Testing Verification and Validation, Montreal, Canada, April 2012
- GECCO'12, The International Conference on Genetic and Evolutionary Computation Conference, Philadelphia, USA? July 2012
- GPCE'12, The International Conference on Generative Programming and Components Engineering
- NIER'12, The New Ideas and Emerging Results track at ICSE'12, Zurich, Switzerland, May 2011.
- AST workshop on Automated Software Testing, at ICSE'11, Honolulu, USA, May 2011
- Mutation'12 workshop at ICST'12, Montreal, Canada, April 2012

9.1.2.2. Jean-Marc Jézéquel

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

- MODELS 2012 The 15th International Conference on Model Driven Engineering Languages and Systems Innsbruck, Austria, October 2012

9.1.2.3. Jean-Marc Jézéquel

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

- Models@run.time 7th International Workshop on Models at Run-Time, associated to MODELS 2012, Innsbruck, Austria, October 2012

9.1.2.4. Benoit Combemale

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

- APSEC'12, The 19th Asia-Pacific Software Engineering Conference, Hong Kong, 2012.
- SEFM'12, The 9th International Conference on Software Engineering and Formal Methods, 2012.
- AMT workshop on Analysis of Model Transformations (AMT), at MODELS'12, Innsbruck, Austria, 2012.
- ICMT'12, The 5th International Conference on Model Transformation, 2012.
- ECMFA'12, The 8th European Conference on Modelling Foundations and Applications, Lyngby, Denmark, july 2012.
- PMDE workshop at ECMFA'12, Lyngby, Denmark, july 2012.
- CSP track at WETICE'12, Toulouse, France, 2012.

9.1.2.5. Olivier Barais

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

- CIEL'12, The 1st french Conference on Software Engineering, Rennes, 2012.
- SEAA'12, The 38th Euromicro Conference on Software Engineering and Advanced Applications, 2012.
- VARYCOMP'12, The 3rd International Workshop on Variability & Composition @Modularity·AOSD.
- SAC'12 The the 27th Symposium On Applied Computing, Trento, Italy, 2012.

9.1.2.6. Noël Plouzeau

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

- CBSE 2012, Component Based Software Engineering, Bertinoro, Italy, june 2012,.

9.1.3. Workshops, Tutorials and Keynotes

9.1.3.1. Jean-Marc Jézéquel

has co-organized VARY'12 (Variability for You) a workshop of MODELS 2012, Innsbruck, Austria, October 2012.

9.1.3.2. Mathieu Acher

gave a tutorial "Next-Generation Model-based Variability Management: Languages and Tools" at MODELS 2012 (15th International Conference on Model Driven Engineering Languages and Systems Innsbruck).

9.2. Miscellaneous

9.2.1. Benoit Baudry

is on the steering committee of the IEEE International Conference on Software Testing Verification and Validation.

He is program committee chair for ICST'13.

9.2.2. Jean-Marc Jézéquel

is on the steering committee of the IEEE/ACM Conference on Aspect Oriented Software Development

9.3. Teaching - Supervision - Juries

9.3.1. 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, Arnaud Blouin and Mathieu Acher 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 was co-chair of the 8th Educators' Symposium at MODELS 2012.

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

9.3.2. Supervision

PhD : Juan Cadavid, *Analyse formelle pour l'assistance à la méta-modélisation*, thèse 12 décembre 2012, J.-M. Jézéquel and B. Baudry

PhD in progress : Erwan Bousse, *Intégration et combinaison des techniques de V&V dans un contexte d'ingénierie système*, thèse, 2012-2015, B. Baudry and B. Combemale

PhD in progress : Suresh Pillay, *Variability Management in Modeling Languages*, thèse, 2011-2014, B. Baudry and B. Combemale

PhD in progress : Julien Richard-Foy, *A DSL factory for modular Web oriented architecture*, thèse CIFRE avec Zenexity, 2011-2014, J.-M. Jézéquel and O. Barais

PhD in progress : Olivier-Nathanael Ben David, *Secure Service Architectures and Design*, thèse 2010-2013, J.-M. Jézéquel and B. Baudry

