IN PARTNERSHIP WITH:
Institut national des sciences appliquées de Rennes
Université Rennes 1

Activity Report 2013

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, end of the Project-Team: 2013 December 31.

1. Members

Research Scientists

Benoit Baudry [Team leader, Inria, Researcher, HdR]
Benoit Combemale [Inria, Researcher, from Sep 2013]
Axel Legay [Inria, Researcher]

Faculty Members

Mathieu Acher [Univ. Rennes I, Associate Professor]
Olivier Barais [Univ. Rennes I, Associate Professor]
Arnaud Blouin [INSA Rennes, Associate Professor]
Johann Bourcier [Univ. Rennes I, Associate Professor]
Jean-Marc Jézéquel [Univ. Rennes I, Professor, HdR]
Naouel Moha [Univ. Rennes I, Associate Professor]
Noël Plouzeau [Univ. Rennes I, Associate Professor]

External Collaborators

Olivier Beaudoux [Associate Professor, ESEO, Angers, France, from October 2013]
Robert France [Full Professor, Colorado State University, USA, from October 2013]

Engineers

Didier Vojtisek [Inria, Senior Engineer]
Benoit Boyer [Inria, FP7 DANSE project, from Jan 2013]
Kevin Corre [Inria, from Jan 2013]
Fabien Coulon [Inria, Caisse des Dépôts et Consignations, from Feb 2013]
Jean-Émile Dartois [Inria]
Erwan Daubert [Univ. Rennes I]
Benoit Delahaye [Inria, FP7 DALI project, from Jan 2013 until Aug 2013]
André Elie [Inria, Caisse des Dépôts et Consignations, from Jun 2013]
Rudolf Fahrenberg [Inria, FP7 DALI project, from Jan 2013]
Vincent Fontanella [Inria, ANR GeMoC project, from Feb 2013 until Oct 2013]
Sean Sedwards [Inria, Conseil Régional de Bretagne, from Jan 2013]
Maxime Tricoire [Inria, FP7 HEADS project]

PhD Students

Francisco-Javier Acosta Padilla [Univ. Rennes I]
Olivier-Nathanael Ben David [Inria, FP7 ALES FP 7 NESSos project, until Sep 2013]
Sana Ben Nasr [Inria, Caisse des Dépôts et Consignations, from Feb 2013]
Jacky Bourgeois [Co-tutelle Univ. Rennes 1 with Open University]
Mohamed Boussaa [Inria, from Nov 2013]
Erwan Bousse [Univ. Rennes I]
Guillaume Bécan [Univ. Rennes I (MESR)]
Antonio de Mattos [Univ. Rennes I, until Feb 2013]
Thomas Degueule [Inria, Min. du Redressement Productif, from Oct 2013]
João Bosco Ferreira Filho [Inria, Conseil Régional de Bretagne]
Inti Gonzalez Herrera [Univ. Rennes I]

Clément Guy [Univ. Rennes I, until Sep 2013]
Paul Istoan [Univ. Luxembourg, until Feb 2013]
Cyrille Jégourel [Inria, FP7 DANSE project, from Jan 2013]
Tam Le Nhan [Inria, until Sep 2013]
Valéria Lelli Leitão Dantas [Inria, Caisse des Dépôts et Consignations]
David Mendez Acuna [Univ. Rennes I, from Dec 2013]
Ivan Paez Anaya [Univ. Rennes I]
Suresh Pillay [Univ. Rennes I, until Oct 2013]
Julien Richard-Foy [Zenexity, CIFRE]
Emmanuelle Rouillé [Sodifrance, CIFRE, until Oct 2013]
Hamza Samih [All4Tec, CIFRE]
Nicolas Sannier [EDF, CIFRE, until Oct 2013]
Aymeric Hervieu [Kereval, CIFRE, until Oct 2013]
Kwaku Yeboah-Antwi [Inria, FP7 DIVERSIFY project, from Sep 2013]

Post-Doctoral Fellows

Simon Allier [Inria, FP7 DIVERSIFY project, from Apr 2013]
Walter Rudametkin Ivey [Inria, Min. du Redressement Productif, from Jun 2013]
Edward Mauricio Alferez Salinas [Inria, Min. du Redressement Productif, from Jun 2013]
Marco Biazzini [Inria, Thales]
Fabio Mesquita Buiati [Univ. Rennes I, until Jun 2013]
Louis-Marie Traonouez [Inria, from Jan 2013]

Visiting Scientists

Rima Al Ali [RELATE fellow, Charles University, Czech Republic from Sep 2013 until Nov 2013]
Saurabh Deepak Bansod [Inria, from Jun 2013 until Jul 2013]
Marianela Ciolfi Felice [Inria, from Mar 2013 until Aug 2013]
Jose Angel Galindo Duarte [Univ. Sevilla, Spain, from Jul 2013 until Dec 2013]
Ilias Gerostathopoulos [RELATE fellow, Charles University, Czech Republic, from Apr 2013 until May 2013]
Ioannis Kavvouras [Inria, from Mar 2013 until Jul 2013]
Aleksandar Milenkosky [RELATE fellow, KIT, Germany, from Mar 2013 until May 2013]
Francisco Moyano [PhD student, Univ. of Malaga, Spain, from Jan 2013 until Oct 2013]
Bernhard Rumpe [Professor, Univ. of Aachen, from May 2013 until Jun 2013]
Houari Sahraoui [Professor, Univ. de Montréal, Canada, from Mar 2013 until Apr 2013]
Wuliang Sun [Inria, until Jul 2013]
Edoardo Vacchi [PhD student, Univ. of Milano, Italy, from Sep 2013 until Oct 2013]

Administrative Assistant

Loic Lesage [Inria]

Others

Clement Briand [Univ. Rennes I, Master Student, from Jun 2013 until Sep 2013]
Alexandre Carteron [Univ. Rennes I, Master Student, from Jul 2013 until Sep 2013]
Yorick Perret [Inria, Master Student, from Mar 2013 until Jul 2013]
Rudi Tuvache [Univ. Rennes I, Master Student, from May 2013 until Jul 2013]
Woody Robert [Univ. Rennes I, from May 2013 until Aug 2013]

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

- Prof. Robert France (Colorado State University, USA) has been awarded an Inria International Chair. He will join the team in the next five years to develop a research project on multi-concern software modeling and variability.

- The paper Generating Counterexamples of Model-based Software Product Lines: An Exploratory (J. B. Ferreira Filho, O. Barais, M. Acher, J. Le Noir, B. Baudry.) received a best student paper award at the International Software Product Line Conference
- 9 PhD students defended their thesis in 2013

BEST PAPER AWARD :

[49] **Generating Counterexamples of Model-based Software Product Lines: An Exploratory Study in SPLC - 17th International Software Product Line Conference**. J. B. FERREIRA FILHO, O. BARAIS, M. ACHER, J. LE NOIR, B. BAUDRY.

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 [80], [78]: 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 [82].

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 [79]. 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 [88] 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 [81]).

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 [83] 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) [89]. 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) [77].

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 slightly 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 login) 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 [87], 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 detailed 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 detailed 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, Fabien Coulon, Thomas Degueule, François Fouquet, David Mendez Acuna, Clément Guy, Jean-Marc Jézéquel.

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 highlighting, 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,...

After a first refactoring of Kermeta in 2011 to ease the integration of EMF and to focus on a fully compiled mode, we did a new refactoring of Kermeta in 2013 to leverage on xTend. The Kermeta action language is now defined as an extension of xTend proposing model-specific features (e.g., model type, containment, opposite) and an open class mechanism for aspect weaving. The main objective of this new refactoring was to benefit from the model-non-specific features of xTend (including the basics of the action language and its respective tooling such as editor, type checker and compiler), and to focus in our development on the innovative solutions for MDE.

Especially, in addition to an xTend extension dedicated to model manipulation, we started to integrate in Kermeta various facilities to support a software language engineering (slicing, pruning, reuse, variability management...).

Moreover, while this version of Kermeta is a DSML development workbench that provide good support for developing independent DSMLs, little or no support is provided for integrated use of multiple DSMLs. The lack of support for explicitly relating concepts expressed in different DSMLs makes it very difficult for developers to reason about information spread across models describing different system aspects.

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 model polymorphism (e.g., reuse of algorithms and transformations accross diffrent metamodels), as well as language inheritance, evolution and interoperability.

Impact:

Kermeta is already quite well used by the community as a research platform for trying MDE ideas both in the academic community and in corporate R&D. Many softwares tools are built on top of Kermeta either within the Triskell team, within other Inria teams or in other companies and research institutes:

- The following tools have been built within the Triskell team : K-CVL (implementation of the OMG CVL standard), Kompren (model slicing tool), Malai, Pramana. Kermeta is also used in all the collaborative projects Triskell is involved with, and is the catalyst of many collaborations in industrial contracts.
- The following tools have been built using Kermeta (or use some transformations written in Kermeta) in other Inria teams:
  - Gecos (CAIRN): C compiler infrastructure following the Model Driven Engineering. It leverages the Eclipse Modeling Framework and uses Eclipse as an underlying infrastructure. Consequently, the grammar of the source languages and the intermediate representations become metamodels, and the compilation passes become model transformations.
  - Timesquare (AOSTE) is a language based on the formal Clock Constraint Specification Language (CCSL), which allows the manipulation of logical time.
  - Polychrony (ESPRESSO) is a toolset for a polychronous data-flow language (Signal)
- The following tools have been built using Kermeta outside of Inria:
  - Modhel'x (Supelec) is a framework for simulating multi-formalism models.
  - RAM (Mc Gill University) Reusable Aspect Models is an aspect-oriented multi-view modeling approach that integrates class diagram, sequence diagram and state diagram AOM techniques.

Since 2008, we invested a large effort to transfer these concepts in industry and the standardization bodies. Especially, we have initiated some collaborations with the Eclipse Foundation and OMG to include some Kermeta concepts (model typing, static introduction, ECORE/OCL/Kermeta composition, etc.) in the MXF project proposal [1] of the Eclipse Modeling Project.

According to google scholar [2], the Kermeta platform was used or cited in more than 800 papers. It has been downloaded about 1000 times per year since 2006[3].

5.2. Kevoree

Participants: Olivier Barais [correspondant], François Fouquet, Erwan Daubert, Jean-Émile Dartois, Johann Bourcier, Noël Plouzeau, Maxime Tricoire, Francisco-Javier Acosta Padilla, Jacky Bourgeois, Mohamed Boussaa, Antonio de Mattos, Thomas Degueule, Inti Gonzalez Herrera, Tam Le Nhan, Ivan Paez Anaya.

Kevoree is an open-source models@runtime platform [4] to properly support the dynamic adaptation of distributed systems. Models@runtime basically pushes the idea of reflection [85] 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 [85] and the Entimid project [86]. With Kevoree we push our vision of models@runtime [84] 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 [5].
- SpringSource Dynamic Module [6]
- GCM-Proactive [7]
- OSGi [8]
- Chef [9]
- Vagran [10]

[1] cf. http://www.eclipse.org/proposals/mxf
[2] http://scholar.google.fr/scholar?q=kermeta+model
[3] according to the unique visitors count on the Kermeta update site. Cf. http://kermeta.org/awstats.pl?month=all&year=2010&output=main&config=kermeta.org
[4] http://www.kevoree.org
[5] http://frascati.ow2.org
[6] http://docs.spring.io/osgi/docs/1.2.1/reference/html/
[7] http://proactive.inria.fr/
[8] http://www.osgi.org
[9] http://wiki.opscode.com/display/chef/Deploy+Resource
[10] http://vagrantup.com/

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:

A tutorial have been performed at the Middleware conference in december 2013.

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, Kotlin, Javascript

5.3. FAMILIAR

Participants: Mathieu Acher [correspondant], Olivier Barais, Guillaume Bécan, Aymeric Hervieu, Julien Richard-Foy, Sana Ben Nasr, Edward Mauricio Alferez Salinas, João Ferreira Filho, Didier Vojtisek, Benoit Baudry.

Modeling and reasoning about configuration options is crucial for the effective management of configurable software systems and product lines. The FAMILIAR project provides dedicated languages, APIs, and comprehensive environments for that purpose. Specifically, FAMILIAR provides support for feature models (by far the most popular notation). There are more than 20 years of research [75] and the formalism of feature models is widely used in the industry [76]. FAMILIAR (for FeAture Model scrIpt Language for manIpulation and Automatic Reasoning) provides a scripting language for importing, exporting, composing, decomposing, editing, configuring, computing "diffs", refactoring, reverse engineering, testing, and reasoning about (multiple) feature models. For interoperability, many bridges with existing feature modeling languages are implemented. All these operations can be combined to realize complex variability management tasks: extraction of feature models from software artifacts [23], product line evolution [35], management of multiple models [34], model-based validation of SPLs [49], large scale configuration of feature models [68], etc. The level of maturity of the Familiar platform is TRL 3 (New technology tested Prototype built and functionality demonstrated through testing over a limited range of operating conditions. These tests can be done on a scaled version if scalable)

Main competitors:

- FAMA
- TVL
- Clafer
- pure::variants

Main innovative features:

- reverse engineering of variability models from multiple kinds of artefacts
- composition of multiple variability models (e.g., for combining different sources of variability)
- slicing of variability model (e.g., for scheduling a configuration process in different steps)
- connection with the Common Variability Language (CVL)

Impact:

The results are connected to the CVL standardization initiative. From a research perspective, FAMILIAR helps to support all the research activity on variability modeling (e.g., design of new operators, benchmarking). Several tutorials have been performed at SPLC (the major conference in software product lines), at ECOOP, at CIEL and MODELS in 2012 and 2013. FAMILIAR is also used in the context of teaching activities. From an industrial perspective, the languages and tools have already been applied in practical contexts in different application domains (medical imaging, video surveillance, system engineering, web configurators, etc.) and for various purposes. This platform is also used for supporting the transfer activity with company such as Thales or Kereval. FAMILIAR is currently involved in different research projects (in the Merge Itea project, in the MOTIV project, in the VaryMDE project).

FAMILIAR is distributed under the terms of the LGPL and EPL open source license.

See also the web page http://familiar-project.github.com.

- Version: 1.2
- Programming language: Java, Scala

6. New Results

6.1. Support for Reverse Engineering and Maintaining Feature Models

Feature Models (FMs) are a popular formalism for modelling and reasoning about commonality and variability of a system. In essence, FMs aim to define a set of valid combinations of features, also called configurations. In [35], we tackle the problem of synthesising an FM from a set of configurations. The main challenge is that numerous candidate FMs can be extracted from the same input configurations, yet only a few of them are meaningful and maintainable. We first characterise the different meanings of FMs and identify the key properties allowing to discriminate between them. We then develop a generic synthesis procedure capable of restituting the intended meanings of FMs based on inferred [72] or user-specified knowledge. Using tool support, we show how the integration of knowledge into FM synthesis can be realized in different practical application scenarios that involve reverse engineering and maintaining FMs.

6.2. Feature Model Extraction from Large Collections of Informal Product Descriptions

Feature Models (FMs) are used extensively in software product line engineering to help generate and validate individual product configurations and to provide support for domain analysis. As FM construction can be tedious and time-consuming, researchers have previously developed techniques for extracting FMs from sets of formally specified individual configurations, or from software requirements specifications for families of existing products. However, such artifacts are often not available. In [44] we present a novel, automated approach for constructing FMs from publicly available product descriptions found in online product repositories and marketing websites such as SoftPedia and CNET. While each individual product description provides only a partial view of features in the domain, a large set of descriptions can provide fairly comprehensive coverage. Our approach utilizes hundreds of partial product descriptions to construct an FM and is described and evaluated against antivirus product descriptions mined from SoftPedia.

6.3. On Product Comparison Matrices and Variability Models

Product comparison matrices (PCMs) provide a convenient way to document the discriminant features of a family of related products and now abound on the internet. Despite their apparent simplicity, the information present in existing PCMs can be very heterogeneous, partial, ambiguous, hard to exploit by users who desire to choose an appropriate product. Variability Models (VMs) can be employed to formulate in a more precise way the semantics of PCMs and enable automated reasoning such as assisted configuration. Yet, the gap between PCMs and VMs should be precisely understood and automated techniques should support the transition between the two. We propose variability patterns that describe PCMs content and conduct an empirical analysis of 300+ PCMs mined from Wikipedia [62], we also identify the limits of existing comparators, configurators and PCMs [67], [62]. Our findings are a first step toward better engineering techniques for maintaining and configuring PCMs.

6.4. Generating Counterexamples of Model-based Software Product Lines: An Exploratory Study

Model-based Software Product Line (MSPL) engineering aims at deriving customized models corresponding to individual products of a family. The design space of an MSPL is extremely complex to manage for the engineer, since the number of variants may be exponential and the derived product models have to conform to numerous well-formedness and business rules. We provide a way to generate MSPLs, called counterexamples, that can produce invalid product models despite a valid configuration in the variability model [49]. We provide a systematic and automated process, based on the Common Variability Language (CVL), to randomly search the space of MSPLs for a specific formalism. We validate the effectiveness of this process for three formalisms at different scales (up to 247 metaclasses and 684 rules).

6.5. Composing your Compositions of Variability Models

Modeling and managing variability is a key activity in a growing number of software engineering contexts. Support for composing variability models is arising in many engineering scenarios, for instance, when several subsystems or modeling artifacts, each coming with their own variability and possibly developed by different stakeholders, should be combined together. We consider in [34] the problem of composing feature models (FMs), a widely used formalism for representing and reasoning about a set of variability choices. We show that several composition operators can actually be defined, depending on both matching/merging strategies and semantic properties expected in the composed FM. We present four alternative forms and their implementations. We discuss their relative trade-offs w.r.t. reasoning, customizability, traceability, composability and quality of the resulting feature diagram. We summarize these findings in a reading grid which is validated by revisiting some relevant existing works. Our contribution should assist developers in choosing and implementing the right composition operators.