PhD in progress : Clément Guy, *Generic Definition of Domain Specific Analysis using MDE*, thèse 2010-2013, J.-M. Jézéquel and B. Combemale

PhD in progress : Tam Le Nhan, *Model-Driven Software Engineering for Cloud Computing*, thèse 2010-2013, J.-M. Jézéquel and G. Sunye

PhD in progress : Emmanuelle Rouillé, *Processus Logiciels dirigés par les intentions*, thèse CIFRE avec SodiFrance, 2010-2013, J.-M. Jézéquel, B. Combemale and O. Barais

PhD in progress : Nicolas Sannier, *IDM pour l'ingénierie des exigences*, thèse CIFRE avec EDF 2010-2013, B. Baudry

PhD in progress : Stephen Creff, *Hierarchical Product Lines*, thèse 2009-2012, J.-M. Jézéquel and J. Champeau

PhD in progress : Francois Fouquet, *Contracts for Enterprise Service Bus*, thèse 2009-2012, J.-M. Jézéquel and N. Plouzeau

PhD in progress : Paul Istoan, *SOA Product Lines*, thèse en co-tutuelle avec l'université du Luxembourg, 2009-2012, J.-M. Jézéquel and N. Guelfi

PhD in progress : Viet-Hoa Nguyen, *Model Driven Design Methodology for Trusted Real-Time Components*, thèse 2009-2012, J.-M. Jézéquel and N. Plouzeau

PhD in progress : Valéria Lelli, *On Testing Interactive Systems*, thèse 2012-2015, B. Baudry and A. Blouin

PhD in progress : Antonio Mattos Junior, *On Validating Models at Runtime adaptation*, thèse 2012-2014, N. Plouzeau and O. Barais

9.3.3. Juries

9.3.3.1. Benoit Baudry

was in the examination committee of the following PhD thesis:

- Ahmed Turki, Mars 2012, Université d'Orléans (Referee)
- María Francisca Rosique Contereras, June 2012, Universidad Politécnica de Cartagena (Referee)
- Rajwinder Kaur Panesar-Walawege, August 2012, University of Oslo (Referee)
- Amine Raji, March 2012, ENSTA-Bretagne (Referee)
- Florian Noyrit, October 2012, Université de Paris XI (Referee)
- Marie Pelleau, November 2012, Université de Nantes (Referee)
- Marc Palyart, December 2012, Université de Toulouse (Member)
- Thang Pham, December 2012, Telecom Bretagne (Member)

9.3.3.2. Jean-Marc Jézéquel

was in the examination committee of the following HDR and PhD thesis:

- Cédric Fleury, June 2012, Université Rennes 1 (President)
- Eduardo Mazza, June 2012, University of Grenoble (Referee)
- Marcos Aurelio Almeida da Silva, June 2012, University of Paris 6 (Referee)
- Ajay Kattepur, November 2012, Université Rennes 1 (President)
- Thierry Duval (HDR), November 2012, Université Rennes 1 (President)
- Bertrand Couasnon (HDR), December 2012, Université Rennes 1 (President)

9.3.3.3. Olivier Barais

was in the examination committee of the Ali Hassan PhD (Member).

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, pp. 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 p. , 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, pp. 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, pp. 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, pp. 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, pp. 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, pp. 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, pp. 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, pp. 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, pp. 557–586

Publications of the year

Articles in International Peer-Reviewed Journals

[12] M. ACHER, P. COLLET, P. LAHIRE, R. FRANCE. *FAMILIAR: A Domain-Specific Language for Large Scale Management of Feature Models*, in "Science of Computer Programming", December 2012 [*DOI :* 10.1016/J.SCICO.2012.12.004], http://hal.inria.fr/hal-00767175

[13] M. BIAZZINI, A. MONTRESOR. *P2POEM: Function optimization in P2P networks*, in "Peer-to-Peer Networking and Applications", July 2012, pp. 1-20 [*DOI :* 10.1007/S12083-012-0152-8], http://hal.inria.fr/hal-00726413