6.6. Extraction and Evolution of Architectural Variability Models in Plugin-based Systems

Variability management is a key issue when building and evolving software-intensive systems, making it possible to extend, configure, customize and adapt such systems to customers' needs and specific deployment contexts. A wide form of variability can be found in extensible software systems, typically built on top of plugin-based architectures that offer a (large) number of configuration options through plugins. In an ideal world, a software architect should be able to generate a system variant on-demand, corresponding to a particular assembly of plugins. To this end, the variation points and constraints between architectural elements should be properly modeled and maintained over time (i.e., for each version of an architecture). A crucial, yet error-prone and time-consuming, task for a software architect is to build an accurate representation of the variability of an architecture, in order to prevent unsafe architectural variants and reach the highest possible level of flexibility. In [23], we propose a reverse engineering process for producing a variability model (i.e., a feature model) of a plugin-based architecture. We develop automated techniques to extract and combine different variability descriptions, including a hierarchical software architecture model, a plugin dependency model and the software architect knowledge. By computing and reasoning about differences between versions of architectural feature models, software architect can control both the variability extraction and evolution processes. The proposed approach has been applied to a representative, large-scale plugin-based system (FraSCAti), considering different versions of its architecture. We report on our experience in this context.

6.7. FAMILIAR: A Domain-Specific Language for Large Scale Management of Feature Models

The feature model formalism has become the de facto standard for managing variability in software product lines (SPLs). In practice, developing an SPL can involve modeling a large number of features representing di erent viewpoints, sub-systems or concerns of the software system. This activity is generally tedious and error-prone. In [24], we present FAMILIAR a Domain-Specific Language (DSL) that is dedicated to the large scale management of feature models and that complements existing tool support. The language provides a powerful support for separating concerns in feature modeling, through the provision of composition and decomposition operators, reasoning facilities and scripting capabilities with modularization mechanisms. We illustrate how an SPL consisting of medical imaging services can be practically managed using reusable FAMILIAR scripts that implement reasoning mechanisms. We also report on various usages and applications of FAMILIAR and its operators, to demonstrate their applicability to di erent domains and use for di erent purposes.

6.8. Web Configurators

Nowadays, mass customization has been embraced by a large portion of the industry. As a result, the web abounds with sales configurators that help customers tailor all kinds of goods and services to their specific needs. In many cases, configurators have become the single entry point for placing customer orders. As such, they are strategic components of companies' information systems and must meet stringent reliability, usability and evolvability requirements. However, the state of the art lacks guidelines and tools for efficiently engineering web sales configurators. To tackle this problem, empirical data on current practice is required. The paper [51] reports on a systematic study of 111 web sales configurators along three essential dimensions: rendering of configuration options, constraint handling, and configuration process support. Based on this, we highlight good and bad practices in engineering web sales configurator. The reported quantitative and qualitative results open avenues for the elaboration of methodologies to (re-)engineer web sales configurators. In [48] we focus on how to associate product configurations to visual representations in a Web configurator. We present a formal statement of the problem and a model-driven perspective.

6.9. Separating Concerns in Feature Models

Feature models (FMs) are a popular formalism to describe the commonality and variability of a set of assets in a software product line (SPL). SPLs usually involve large and complex FMs that describe thousands of features whose legal combinations are governed by many and often complex rules. The size and complexity of these models is partly explained by the large number of concerns considered by SPL practitioners when managing and configuring FMs. In the chapter [68], we first survey concerns and their separation in FMs, highlighting the need for more modular and scalable techniques. We then revisit the concept of view as a simplified representation of an FM. We finally describe a set of techniques to specify, visualize and verify the coverage of a set of views. These techniques are implemented in complementary tools providing practical support for feature-based configuration and large scale management of FMs.

6.10. Bridging the Chasm between Executable Metamodeling and Models of Computation

The complete and executable definition of a Domain Specific Language (DSL) relies on the specification of two essential facets: a model of the domain-specific concepts with actions and their semantics; a scheduling model that orchestrates the actions of a domain-specific model. Metamodels can capture the former facet, while Models of Computation (MoCs) capture the latter facet. Unfortunately, theories and tools for metamodeling and MoCs have evolved independently, creating a cultural and technical chasm between the two communities. We introduce a new framework to bridge a metamodel and a MoC in a modular fashion [43]. This bridge allows (i) the complete and executable definition of a DSL, (ii) the reuse of MoCs for different domain-specific metamodels, and (iii) the use of different MoCs for a given metamodel, to cope with variation points of a DSL.

6.11. Reifying Concurrency for Executable Metamodeling

Current metamodeling techniques can be used to specify the syntax and semantics of domain specific modeling languages (DSMLs). However, there is currently very little support for explicitly specifying concurrency semantics using metamodels. We reify concurrency as a metamodeling facility, leveraging formalization work from the concurrency theory and models of computation (MoC) community [42]. The essential contribution of this paper is a proposed language workbench for binding domain-specific concepts and models of computation through an explicit event structure at the metamodel level. We illustrate these novel metamodeling facilities for designing two variants of a concurrent and timed final state machine, and provide other experiments to validate the scope of our approach.

6.12. Using Model Types to Support Contract-Aware Model Substitutability

Model typing brings the benefit associated with well-defined type systems to model-driven development (MDD) through the assignment of specific types to models. In particular, model type systems enable reuse of model manipulation operations (e.g., model transformations), where manipulations defined for models of a supertype can be used to manipulate models of subtypes. Existing model typing approaches are limited to structural typing defined in terms of object-oriented metamodels (e.g., MOF) in which the only structural (well-formedness) constraints are those that can be expressed directly in metamodeling notations (e.g., multiplicity and element containment constraints). We propose an extension to model typing that takes into consideration structural invariants, other than those that can be expressed directly in metamodeling notation, and specifications of behaviors associated with model types [64]. The approach supports contract-aware substitutability, where contracts are defined in terms of invariants and pre-/postconditions expressed using OCL. Support for behavioral typing paves the way for behavioral substitutability. We also describe a technique to rigorously reason about model type substitutability as supported by contracts and apply the technique in use cases from the optimizing compiler community.

6.13. Variability Support in Domain-Specific Language Development

Domain Specific Languages (DSLs) are widely adopted to capitalize on business domain experiences. Consequently, DSL development is becoming a recurring activity. Unfortunately, even though it has its benefits, language development is a complex and time-consuming task. Languages are commonly realized from scratch, even when they share some concepts and even though they could share bits of tool support. This cost can be reduced by employing modern modular programming techniques that foster code reuse. However, selecting and composing these modules is often only within the reach of a skilled DSL developer. We propose to combine modular language development and variability management, with the objective of capitalizing on existing assets [63]. This approach explicitly models the dependencies between language components, thereby allowing a domain expert to configure a desired DSL, and automatically derive its implementation. The approach is tool supported, using Neverlang to implement language components, and the Common Variability Language (CVL) for managing the variability and automating the configuration. We illustrate our approach with the help of different case studies, including the implementation of a family of DSLs to describe variants of state machines.

6.14. Automatically Searching for Metamodel Well-Formedness Rules in Examples and Counter-Examples

Current metamodeling formalisms support the definition of a metamodel with two views: classes and relations, that form the core of the metamodel, and well-formedness rules, that constraints the set of valid models. While a safe application of automatic operations on models requires a precise definition of the domain using the two views, most metamodels currently present in repositories have only the first one part. We propose in [47] to start from valid and invalid model examples in order to automatically retrieve well-formedness rules in OCL using Genetic Programming. The approach is evaluated on metamodels for state machines and features diagrams. The experiments aim at demonstrating the feasibility of the approach and at illustrating some important design decisions that must be considered when using this technique.

6.15. Building Modular and Efficient DSLs: Mashup of Meta-Languages and its Implementation in the Kermeta Language Workbench

With the growing use of domain-specific languages (DSL) in industry, DSL design and implementation goes far beyond an activity for a few experts only and becomes a challenging task for thousands of software engineers. DSL implementation indeed requires engineers to care for various concerns, from abstract syntax, static semantics, behavioral semantics, to extra-functional issues such as run-time performance. We propose an approach that uses one meta-language per language implementation concern [27] in the new version (v2) of the Kermeta workbench. We show that the usage and combination of those meta-languages is simple and intuitive enough to deserve the term "mashup". We evaluate the approach by completely implementing the non trivial fUML modeling language, a semantically sound and executable subset of the Unified Modeling Language (UML) ; Kompren, a DSL for designing and implementing model slicers ; and KCVL, the Commun Variability Language dedicated to variability management in software design models [65].

6.16. On the Globalization of Modeling Languages

In the software and systems modeling community, research on domain-specific modeling languages (DSMLs) is focused on providing technologies for developing languages and tools that allow domain experts to develop system solutions efficiently in a particular domain. Unfortunately, the current lack of support for explicitly relating concepts expressed in different DSMLs makes it very difficult for software and system engineers to reason about information spread across models describing different system aspects. Supporting coordinated use of DSMLs leads to what we call the globalization of modeling languages. We present a research initiative that broadens the current DSML research focus beyond the development of independent DSMLs to one that provides support for globalized DSMLs, that is, DSMLs that facilitate coordination of work across different domains of expertise [31]. We explore this new grand challenge in recent workshops, *e.g.,* GlobalDSL'13 at ECSA, ECMFA and ECOOP 2013 [69], and GEMOC'13 at MODELS 2013 [70].

6.17. Automating the Maintenance of Non-functional System Properties using Demonstration-based Model Transformation

Given a base model with functional components, maintaining the non-functional properties that crosscut the base model has become an essential modeling task when using DSMLs. We present a demonstration-based approach to automate the maintenance of non-functional properties in DSMLs [29]. Instead of writing model transformation rules explicitly, users demonstrate how to apply the non-functional properties by directly editing the concrete model instances and simulating a single case of the maintenance process. An inference engine generates generic model transformation patterns, which can be refined by users and then reused to automate the same evolution and maintenance task in other models. Our demonstration-based approach has been applied to several scenarios, such as auto-scaling and model layout.

6.18. Improving Reusability and Automation in Software Process Lines

Software processes orchestrate manual or automatic tasks to create new software products that meet the requirements of specific projects. While most of the tasks are about inventiveness, modern developments also require recurrent, boring and time-consuming tasks (e.g., the IDE configuration, or the continuous integration setup). Such tasks struggle to be automated due to their various execution contexts according to the requirements of specific projects. We propose a methodology that benefits from an explicit modeling of a family of processes to identify the possible reuse of automated tasks in software processes [60]. Then, we propose a tool-supported approach that integrates both reuse and automation [61]. It consists of reusing processes from an SPL according to projects' requirements. The processes are bound to components that automate their execution. When the variability of a process to execute is not fully resolved, our approach consists of resolving this variability during the execution of this process. We illustrate our approach on industrial projects in a software company, as well as on a family of processes for designing and implementing modeling languages. Our approach promoted the identification of possible automated tasks for configuring IDEs and continuous integration, their reuse in various projects of the company, and the automation of their execution, while enabling to resolve process variability during the execution.

6.19. Towards Trust-Aware and Self-Adaptive Systems

The dynamic conditions under which Future Internet (FI) applications must execute call for self-adaptive software to cope with unforeseeable changes in the application environment. Software engineering currently provides frameworks to develop reasoning engines that support the runtime adaptation of distributed, heterogeneous applications. However, these frameworks have very limited support to address security concerns of these application, hindering their usage for FI scenarios. We address this challenge by enhancing self-adaptive systems with the concepts of trust and reputation [58]. Trust improves decision-making processes under risk and uncertainty, in turn improving security of self-adaptive FI applications.

6.20. SOA Antipatterns: an Approach for their Specification and Detection

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 in their architecture, called antipatterns. We introduce a novel and innovative approach supported by a framework for specifying and detecting antipatterns in SBSs [25]. We specify ten well-known and common antipatterns, including Multi Service and Tiny Service, and automatically generate their detection algorithms. We validate the detection algorithms in terms of precision and recall on two systems developed independently. This validation demonstrates that our approach enables the specification and detection of SOA antipatterns with an average precision of 90% and recall of 97.5%.

6.21. Automated Measurement of Models of Requirements

One way to formalize system requirements is to express them using the object-oriented paradigm. In this case, the class model representing the structure of requirements is called a requirements metamodel, and requirements themselves are object-based models of natural-language requirements. We show that such object-oriented requirements are well-suited to support a large class of requirements metrics[28]. We define a requirements metamodel and use an automated measurement approach proposed in our previous work to specify requirements metrics. We show that it is possible to integrate 78 metrics from 11 different papers in the proposed framework. The software that computes the requirements metric values is fully generated from the specification of metrics.

6.22. Empirical Evidence of Large-Scale Diversity in API Usage of Object-Oriented Software

In this paper, we study how object-oriented classes are used across thousands of software packages. We concentrate on "usage diversity'", defined as the different statically observable combinations of methods called on the same object. We present empirical evidence that there is a significant usage diversity for many classes. For instance, we observe in our dataset that Java's String is used in 2460 manners. We discuss the reasons of this observed diversity and the consequences on software engineering knowledge and research [56].

6.23. Efficient high-level abstractions for web programming

Writing large Web applications is known to be difficult. One challenge comes from the fact that the application's logic is scattered into heterogeneous clients and servers, making it difficult to share code between both sides or to move code from one side to the other. Another challenge is performance: while Web applications rely on ever more code on the client-side, they may run on smart phones with limited hardware capabilities. These two challenges raise the following problem: how to benefit from high-level languages and libraries making code complexity easier to manage and abstracting over the clients and servers differences without trading this ease of engineering for performance? In [59], we present high-level abstractions defined as deep embedded DSLs in Scala that can generate efficient code leveraging the characteristics of both client and server environments. We compare performance on client-side against other candidate technologies and against hand written low-level JavaScript code. Though code written with our DSL has a high level of abstraction, our benchmark on a real world application reports that it runs as fast as hand tuned low-level JavaScript code.

6.24. Exploring Optimal Service Compositions in Highly Heterogeneous and Dynamic Service-Based Systems

Service-oriented pervasive systems, composed of a large number of devices with heterogeneous capabilities where devices' resources are abstracted as software services, challenge the creation of high-quality composite applications. Resource heterogeneity, dynamic network connectivity, and a large number of highly distributed service providers complicate the process of creating applications with specific QoS requirements. Existing approaches to service composition control the QoS of an application solely by changing the set of participating concrete services which is not suitable for ad-hoc service-based systems characterised by high intermittent connectivity and resource heterogeneity. In [46], we propose a flexible way of formulating composition configurations suitable for such service-based systems. Our formulation proposes the combined consideration of the following factors that affect the QoS of a composed service: (a) service selection, (b) orchestration partitioning, and (c) orchestrator node selection. We show that the proposed formulation enables the definition of service composition configurations with 49% lower response time, 28% lower network latency, 36% lower energy consumption, and 13% higher success ratio compared to those defined with the traditional approach. In [45], we present the problem of efficiently exploring at runtime the serach space of possible configurations for a service orchestration with various Quality of Services.

7. Bilateral Contracts and Grants with Industry

7.1. VaryMDE

Participants: Benoit Combemale, Olivier Barais, Mathieu Acher, Jean-Marc Jézéquel, João Bosco Ferreira Filho, Suresh Pillay, David Mendez Acuna.

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. 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

7.3. Zenexity

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

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

7.4. Technology transfer

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.

Until october 2013 the DAUM platform was funded by an Inria Technology Development activity. In 2013 Triskell was granted for a specific funding (one year of engineer salary) to prepare the transfer of DAUM to the industry and prepare the creation of a startup focused on tools and applications for tactical information and decision systems on the field. The startup planned will involve seven members of the Triskell team.

8. Partnerships and Cooperations

8.1. National Initiatives

8.1.1. ANR GEMOC

Participants: Benoit Combemale, Didier Vojtisek, Olivier Barais, Arnaud Blouin, Erwan Bousse, 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 Herrera, Erwan Daubert, Walter Rudametkin Ivey.

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, Fabien Coulon, Valéria Lelli Leitão Dantas, 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 Projects

8.2.1.1. NeSSoS

Type: COOPERATION

Defi: Trustworthy ICT

Instrument: Network of Excellence

Objectif: Trustworthy ICT

Duration: October 2010 - March 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/

Inria contact: V. Issarny

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

8.2.1.2. DIVERSIFY

Type: COOPERATION

Defi: Foundation of Collaborative Adaptive Systems

Instrument: Specific Targeted Research Project

Objectif: NC

Duration: February 2013 - January 2016

Coordinator: Inria

Partner: SINTEF (Norway), Trinity College Dublin (Ireland), Université de Rennes 1 (France)

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

Inria contact: Benoît Baudry

Abstract: DIVERSIFY aims at favoring spontaneous diversification in software systems in order to increase their adaptive capacities. This objective is founded on three observations: software has to constantly evolve to face unpredictable changes in its requirements, execution environment or to respond to failure (bugs, attacks, etc.); the emergence and maintenance of high levels of diversity are essential to provide adaptive capacities to many forms of complex systems, ranging from ecological and biological systems to social and economical systems; diversity levels tend to be very low in software systems.

DIVERSIFY explores how the biological evolutionary mechanisms, which sustain high levels of biodiversity in ecosystems (speciation, phenotypic plasticity and natural selection) can be translated in software evolution principles. In this work, we consider evolution as a driver for diversity as a means to increase resilience in software systems. In particular, we are inspired by bipartite ecological relationships to investigate the automatic diversification of the server side of a client-server architecture. This type of software diversity aims at mitigating the risks of software monoculture. The consortium gathers researchers from the software-intensive, distributed systems and the ecology areas in order to transfer ecological concepts and processes as software design principles.

Triskell budget share: 500 keuros

8.2.1.3. HEADS

Type: COOPERATION

Defi:

Instrument: Specific Targeted Research Project

Objectif: NC

Duration: October 2013 - September 2016

Coordinator :SINTEF (Norway)

Partner: SINTEF (Norway), M2MZone (Ireland), TellU (Norway), SoftwareAG (Germany), ATC (Greece), Inria (France)

Inria contact: Benoît Baudry

Abstract: The idea of the HEADS project is to leverage model-driven software engineering and generative programming techniques to provide a new integrated software engineering approach which allow advanced exploitation the full range of diversity and specificity of the future computing continuum. The goal is to empower the software and services industry to better take advantage of the opportunities of the future computing continuum and to effectively provide new innovative services that are seamlessly integrated to the physical world making them more pervasive, more robust, more reactive and closer (physically, socially, emotionally, etc.) to their users.

Triskell budget share: 400 keuros

8.2.2. Collaborations in European Programs, except FP7

8.2.2.1. 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.