[14] A. BLOUIN, B. COMBEMALE, B. BAUDRY, O. BEAUDOUX. *Kompren: Modeling and Generating Model Slicers*, in "Software and Systems Modeling", October 2012, http://hal.inria.fr/hal-00746566

[15] T. DEMONGEOT, E. TOTEL, V. VIET TRIEM TONG, Y. LE TRAON. *User Data Confidentiality in an Orchestration of Web Services*, in "International Journal of Information Assurance and Security", 2012, vol. 7, nc p. , http://hal.inria.fr/hal-00735996

[16] P. DHAUSSY, F. BONIOL, J.-C. ROGER, A. RAJI, Y. LE TRAON, B. BAUDRY. *Formalisation de contextes et d'exigences pour la validation formelle de logiciels embarqués*, in "Technique et Science Informatiques (TSI)", October 2012, vol. VOL 31, n^o 6, pp. 797-826 [*DOI :* 10.3166/TSI.31.797-826], http://hal.inria.fr/hal-00744434

[17] X. DOLQUES, M. HUCHARD, C. NEBUT, P. REITZ. *Fixing Generalization Defects in UML Use Case Diagrams*, in "Fundamenta Informaticae", January 2012, vol. 115, n^o 4, pp. 327-356 [*DOI :* 10.3233/FI-2012-658], http://hal.inria.fr/lirmm-00726993

[18] J.-M. JÉZÉQUEL, B. COMBEMALE, S. DERRIEN, C. GUY, S. RAJOPADHYE. *Bridging the Chasm Between MDE and the World of Compilation*, in "Journal of Software and Systems Modeling (SoSyM)", October 2012, vol. 11, n^{o} 4, pp. 581-597 [*DOI :* 10.1007/s10270-012-0266-8], http://hal.inria.fr/hal-00717219

[19] P.-A. MULLER, F. FONDEMENT, B. BAUDRY, B. COMBEMALE. *Modeling Modeling Modeling*, in "SOSYM", 2012, vol. 11, n^{o} 3, pp. 347-359 [*DOI :* 10.1007/s10270-010-0172-x], http://hal.inria.fr/inria-00538459

[20] F. MUNOZ, B. BAUDRY, R. DELAMARE, Y. LE TRAON. *Usage and Testability of AOP: an empirical study of AspectJ*, in "Information and Software Technology", August 2012, http://hal.inria.fr/hal-00726119

[21] S. SEN, N. MOHA, V. MAHÉ, O. BARAIS, B. BAUDRY, J.-M. JÉZÉQUEL. *Reusable model transformations*, in "Software and Systems Modeling (SoSyM)", 2012, vol. 11, n^{o} 1, pp. 111-125, 10.1007/s10664-009-9124-x [*DOI :* 10.1007/s10270-010-0181-9], http://hal.inria.fr/inria-00542766

Articles in Non Peer-Reviewed Journals

[22] D. VOJTISEK. *Comment allons-nous développer d'ici 5 à 10 ans ?*, in "Programmez!", March 2012, vol. 150, pp. 44-46, http://hal.inria.fr/hal-00714953

International Conferences with Proceedings

[23] M. AMRANI, L. LUCIO, G. SELIM, B. COMBEMALE, J. DINGEL, H. VANGHELUWE, Y. LE TRAON, J. R. CORDY. *A Tridimensional Approach for Studying the Formal Verification of Model Transformations*, in "Verification and validation Of model Transformations (VOLT)", Montreal, Canada, IEEE, April 2012, http://hal.inria.fr/hal-00684717

[24] Z. AZMEH, M. HUCHARD, F. HAMOUI, N. MOHA. *From Abstract to Executable BPEL Processes with Continuity Support*, in "ICWS 2012: 19th International Conference on Web Services, Application and Experience Track", Honolulu, Hawaii, United States, C. GOBLE, P. CHEN, J. ZHANG (editors), IEEE Computer Society, June 2012, pp. 368-374 [*DOI :* 10.1109/ICWS.2012.124], http://hal.inria.fr/lirmm-00727018