8.2.2.2. 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. Participation In other International Programs

8.3.1.1. 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) - Gerson SUNYE

Duration: Jul 2011 - Jun 2013

8.3.1.2. SPLIT

Program: International joint supervision of PhD agreement

Title: Aspect-oriented modeling and software product line

Inria principal investigator: Jean-Marc JEZEQUEL

International Partner (Institution - Laboratory - Researcher):

University of Luxembourg (Luxembourg) - Jean-Marc JEZEQUEL

Duration: Apr 2010 - Mar 2013

See also: http://www.fnr.lu/fr/Research-Programmes/Research-Programmes/Projects/Combine-Software-Product-Line-and-Aspect-Oriented-Software-Development-SPLIT

Combine Software Product Line and Aspect-Oriented Software Development - SPLIT

8.3.1.3. MBSAR

Program: CNRS PICS

Title: Model-Based Security Analysis at Runtime (MBSAR)

Inria principal investigator: Benoit Combemale

International Partner (Institution - Laboratory - Researcher):

Colorado State University (USA) - Software Assurance Lab - Robert B. France

Duration: Jan 2013 - Dec 2015

See also: http://gemoc.org/mbsar/

MBSAR develop model-based techniques for runtime analysis and enforcement of security policies in adaptive software systems.

8.4. International Research Visitors

8.4.1. Visits of International Scientists

8.4.1.1. Internships

Ioannis Kavvouras

Subject: Spontaneous diversification in software components

Date: from Mar 2013 until Jul 2013

Institution: Université Nationale Capodistrienne d'Athènes (Greece)

Marianela Ciolfi Felice

Subject: Draw your Products! A Model-based Approach

Date: from Mar 2013 until Aug 2013

Institution: National University of the Center of the Buenos Aires Province (Argentina)

Wuliang Sun

Subject: Synthesis of Feature-based Model Templates

Date: from Jun 2013 until Jul 2013

Institution: Colorado State University (United States)

8.4.2. Visits to International Teams

Participant: Arnaud Blouin.

Date: May 2013

Visited Institution: the laboratory for research on technology for ecommerce (LATECE) at the University of Québec at Montréal (UQÀM), Canada

8.4.3. Inria International Chair

Prof. Robert B. France [11] was granted by an Inria international chair for the period 2013-2017. Prof. France collaborate intensively with many members of the Triskell team on various joint work, e.g., the Familiar project and the GEMOC initiative. The Inria International Chair will allow Prof. France to make different long visits in the team along the period.

8.4.4. International initiative GEMOC

International initiative GEMOC

The GEMOC initiative (cf. http://www.gemoc.org) is an open and international initiative launched in 2013 that coordinate research partners worldwide to develop breakthrough software language engineering (SLE) approaches that support global software engineering through the use of multiple domain-specific languages. GEMOC members aim to provide effective SLE solutions to problems associated with the design and implementation of collaborative, interoperable and composable modeling languages.

The GEMOC initiative aims to provide a framework that facilitates collaborative work on the challenges of using of multiple domain-specific languages in software development projects. The framework consists of mechanisms for coordinating the work of members, and for disseminating research results and other related information on GEMOC activities. The framework also provides the required infrastructure for sharing artifacts produced by members, including publications, case studies, and tools.

The governance of the GEMOC initiative is ensured by the Advisory Board. The role of the Advisory Board is to coordinate the GEMOC work and to ensure proper dissemination of work products and information about GEMOC events (e.g., meetings, workshops).

Benoit Combemale is the co-founder and currently acts as principal coordinator of the GEMOC initiative. Benoit Combemale and Jean-Marc Jézéquel are part of the Advisory Board, and 9 Triskell members are part of the GEMOC initiative.

9. Dissemination

9.1. Scientific Animation

9.1.1. Journals

9.1.1.1. Benoit Baudry

is an Associate Editor of the following journal:

- Journal of Systems and Software (JSS)
- Journal on Software and System Modeling (SoSYM)

9.1.1.2. Jean-Marc Jézéquel

is an Associate Editor of the following journal:

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

[11] Colorado State University, USA. See. http://www.cs.colostate.edu/~france/

9.1.1.3. Noël Plouzeau

is chief editor of a special issue of the TSI French research journal, dedicated to software engineering.

9.1.2. Conferences

9.1.2.1. Benoit Baudry

has been chair for the program committee of ICST'13, the 6th International Conference on Software Testing Verification and Validation, Luxembourg, April 2013, and a member of the program committee of the following conferences:

- MODELS 2013 The 16th International Conference on Model Driven Engineering Languages and Systems Miami, USA, October 2013
- SEIP 2013 track, the Software Engineering in Practice track at ICSE'13, San Francisco, USA, May 2013
- AMT 2013, the 2nd International workshop on the Analysis of Model Transformations, Miami, USA, October 2013

9.1.2.2. Benoit Combemale

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

- APSEC'13, The 20th Asia-Pacific Software Engineering Conference, Thailand, 2013.
- SEFM'13, The 10th International Conference on Software Engineering and Formal Methods, Spain, 2013.
- ICMT'13, The 6th International Conference on Model Transformation, Hungary, 2013.
- ECMFA'13, The 9th European Conference on Modelling Foundations and Applications, Montpellier, 2013.
- QSIC13, The 13th International Conference on Quality Software, China, 2013.
- Educators' Symposium at MODELS'13, USA, 2013.
- AMT workshop on Analysis of Model Transformations (AMT), at MODELS'13, USA, 2013.
- CMA workshop on Comparing Modeling Approaches (CMA), at MODELS'13, USA, 2013.
- CMA workshop on Comparing Modeling Approaches (CMA), at RE'13, Brazil, 2013.
- VOLT workshop on Verification Of ModeL Transformation (VOLT), at ICMT'13, Hungary, 2013.

Benoit Combemale has also been an external reviewer for IEEE Transactions on Software Engineering, Journal of Software and Systems Modeling (SoSyM), and ICSE'13 (International Conference on Software Engineering).

9.1.2.3. Jean-Marc Jézéquel

has been a member of the program board (PB) or program committee (PC) of the following conferences:

- MODELS 2013, The 16th International Conference on Model Driven Engineering Languages and Systems Miami, USA, October 2013 (PB)
- ICSE 2013 The 35th International Conference on Software Engineering, San Francisco, USA, May 18-26, 2013 (PC)
- ICSE 2013 Doctoral Symposium San Francisco, USA, May 18-26, 2013 (PC)
- MiSE 2013 5th International Workshop on Modeling in Software Engineering, San Francisco, USA, May 18-26, 2013 (PC)
- VaMoS 2013 The Seventh International Workshop on Variability Modelling of Software-intensive Systems January 23-25, 2013 - CNR, Pisa, Italy (PC)

9.1.2.4. Olivier Barais

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

- SC'13, The 12th International Conference on Software Composition, Budapest, Hungary
- SEAA'13, The 39th Euromicro Conference on Software Engineering and Advanced Applications (SEAA 2013) Santander, Spain 2013
- SAC'13, The 28th Symposium On Applied Computing Coimbra, Portugal.
- VARICOMP'13, The 4th International Workshop on Variability & Composition @Modularity·AOSD, Fukuoka, Japan
- VAO'13, The 1st Workshop on View-Based, Aspect-Oriented and Orthographic Software Modelling, Montpellier, France
- WEITICE'2013, the 22nd IEEE International Conference on Collaboration Technologies and Infrastructure Hammamet, Tunisia
- CSMR2013 tools demo: the 17th European Conference on Software Maintenance and Reengineering tools demo track
- CIEL, the 2nd French Conference in Software Engineering. Nancy,

Olivier Barais has also been an external reviewer for IEEE Transactions on Software Engineering, Journal of Software and Systems Modeling (SoSyM), Journal of Software System (JSS), ACM TaaS (Transactions on Autonomous and Adaptive Systems), STVR (Software Testing, Verification and Reliability) and CBSE'13 (International Conference on Software Engineering).

9.1.2.5. Arnaud Blouin

has been external/sub reviewer for: IEEE Transactions on Software Engineering, Journal of Software and Systems Modeling (SoSyM), ICSE'13 (International Conference on Software Engineering), EICS'13 (ACM SIGCHI symposium on Engineering interactive computing systems), ICMT'13 (International Conference on Model Transformation), ASPEC'13 (Asia-Pacific Software Engineering Conference), and IHM'13 (French speaking conference on human-computer interaction).

9.1.2.6. Mathieu Acher

has organized the Journée ligne de produits (7th edition) in Paris (Sorbonne). He has been reviewer for ACM Surveys, International Journal on Software Tools for Technology Transfer (STTT), Journal of Systems and Software (JSS), Journal of Software and Systems Modeling (SoSyM), program committee for the Fifth International Workshop on Feature-Oriented Software Development (FOSD'13), the 1st International workshop on Reverse Variability Engineering (REVE'13).

9.1.2.7. Noël Plouzeau

has been program committee chair of the CIEL'2013 conference (the French conference on Software Engineering), Nancy, april 2013, and a member of the program committee of the CBSE 2013, Component Based Software Engineering, Seattle, USA, june 2013.

9.1.2.8. johann Bourcier

has been in the program committee of PECCS 2013 and an external reviewer for TaaS (Transactions on Autonomous and Adaptive Systems), and JSS (Journal of Systems and Software).