[25] O. BEAUDOUX, M. CLAVREUL, A. BLOUIN, M. YANG, O. BARAIS, J.-M. JÉZÉQUEL. *Specifying and Running Rich Graphical Components with Loa*, in "EICS'12: Proceedings of the 4th ACM SIGCHI symposium on Engineering interactive computing systems", Copenhagen, Denmark, June 2012, pp. 169-178 [*DOI :* 10.1145/2305484.2305513], http://hal.inria.fr/hal-00684881

[26] E. BOUSSE, D. MENTRÉ, B. COMBEMALE, B. BAUDRY, K. TAKAYA. *Aligning SysML with the B Method to Provide V&V for Systems Engineering*, in "Model-Driven Engineering, Verification, and Validation 2012 (MoDeVVa 2012)", Innsbruck, Austria, September 2012, http://hal.inria.fr/hal-00741134

[27] J. CADAVID, B. BAUDRY, H. SAHRAOUI. *Searching the boundaries of a modeling space to test metamodels*, in "Fifth IEEE International Conference on Software Testing, Verification and Validation", Montréal, Canada, April 2012, http://hal.inria.fr/hal-00665866

[28] D. CHIOREAN, B. COMBEMALE. *The 8th Educators' Symposium @ MODELS 2012: Software Modeling in Education*, in "Proceedings of the 8th edition of the Educators' Symposium", Innsbruck, Austria, ACM, 2012 [*DOI :* 10.1145/2425936.2425937], http://hal.inria.fr/hal-00773311

[29] B. COMBEMALE, O. BARAIS, O. ALAM, J. KIENZLE. *Using CVL to Operationalize Product Line Development with Reusable Aspect Models*, in "VARY@MoDELS'12: VARiability for You", Innsbruck, Austria, ACM, 2012, http://hal.inria.fr/hal-00730274

[30] B. COMBEMALE, X. CRÉGUT, M. PANTEL. *A Design Pattern to Build Executable DSMLs and associated V&V tools*, in "The 19th Asia-Pacific Software Engineering Conference (APSEC 2012)", Hong Kong, Hong Kong, IEEE, December 2012, http://hal.inria.fr/hal-00739332

[31] B. COMBEMALE, C. HARDEBOLLE, C. JACQUET, F. BOULANGER, B. BAUDRY. *Bridging the Chasm between Executable Metamodeling and Models of Computation*, in "SLE2012 - 5th International Conference on Software Language Engineering", Dresden, Germany, LNCS, Springer, 2012, http://hal.inria.fr/hal-00725643

[32] B. COMBEMALE, X. THIRIOUX, B. BAUDRY. *Formally Defining and Iterating Infinite Models*, in "ACM/IEEE 15th International Conference on Model Driven Engineering Languages & Systems (MODELS 2012)", Innsbruck, Austria, R. FRANCE, ET AL. (editors), LNCS, Springer-Verlag Berlin Heidelberg, 2012, vol. 7590, pp. 119-133 [*DOI :* 10.1007/978-3-642-33666-9_9], http://hal.inria.fr/hal-00716623

[33] S. CREFF, J. CHAMPEAU, J.-M. JEZEQUEL. *Model-Based Product Line Evolution: an Incremental Growing by Extension*, in "16th International Software Product Line Conference SPLC 2012", El Salvador, September 2012, vol. II, pp. 107-114, http://hal.inria.fr/hal-00738772

[34] S. CREFF, J. CHAMPEAU, A. MONÉGIER, J.-M. JÉZÉQUEL. *Relationships Formalization for Model-Based Product Lines*, in "APSEC 2012", Hong Kong, December 2012, 187 p. , http://hal.inria.fr/hal-00773497

[35] S. CREFF, J. CHAMPEAU, A. MONÉGIER, J.-M. JÉZÉQUEL. *Une organisation des Lignes de Produits Logiciels autour d'un motif architectural*, in "CAL 2012", Montpellier, France, May 2012, xx p. , http://hal.inria.fr/hal-00704312