9.1.3. Workshops, Tutorials and Keynotes

9.1.3.1. Benoit Combemale

was an invited keynote speaker at CIEL'13 (the French Conference on Software Engineering) and gave a talk entitled "On Modeling and Testing When Unpredictability Becomes the Pattern" [12]. He also has been invited to give a talk at Neptune'13 (the French Industrial Conference on Model Driven Engineering) entitled

[12] Cf. http://people.irisa.fr/Benoit.Combemale/ciel2013

"On the Globalization of Modeling Languages"[13], at the french day organized by CNRS on MDE and Scientific Models[14] entitled "Language and Model Composition", at the workshop AOM (Barbados) entitled "On Language Composability", and at the Università degli Studi di Milano (Italy) entitled "Model-Driven Engineering for Software Language Engineering".

Benoit Combemale also has been co-chair of the workshops GlobalDSL'13 at ECOOP (cf. http://gemoc.org/globaldsl13, published on the ACM DL), and GEMOC'13 at MODELS (cf. http://gemoc.org/gemoc2013, published on CEUR).

9.1.4. Expertises

9.1.4.1. Benoit Baudry

is on the scientific advisory board of the Software Verification and Validation Lab from the Security and Trust research center in Luxembourg.

9.1.4.2. Arnaud Blouin

is a member of the Advisory Board of the FNR Core project MoDEL (2013-2015, Luxembourg) that works on the use of model-driven software product line for generating and testing user interfaces.

9.1.4.3. Benoit Combemale

is the coordinator of the international GEMOC initiative, and a member of its Advisory Board.

9.1.4.4. Jean-Marc Jézéquel

is a member of the Advisory Board of the NSF REMODD Project (Repository for Model Driven Development), and of the Advisory Board of the international GEMOC initiative.

9.2. Teaching - Supervision - Juries

9.2.1. 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 (from january 2013 to august 2013) and Noël Plouzeau (september 2013) are the managers of the Master2 Pro in Computer Science at the University of Rennes.

Johann bourcier is co-manager of the Home-Automation option at the ESIR engineering school in Rennes.

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

9.2.2. Supervision

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

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

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

[13] Cf. http://people.irisa.fr/Benoit.Combemale/neptune2013
[14] Cf. http://devlog.cnrs.fr/journee-idm-et-modeles-scientifiques

PhD : Aymeric Hervieu, *Model-based testing: handling performance in embedded software product lines*, thèse CIFRE avec Kereval 2010-2013, B. Baudry and A. Gotlieb

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

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

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

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

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 : Erwan Bousse, *Intégration et combinaison des techniques de V&V dans un contexte d'ingénierie cystème*, thèse, 2012-2015, 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 : João Bosco Ferreira Filho, *Variability modeling in software intensive systems*, thèse 2011-2014, O. Barais, B. Baudry and M. Acher

PhD in progress : Hamza Samih, *Extending model-based testing with variability and security management*, thèse CIFRE avec All4Tec, 2011-2014, B. Baudry

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

PhD in progress : Kwaku Yeboah-Antwi, *Runtime emergence of software diversity*, thèse 2013-2016, B. Baudry and O. Barais

PhD in progress : Sana Ben Nasr, *Modeling variability in regulatory requirements*, thèse 2013-2016, B. Baudry and M. Acher

PhD in progress : Mohamed Boussaa, *An Architecture for Testing Large-Scale Dynamic Distributed Systems*, thèse 2013-2016, B. Baudry and O. Barais

PhD in progress : Thomas Degueule, *Next Generation of MDE Tooling Support*, thèse 2013-2016, O. Barais and A. Blouin

PhD in progress : David Mendez Acuna, *Variability in Modeling Languages*, thèse 2013-2016, B. Baudry and B. Combemale

PhD in progress : Ivan Paez Anaya, *Proactive Adaptation in Pervasive Environment*, thèse 2012-2015, J.Bourcier, N. Plouzeau and J.-M. Jézéquel

PhD in progress : Inti Gonzalez Herrera, *Ressources reservation in Pervasive Middleware*, thèse 2012-2015, J. Bourcier, O. Barais

PhD in progress : Jacky Bourgeois, *Automatic Synchronization of Energy Production and Consumption*, thèse 2012-2015 co-tutelle avec the Open University, J.Bourcier, B. Baudry and G. Kortuem

PhD in progress : Francisco-JavierAcosta Padilla, *Automatic Deployment and Reconfigurations in Internet of Things*, thèse 2013-2016 co-encadrement avec l'équipe ACES, J. Bourcier, F. Weis

9.2.3. Juries

9.2.3.1. Benoit Baudry

was in the examination committee of the following PhD thesis:

- Walter Rudametkin Ivey, January 2013, Université de Grenoble (Referee)
- Amel Bennaceur, July 2013, Université Pierre et Marie Curie (Referee)
- Hajer Saada, December 2013, Université de Montpellier (Referee)
- Akram Ajouli, Septembre 2013, Université de Nantes (Member)

9.2.3.2. Benoit Combemale

was in the examination committee of the following PhD thesis:

- Clément Guy, December 2013, University of Rennes (Co-supervisor)
- Moussa Amrani, November 2013, University of Luxembourg (Member)
- Mounira Kezadri, July 2013, University of Toulouse (Member)
- Robert Guduvan, April 2013, University of Toulouse (Member)
- Laurent Wouters, January 2013, LIP6 (Member)

9.2.3.3. Noël Plouzeau

was in the examination committee of the following PhD thesis:

- François Fouquet, March 2013, University of Rennes (Co-supervisor)
- Nguyen Viet-Hoa, December 2013, University of Rennes (Co-supervisor)

9.2.3.4. Olivier Barais

was in the examination committee of the following PhD thesis:

- François Fouquet, March 2013, University of Rennes (Co-supervisor)
- Erwan Daubert, May 2013, University of Rennes (Co-supervisor)
- Pankesh Patel, November 2013, Université Pierre et Marie Curie (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° 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° 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° 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° 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

Doctoral Dissertations and Habilitation Theses

[12] S. CREFF. , *Une modélisation de la variabilité multidimensionnelle pour une évolution incrémentale des lignes de produits*, Université Rennes 1, December 2013, http://hal.inria.fr/tel-00926119

[13] E. DAUBERT. , *Adaptation et cloud computing : un besoin d'abstraction pour une gestion transverse*, INSA de Rennes, May 2013, http://hal.inria.fr/tel-00904364

[14] F. FOUQUET. , *Kevoree : Model@Runtime pour le developpement continu de syste`mes adaptatifs distribues heteroge`nes*, Université Rennes 1, March 2013, http://hal.inria.fr/tel-00831018

[15] C. GUY. , *Facilités de typage pour l'ingénierie des langages*, Université Rennes 1, December 2013, http://hal.inria.fr/tel-00917789

[16] A. HERVIEU. , *Approche à contraintes pour la sélection de Covering Array*, Université Rennes 1, December 2013, http://hal.inria.fr/tel-00915223

[17] P. ISTOAN. , *Methodology for the derivation of product behaviour in a Software Product Line*, Université Rennes 1 and Université du Luxembourg, February 2013, http://hal.inria.fr/tel-00925479

[18] P. ISTOAN. , *Méthodologie pour la dérivation comportementale de produits dans une ligne de produit logicielle*, Université Rennes 1 and université du Luxembourg, February 2013, http://hal.inria.fr/tel-00926141

[19] T. LE NHAN. , *Ingénierie dirigée par les modèles pour le provisioning d'images de machines virtuelles pour l'informatique en nuage*, Université Rennes 1, December 2013, http://hal.inria.fr/tel-00926228

[20] T. LE NHAN. , *Model-Driven Software Engineering for Virtual Machine Images Provisioning in Cloud Computing*, Université Rennes 1, December 2013, http://hal.inria.fr/tel-00923811

[21] V. H. NGUYEN. , *Une methode fondee sur les mode`les pour gerer les proprietes temporelles des syste`mes a` composants logiciels*, Université Rennes 1, December 2013, http://hal.inria.fr/tel-00923305

[22] N. SANNIER. , *INCREMENT une approche hybride pour modéliser et analyser dans le large les exigences réglementaires de sûreté*, Université Rennes 1, December 2013, http://hal.inria.fr/tel-00941881

Articles in International Peer-Reviewed Journals

[23] M. ACHER, A. CLEVE, P. COLLET, P. MERLE, L. DUCHIEN, P. LAHIRE. *Extraction and Evolution of Architectural Variability Models in Plugin-based Systems*, in "Software & Systems Modeling (SoSyM)", July 2013, 27 p. [*DOI :* 10.1007/S10270-013-0364-2], http://hal.inria.fr/hal-00859472

[24] 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", June 2013, vol. 78, n^o 6, pp. 657 - 681 [*DOI :* 10.1016/J.SCICO.2012.12.004], http://hal.inria.fr/hal-00767175

[25] P. FRANCIS, N. MATHIEU, M. NAOUEL, G. YANN-GAEL, B. BAUDRY, J.-M. JÉZÉQUEL. *SOA Antipatterns: an Approach for their Specification and Detection*, in "International Journal of Cooperative Information Systems", September 2013, http://hal.inria.fr/hal-00916299