[36] X. DEVROEY, M. CORDY, G. PERROUIN, E.-Y. KANG, P.-Y. SCHOBBENS, P. HEYMANS, A. LEGAY, B. BAUDRY. *A Vision for Behavioural Model-Driven Validation of Software Product Lines*, in "International Symposium On Leveraging Applications of Formal Methods, Verification and Validation", Amirandes, Heraclion, Greece, October 2012, http://hal.inria.fr/hal-00726121

[37] X. DEVROEY, M. CORDY, G. PERROUIN, E.-Y. KANG, P.-Y. SCHOBBENS, P. HEYMANS, A. LEGAY, B. BAUDRY. *Towards Behavioural Model-Driven Validation of Software Product Lines*, in "5th International Sympoisum on Leveraging Applications of Formal Methods", Héraklion, Crète, Greece, October 2012, http://hal.inria.fr/hal-00718180

[38] J. FILHO, O. BARAIS, B. BAUDRY, J. LE NOIR. *Leveraging variability modeling for multi-dimensional Model-driven Software Product Lines*, in "3rd International Workshop on Product Line Approaches in Software Engineering (PLEASE)", Zurich, Switzerland, June 2012, pp. 5-8 [*DOI :* 10.1109/PLEASE.2012.6229774], http://hal.inria.fr/hal-00726402

[39] J. FILHO, O. BARAIS, B. BAUDRY, W. VIANA, R. ANDRADE. *An Approach for Semantic Enrichment of Software Product Lines*, in "Third International Workshop on Knowledge Oriented Product Line Engineering (KOPLE)", Salvador, Brazil, September 2012, http://hal.inria.fr/hal-00726405

[40] F. FOUQUET, O. BARAIS, N. PLOUZEAU, J.-M. JÉZÉQUEL, B. MORIN, F. FLEUREY. *A Dynamic Component Model for Cyber Physical Systems*, in "15th International ACM SIGSOFT Symposium on Component Based Software Engineering", Bertinoro, Italy, July 2012, http://hal.inria.fr/hal-00713769

[41] F. FOUQUET, E. DAUBERT, N. PLOUZEAU, O. BARAIS, J. BOURCIER, J.-M. JÉZÉQUEL. *Dissemination of reconfiguration policies on mesh networks*, in "DAIS 2012", Stockholm, Sweden, June 2012, http://hal.inria.fr/hal-00688707

[42] F. FOUQUET, G. NAIN, B. MORIN, E. DAUBERT, O. BARAIS, N. PLOUZEAU, J.-M. JÉZÉQUEL. *An Eclipse Modelling Framework Alternative to Meet the Models@Runtime Requirements*, in "Models 2012", Innsbruck, Austria, October 2012, http://hal.inria.fr/hal-00714558

[43] C. GUY, B. COMBEMALE, S. DERRIEN, J. STEEL, J.-M. JÉZÉQUEL. *On Model Subtyping*, in "ECMFA - 8th European Conference on Modelling Foundations and Applications", Kgs. Lyngby, Denmark, July 2012, http://hal.inria.fr/hal-00695034

[44] H. HANEN, J. BOURCIER. *Dependability-Driven Runtime Management of Service Oriented Architectures*, in "PESOS - 4th International Workshop on Principles of Engineering Service-Oriented Systems - 2012", Zurich, Switzerland, June 2012, http://hal.inria.fr/hal-00714357

[45] A. HERVIEU, B. BAUDRY, A. GOTLIEB. *Managing Execution Environment Variability during Software Testing: an industrial experience*, in "International Conference on Testing Software and Systems", Aalborg, Denmark, Springer, November 2012, http://hal.inria.fr/hal-00726137

[46] C. JEANNERET, M. GLINZ, B. BAUDRY, B. COMBEMALE. *Impact of Footprinting on Model Quality: An Experimental Evaluation*, in "Model-Driven Requirements Engineering (MoDRE) Workshop at Re 2012", Chicago, Illinois, United States, IEEE, 2012, pp. 78-87, http://hal.inria.fr/hal-00722036

[47] T. LE NHAN, G. SUNYÉ, J.-M. JÉZÉQUEL. *A Model-Driven Approach for Virtual Machine Image Provisioning in Cloud Computing*, in "European Conference on Service-Oriented and Cloud Computing (ESOCC 2012)", Bertinoro, Italy, Springer-Verlag Berlin Heidelberg, September 2012, pp. 107-121, http://hal.inria.fr/hal-00726086

[48] J. MARCHAND, B. COMBEMALE, B. BAUDRY. *A Categorical Model of Model Merging and Weaving*, in "MiSe 2012 - 4th International Workshop on Modeling in Software Engineering", Zurich, Switzerland, Conference Publishing Solutions, June 2012, http://hal.inria.fr/hal-00714373

[49] J. MEIRA, E. CUNHA DE ALMEIDA, Y. LE TRAON, G. SUNYÉ. *Peer-to-Peer Load Testing*, in "2012 IEEE Fifth International Conference on Software Testing, Verification and Validation", Montreal, Canada, G. ANTONIOL, A. BERTOLINO, Y. LABICHE (editors), 2012, pp. 642-647, http://hal.inria.fr/hal-00726404

[50] N. MOHA, F. PALMA, M. NAYROLLES, B. JOYEN CONSEIL, G. YANN-GAEL, B. BAUDRY, J.-M. JÉZÉQUEL. *Specification and Detection of SOA Antipatterns*, in "International Conference on Service Oriented Computing", Shanghai, China, C. LIU, H. LUDWIG, F. TOUMANI (editors), November 2012, http://hal.inria.fr/hal-00722472

[51] V. H. NGUYEN, F. FOUQUET, N. PLOUZEAU, O. BARAIS. *A Process for Continuous Validation of Self-Adapting Component Based Systems*, in "7th International Workshop on Models@run.time of the MODELS 2012 Conference.", Innsbruck, Austria, December 2012, http://hal.inria.fr/hal-00764706

[52] G. PERROUIN, B. MORIN, F. CHAUVEL, F. FLEUREY, J. KLEIN, Y. LE TRAON, O. BARAIS, J.-M. JÉZÉQUEL. *Towards flexible evolution of Dynamically Adaptive Systems*, in "34th International Conference on Software Engineering (ICSE), NIER Track", Zurich, Switzerland, June 2012, pp. 1353-1356 [*DOI :* 10.1109/ICSE.2012.6227081], http://hal.inria.fr/hal-00726415

[53] S. ROUIBIA, M. GHAREED, B. PARREIN, M. BIAZZINI, R. CARVAJAL-GOMEZ, A. PEREZ-ESPINOSA, P. SERRANO-ALVARADO. *Towards a Hybrid Client/Server and P2P Architecture for Content Delivery over the Internet*, in "CFIP/NOTERE", Bayonne, France, October 2012, 1 p. , http://hal.inria.fr/hal-00761483

[54] E. ROUILLÉ, B. COMBEMALE, O. BARAIS, D. TOUZET, J.-M. JÉZÉQUEL. *Leveraging CVL to Manage Variability in Software Process Lines*, in "Asia-Pacific Software Engineering Conference", Hong Kong, China, December 2012, http://hal.inria.fr/hal-00735881

[55] N. SANNIER, B. BAUDRY. *Defining and Retrieving Themes in Nuclear Regulations*, in "Fifth International Workshop on Requirements Engineering and Law (RELAW 2012)", Chicago, United States, September 2012, http://hal.inria.fr/hal-00721826

[56] N. SANNIER, B. BAUDRY. *Toward Multilevel Textual Requirements Traceability Using Model-Driven Engineering and Information Retrieval*, in "Second International Model-Driven Requirements Engineering (MoDRE) Workshop", Chicago, United States, September 2012, http://hal.inria.fr/hal-00721806

National Conferences with Proceedings

[57] O. FINOT, J.-M. MOTTU, G. SUNYÉ, C. ATTIOGBÉ. *Comparaison de Modèles Filtrée pour le Test de Transformations de Modèles*, in "CIEL 2012", Rennes, France, 2012, http://hal.inria.fr/hal-00726538