[26] C. JARD, D. LIME, O. H. ROUX, L.-M. TRAONOUEZ. *Symbolic Unfolding of Parametric Stopwatch Petri Nets*, in "Formal Methods in System Design", May 2013, vol. 43, n^o 3, pp. 493-519 [*DOI :* 10.1007/S10703-013-0188-2], http://hal.inria.fr/hal-00840368

[27] J.-M. JÉZÉQUEL, B. COMBEMALE, O. BARAIS, M. MONPERRUS, F. FOUQUET. *Mashup of Meta-Languages and its Implementation in the Kermeta Language Workbench*, in "Software and Systems Modeling", 2013, http://hal.inria.fr/hal-00829839

[28] M. MONPERRUS, B. BAUDRY, J. CHAMPEAU, B. HOELTZENER, J.-M. JÉZÉQUEL. *Automated Measurement of Models of Requirements*, in "Software Quality Journal", 2013, vol. 21, n^o 1, pp. 3-22, WOS [*DOI :* 10.1007/S11219-011-9163-6], http://hal.inria.fr/hal-00646876

[29] Y. SUN, J. GRAY, R. DELAMARE, B. BAUDRY, J. WHITE. *Automating the Maintenance of Non-functional System Properties using Demonstration-based Model Transformation*, in "Journal of Software Maintenance and Evolution: Research and Practice", July 2013, http://hal.inria.fr/hal-00916287

[30] G. SUNYÉ, E. CUNHA DE ALMEIDA, Y. LE TRAON, B. BAUDRY, J.-M. JÉZÉQUEL. *Model-Based Testing of Global Properties on Large-Scale Distributed Systems*, in "Information and Software Technology", 2014, http://hal.inria.fr/hal-00942576

Articles in National Peer-Reviewed Journals

[31] B. COMBEMALE, J. DEANTONI, A. KOUDRI, J. LE NOIR. *Le nouveau défi de la coordination des langages de modélisation. Gestion de l'hétérogénéité des modèles dans le développement et l'exécution de systèmes logiciels complexes*, in "Génie logiciel", 2013, n^o 105, pp. 4-11, http://hal.inria.fr/hal-00854221

Invited Conferences

[32] P. COLLET, P. LAHIRE, M. ACHER, R. FRANCE. *Feature Model Management: Smart Operations and Language Support*, in "ACM/IEEE 16th International Conference on Model Driven Engineering Languages and Systems", Miami, United States, September 2013, http://hal.inria.fr/hal-00913157

International Conferences with Proceedings

[33] E. K. ABBASI, M. ACHER, P. HEYMANS, A. CLEVE. *Reverse Engineering Web Configurators*, in "17th European Conference on Software Maintenance and Reengineering (CSMR)", Antwerp, Belgium, IEEE, February 2014, http://hal.inria.fr/hal-00913139

[34] M. ACHER, B. COMBEMALE, P. COLLET, O. BARAIS, P. LAHIRE, R. FRANCE. *Composing your Compositions of Variability Models*, in "ACM/IEEE 16th International Conference on Model Driven Engineering Languages and Systems (MODELS'13)", Miami, United States, September 2013, vol. Lecture Notes in Computer Science, 17 p. , http://hal.inria.fr/hal-00859473

[35] M. ACHER, P. HEYMANS, A. CLEVE, J.-L. HAINAUT, B. BAUDRY. *Support for Reverse Engineering and Maintaining Feature Models*, in "VaMoS'13 - Seventh International Workshop on Variability Modelling of Software-Intensive Systems", Pisa, Italy, ACM, January 2013, http://hal.inria.fr/hal-00766786

[36] M. ACHER, R. E. LOPEZ-HERREJON, R. RABISER. *A Survey on Teaching of Software Product Lines*, in "Eight International Workshop on Variability Modelling of Software-Intensive Systems (VaMoS'14)", Nice, France, ACM, January 2014, pp. 1-8 [*DOI :* 10.1145/2556624.2556629], http://hal.inria.fr/hal-00916746

[37] B. BAUDRY, M. MONPERRUS, C. MONY, F. CHAUVEL, F. FLEUREY, S. CLARKE. *DIVERSIFY - Ecology-inspired software evolution for diversity emergence*, in "CSMR", Antwerp, Belgium, IEEE, February 2014, pp. 395-398, http://hal.inria.fr/hal-00916281

[38] O. BEAUDOUX, M. CLAVREUL, A. BLOUIN. *Binding Orthogonal Views for User Interface Design*, in "Proceedings of the 1st Workshop on View-Based, Aspect-Oriented and Orthographic Software Modelling (VAO'13)", Montpellier, France, ACM press, 2013, in press, http://hal.inria.fr/hal-00826705

[39] S. BENSALEM, M. BOZGA, B. BOYER, A. LEGAY. *Incremental Generation of Linear Invariants for Component-Based Systems*, in "3th International Conference on Application of Concurrency to System Design (ACSD)", Barcelona, Spain, 2013, pp. 1-10, http://hal.inria.fr/hal-00878738

[40] M. BIAZZINI, P. SERRANO-ALVARADO, R. CARVAJAL-GOMEZ. *Towards improving user satisfaction in decentralized P2P networks*, in "COLLABORATECOM", United States, October 2013, 1 p. , http://hal.inria.fr/hal-00871672

[41] B. COMBEMALE, J. DEANTONI, R. FRANCE, F. BOULANGER, S. MOSSER, M. PANTEL, B. RUMPE, R. SALAY, M. SCHINDLER. *Report on the First Workshop On the Globalization of Modeling Languages (GEMOC 2013)*, in "GEMOC+AMINO@MoDELS: GEMOC - 1st International Workshop On the Globalization of Modeling Languages (2013)", Miami, United States, CEUR, 2013, vol. Vol-1102, pp. 3-13, http://hal.inria.fr/hal-00926244

[42] B. COMBEMALE, J. DEANTONI, M. VARA LARSEN, F. MALLET, O. BARAIS, B. BAUDRY, R. FRANCE. *Reifying Concurrency for Executable Metamodeling*, in "SLE - 6th International Conference on Software Language Engineering", Indianapolis, IN, United States, M. ERWIG, R. F. PAIGE, E. VAN WYK (editors), Lecture Notes in Computer Science, Springer, 2013, vol. 8225, pp. 365-384 [*DOI :* 10.1007/978-3-319-02654-1_20], http://hal.inria.fr/hal-00850770

[43] B. COMBEMALE, C. HARDEBOLLE, C. JACQUET, F. BOULANGER, B. BAUDRY. *Bridging the Chasm between Executable Metamodeling and Models of Computation*, in "5th International Conference, SLE 2012",

Desden, Germany, K. CZARNECKI, G. HEDIN (editors), Lecture Notes in Computer Science, Springer, February 2013, vol. 7745, pp. 184-203, ISBN: 978-3-642-36088-6, http://hal.inria.fr/hal-00905067

[44] J.-M. DAVRIL, E. DELFOSSE, N. HARIRI, M. ACHER, J. CLELANG-HUANG, P. HEYMANS. *Feature Model Extraction from Large Collections of Informal Product Descriptions*, in "European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering (ESEC/FSE'13)", Saint Petersburg, Russian Federation, August 2013, pp. 290-300 [*DOI :* 10.1145/2491411.2491455], http://hal.inria.fr/hal-00859475

[45] D. EFSTATHIOU, P. MCBURNEY, S. ZSCHALER, J. BOURCIER. *Exploring Optimal Service Compositions in Highly Heterogeneous and Dynamic Service-Based Systems*, in "Search Based Software Engineering", Saint Petersbourg, Russian Federation, August 2013, http://hal.inria.fr/hal-00859877

[46] D. EFSTATHIOU, P. MCBURNEY, S. ZSCHALER, J. BOURCIER. *Flexible QoS-Aware Service Composition in Highly Heterogeneous and Dynamic Service-Based Systems*, in "WiMob - The 9th IEEE International Conference on Wireless and Mobile Computing, Networking and Communications - 2013", Lyon, France, October 2013, http://hal.inria.fr/hal-00859891

[47] M. FAUNES, J. CADAVID, B. BAUDRY, H. SAHRAOUI, B. COMBEMALE. *Automatically Searching for Metamodel Well-Formedness Rules in Examples and Counter-Examples*, in "MODELS - ACM/IEEE 16th International Conference on Model Driven Engineering Languages and Systems", Miami, United States, Lecture Notes in Computer Science, Springer, 2013, pp. 187-202, http://hal.inria.fr/hal-00923789

[48] M. FELICE CIOLFI, J. B. FERREIRA FILHO, M. ACHER, A. BLOUIN, O. BARAIS. *Interactive Visualisation of Products in Online Configurators: A Case Study for Variability Modelling Technologies*, in "MAPLE/SCALE 2013: Joint Workshop on Model-driven Approaches in SPL Engineering, and Workshop on Scalable Modeling Techniques for SPL", Tokio, Japan, August 2013, http://hal.inria.fr/hal-00842656

[49] *Best Paper*
J. B. FERREIRA FILHO, O. BARAIS, M. ACHER, J. LE NOIR, B. BAUDRY. *Generating Counterexamples of Model-based Software Product Lines: An Exploratory Study*, in "SPLC - 17th International Software Product Line Conference", Tokyo, Japan, August 2013, http://hal.inria.fr/hal-00837523.

[50] T. GENET, T. LE GALL, A. LEGAY, V. MURAT. *Tree Regular Model Checking for Lattice-Based Automata*, in "CIAA - 18th International Conference on Implementation and Application of Automata", Halifax, Canada, LNCS, Springer, 2013, vol. 7982, http://hal.inria.fr/hal-00924849

[51] E. KHALIL ABBASI, A. HUBAUX, M. ACHER, Q. BOUCHER, P. HEYMANS. *The Anatomy of a Sales Configurator: An Empirical Study of 111 Cases*, in "CAiSE'13 - 25th International Conference on Advanced Information Systems Engineering - 2013", Valencia, Spain, M. NORRIE, C. SALINESI (editors), June 2013, vol. 7908, pp. 162-177, http://hal.inria.fr/hal-00796555

[52] S. KOUNEV, S. RIZOU, S. ZSCHALER, S. ALEXAKIS, T. BURES, J.-M. JÉZÉQUEL, D. KOURTESIS, S. PANTELOPOULOS. *RELATE: a research training network on engineering and provisioning of service-based cloud applications*, in "Proceedings of the 2013 international workshop on Hot topics in cloud services", Prague, Czech Republic, S. KOUNEV, S. ZSCHALER, K. SACHS (editors), ACM, 2013, pp. 51-54 [*DOI :* 10.1145/2462307.2462319], http://hal.inria.fr/hal-00921120

[53] M. KRAMER, J. KLEIN, J. R. STEEL, B. MORIN, J. KIENZLE, O. BARAIS, J.-M. JÉZÉQUEL. *Achieving Practical Genericity in Model Weaving through Extensibility*, in "Theory and Practice of Model Transformations - 6th International Conference", Budapest, Hungary, K. DUDDY, G. KAPPEL (editors), Springer, 2013, pp. 108-124 [*DOI :* 10.1007/978-3-642-38883-5_12], http://hal.inria.fr/hal-00921058

[54] V. LELLI. *Challenges of Testing for Critical Interactive Systems*, in "International Conference on Software Testing, Verification and Validation", Luxembourg, March 2013, http://hal.inria.fr/hal-00804876

[55] D. MARIJAN, A. GOTLIEB, S. SEN, A. HERVIEU. *Practical Pairwise Testing for Software Product Lines*, in "SPLC 2013", Tokyo, Japan, August 2013, http://hal.inria.fr/hal-00859438

[56] D. MENDEZ, B. BAUDRY, M. MONPERRUS. *Empirical Evidence of Large-Scale Diversity in API Usage of Object-Oriented Software*, in "International Conference on Source Code Analysis and Manipulation (SCAM'2013)", Netherlands, 2013, 10 p. , http://hal.inria.fr/hal-00844753

[57] M. MONPERRUS, B. BAUDRY. *Two Flavors in Automated Software Repair: Rigid Repair and Plastic Repair*, in "Dagstuhl Seminar n°13061 "Fault Prediction, Localization, and Repair"", Germany, 2013, 5 p. , http://hal.inria.fr/hal-00844735

[58] F. MOYANO, B. BAUDRY, J. LOPEZ. *Towards Trust-Aware and Self-Adaptive Systems*, in "IFIPTM", Malaga, Spain, June 2013, http://hal.inria.fr/hal-00916280

[59] J. RICHARD-FOY, O. BARAIS, J.-M. JÉZÉQUEL. *Efficient high-level abstractions for web programming*, in "Generative Programming: Concepts and Experiences, GPCE'13", Indianapolis, United States, J. JARVI, C. KASTNER (editors), ACM, 2013, pp. 53-60 [*DOI :* 10.1145/2517208.2517227], http://hal.inria.fr/hal-00920786

[60] E. ROUILLÉ, O. BARAIS, B. COMBEMALE, T. DAVID, J.-M. JÉZÉQUEL. *Improving Reusability in Software Process Lines*, in "Euromicro Conference on Software Engineering and Advanced Applications", Santander, Spain, September 2013, http://hal.inria.fr/hal-00838771

[61] E. ROUILLÉ, B. COMBEMALE, O. BARAIS, T. DAVID, J.-M. JÉZÉQUEL. *Integrating Software Process Reuse and Automation*, in "Asia-Pacific Software Engineering Conference (APSEC)", Bangkok, Thailand, December 2013, http://hal.inria.fr/hal-00872188

[62] N. SANNIER, M. ACHER, B. BAUDRY. *From Comparison Matrix to Variability Model: The Wikipedia Case Study*, in "28th IEEE/ACM International Conference on Automated Software Engineering", Palo Alto, United States, IEEE, 2013, http://hal.inria.fr/hal-00858491

[63] E. VACCHI, W. CAZZOLA, S. PILLAY, B. COMBEMALE. *Variability Support in Domain-Specific Language Development*, in "SLE - 6th International Conference on Software Language Engineering", Indianapolis, IN, United States, M. ERWIG, R. F. PAIGE, E. VAN WYK (editors), Lecture Notes in Computer Science, Springer, 2013, vol. 8225, pp. 76-95 [*DOI :* 10.1007/978-3-319-02654-1_5], http://hal.inria.fr/hal-00914715

[64] S. WULIANG, B. COMBEMALE, S. DERRIEN, R. FRANCE. *Using Model Types to Support Contract-Aware Model Substitutability*, in "9th European Conference on Modelling Foundations and Applications (ECMFA 2013)", Montpellier, France, P. VAN GORP, T. RITTER, L. ROSE (editors), LNCS, Springer-Verlag Berlin Heidelberg, 2013, vol. 7949, pp. 118-133 [*DOI :* 10.1007/978-3-642-39013-5_9], http://hal.inria.fr/hal-00808770

National Conferences with Proceedings

[65] O. BARAIS, B. BAUDRY, A. BLOUIN, B. COMBEMALE, J.-M. JÉZÉQUEL, D. VOJTISEK. *A Demonstration for Building Modular and Efficient DSLs: The Kermeta v2 Experience*, in "Conférence en IngénierieE du Logiciel (CIEL)", Nancy, France, April 2013, http://hal.inria.fr/hal-00796009

[66] G. BÉCAN, S. BEN NASR, B. BAUDRY. *An Ontologic-Aware Feature Modeling Environment*, in "Journées lignes de produits - 2013", Paris, France, November 2013, http://hal.inria.fr/hal-00915662

[67] N. SANNIER, G. BÉCAN, S. BEN NASR, B. BAUDRY. *On Product Comparison Matrices and Variability Models from a Product Comparison/Configuration Perspective*, in "Journée lignes de produits - 2013", Paris, France, November 2013, http://hal.inria.fr/hal-00915657

Scientific Books (or Scientific Book chapters)

[68] A. HUBAUX, M. ACHER, T. T. TUN, 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, August 2013, pp. 3-28 [*DOI :* 10.1007/978-3-642-36654-3_1], http://hal.inria.fr/hal-00767213

Books or Proceedings Editing

[69] B. COMBEMALE, W. CAZZOLA, R. B. FRANCE (editors). , *GlobalDSL '13: Proceedings of the First Workshop on the Globalization of Domain Specific Languages*, ACM, 2013, 33 p. , http://hal.inria.fr/hal-00916292

[70] B. COMBEMALE, J. DEANTONI, R. B. FRANCE, B. BARN, T. CLARK, U. FRANK, V. KULKARNI, D. TURK (editors). , *Joint Proceedings of GEMOC '13 and AMINO '13 @ MODELS 2013*, CEUR-WS, 2013, vol. 1102, 88 p. , http://hal.inria.fr/hal-00916282

Research Reports

[71] A. ARNOLD, B. BOYER, A. LEGAY. , *Contracts and Behavioral Patterns for Systems of systems: The EU IP DANSE approach*, January 2013, 21 p. , http://hal.inria.fr/hal-00778039

[72] G. BÉCAN, M. ACHER, B. BAUDRY, S. BEN NASR. , *Breathing Ontological Knowledge Into Feature Model Management*, Inria, October 2013, n° RT-0441, 15 p. , http://hal.inria.fr/hal-00874867

[73] B. DELAHAYE, A. LEGAY, S. SEDWARDS. , *A Simple and Efficient Statistical Model Checking Algorithm to Evaluate Markov Decision Processes*, September 2013, http://hal.inria.fr/hal-00856704

[74] D. EL KATEB, F. FOUQUET, J. BOURCIER, Y. LE TRAON. , *Artificial Mutation inspired Hyper-heuristic for Runtime Usage of Multi-objective Algorithms*, November 2013, http://hal.inria.fr/hal-00948329

References in notes

[75] D. BENAVIDES, S. SEGURA, A. RUIZ-CORTES. *Automated Analysis of Feature Models 20 years Later: a Literature Review*, in "Information Systems", 2010, vol. 35, n° 6

[76] T. BERGER, R. RUBLACK, D. NAIR, JOANNE M. ATLEE, M. BECKER, K. CZARNECKI, A. WASOWSKI. *A survey of variability modeling in industrial practice*, in "VaMoS'13", ACM, 2013

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

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

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

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

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

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

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

[84] 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

[85] 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

[86] 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

[87] 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

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

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