[58] F. FOUQUET, E. DAUBERT, N. PLOUZEAU, O. BARAIS, J. BOURCIER, A. BLOUIN. *Kevoree : une approche model@runtime pour les systèmes ubiquitaires*, in "UbiMob2012", Anglet, France, June 2012, http://hal.inria.fr/hal-00714557

Conferences without Proceedings

[59] M. ACHER, P. HEYMANS, P. COLLET, P. LAHIRE. *Next-Generation Model-based Variability Management: Languages and Tools*, in "MODELS 2012 - 15th International Conference on Model Driven Engineering Languages & Systems", Innsbruck, Austria, September 2012, http://hal.inria.fr/hal-00767170

Scientific Books (or Scientific Book chapters)

[60] L. D. OLIVIER BARAIS (editor). , *Actes des 4èmes journées du GDR Génie de la Programmation et du Logiciel, Rennes, 2012*, Université de Rennes, June 2012, 256 p. , http://hal.inria.fr/hal-00713033

[61] A. HUBAUX, M. ACHER, P. HEYMANS, P. COLLET, P. LAHIRE. *Separating Concerns in Feature Models: Retrospective and Support for Multi-Views*, in "Domain Engineering: Product Lines, Conceptual Models, and Languages", I. REINHARTZ-BERGER, A. STURM, T. CLARK, J. BETTIN, S. COHEN (editors), Springer, December 2012, http://hal.inria.fr/hal-00767213

[62] J.-M. JÉZÉQUEL, B. COMBEMALE, D. VOJTISEK. , ELLIPSES (editor), *Ingénierie Dirigée par les Modèles : des concepts à la pratique...*, Références sciences, Ellipses, February 2012, 144 p. , http://hal.inria.fr/hal-00648489

Research Reports

[63] B. BAUDRY, M. MONPERRUS. , *Towards Ecology Inspired Software Engineering*, Inria, May 2012, n^o RR-7952, http://hal.inria.fr/hal-00694276

[64] J. CADAVID, B. COMBEMALE, B. BAUDRY. , *Ten years of Meta-Object Facility: an Analysis of Metamodeling Practices*, Inria, February 2012, n^o RR-7882, http://hal.inria.fr/hal-00670652

[65] B. COMBEMALE, X. CRÉGUT, M. PANTEL. , *A Design Pattern for Executable DSML*, Inria, September 2012, n^o RR-8063, 19 p. , http://hal.inria.fr/inria-00540648

Other Publications

[66] A. BLOUIN, B. COMBEMALE, B. BAUDRY. , *Découpez vos Modèles avec Kompren : une Démonstration*, June 2012, Démonstration, 4emes journées nationales du GDR-GPL 2012, p. 201–202, http://hal.inria.fr/hal-00696149

[67] C. GUY. , *On Model Subtyping*, June 2012, http://hal.inria.fr/hal-00726399

[68] N. SANNIER. , *Ingénierie dirigée par les modèles pour structurer et partager un référentiel d'exigences de sûreté dans la durée*, June 2012, 203 p. , http://hal.inria.fr/hal-00718895

References in notes

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

[70] 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, pp. 62–76 [*DOI :* 10.1007/978-3-642-24485-8_6], http://hal.inria.fr/inria-00609072

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

[72] 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

[73] E. DE ALMEIDA, G. SUNYÉ, Y. LE TRAON, P. VALDURIEZ. *Testing peer-to-peer systems*, in "Empirical Software Engineering", 2010, vol. 15, pp. 346-379, 10.1007/s10664-009-9124-x, http://www.irisa.fr/triskell/publis/2010/Almeida10.pdf

[74] 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

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

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

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

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

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

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

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

[82] B. MORIN, O. BARAIS, G. NAIN, J.-M. JÉZÉQUEL. *Taming Dynamically Adaptive Systems with Models and Aspects*, in "31st International Conference on Software Engineering (ICSE'09)", Vancouver, Canada, Canada, 2009, http://hal.inria.fr/inria-00468516

[83] 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

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

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

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

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