Actes des journées du GDR GPL 2023

Groupement de Recherche « Génie de la Programmation et du Logiciel »

Rennes, 5-8 juin 2023

Comité d’organisation

- Olivier Barais
- Mireille Blay-Fornarino
- Benoît Combemale
- Catherine Dubois
- Djamel E. Khelladi
- Nicolas Magaud
- Pascal Poizat
- Responsables des groupes de travail et d’AFADL

Editeurs

- Mireille Blay-Fornarino
- Catherine Dubois
- Nikolai Kosmatov

Préface

C'est avec grand plaisir que nous vous avons accueillis pour les quatorzièmes journées nationales du GDR « Génie de la Programmation et du Logiciel »(GPL) qui se sont déroulées à Rennes du 5 au 8 juin 2023. Les journées nationales, temps fort de l'activité de notre GDR, donnent l'occasion de nous enrichir des travaux récents présentés, d'échanger notamment entre groupes de travail, de découvrir de nouvelles problématiques et de permettre aux jeunes chercheurs de s'ouvrir à la richesse de notre communauté dans un contexte convivial.

Ces journées étaient co-localisées avec la 9ème édition de la conférence internationale ICT4S, International Conference on Information and Communications Technology for Sustainability, ainsi que la 22ème édition de AFADL, l'atelier francophone sur les Approches Formelles dans l'Assistance au Développement de Logiciels.

Nous avons accueilli, dans l'ordre, les cinq conférenciers invités suivants :

- Benoit Baudry (Professeur au KTH Royal Institute of Technology, Stockholm) a présenté une conférence intitulée « The Software Supply Chain » ;
- Hugues Ferreboeuf (The Shift Project) a donné une conférence intitulée « Sustainable Digitalization : Why we need to shift away from Big Tech business models », conférence invitée partagée avec ICT4S ;
- Sébastien Bardin (CEA), invité partagé avec AFADL, a mis l'accent sur la sécurité dans son exposé « Revisiting Program Analysis through the Security Lens » ;
- Sandrine Blazy (Université de Rennes et IRISA) a présenté ses travaux dans un exposé intitulé « How to provide proof that software is bug-free ? Verified compilation to the rescue », travaux récompensés par la médaille d'argent 2023 du CNRS ;
- enfin, Jean Jousel (CEA, médaille d'or du CNRS 2002), invité commun avec ICT4S, nous a invités à réfléchir sur le thème « Global warming : The need for a new model of development and the key role for higher education ».

En plus de ces conférences invitées, les groupes de travail du GDR GPL ont animé dix sessions, dont deux partagées avec l'atelier AFADL.

Le GDR GPL a à cœur de mettre à l'honneur les jeunes chercheurs. C'est pourquoi nous décernons, depuis onze ans, un prix de thèse du GDR. Le prix de thèse GPL 2022 a été attribué à Paulo Emilio de Vilhena pour sa thèse intitulée « Preuve de programmes avec effect handlers ». Un accessit a été décerné à Faezeh Khorram pour sa thèse intitulée « Un environnement de test pour les langages dédiés exécutables ». Ils ont tous deux présenté leurs travaux lors d'une session

plénière dédiée. Le jury chargé de sélectionner les lauréats a été présidé par Pascal Poizat, que nous remercions vivement, ainsi que l'ensemble des membres du jury.

Nous remercions Nicolas Magaud qui a pris la responsabilité, pour la deuxième année consécutive, de la session posters et outils, toujours riche en discussions. A la fin des journées, le prix du meilleur poster des journées GPL 2023 a été attribué à Mamy Razafintsialonina (CEA LIST) pour son poster intitulé « Comment améliorer l'efficacité de l'analyse statique des programmes ? ».

Avant de clôturer cette préface, nous tenons à remercier tous ceux qui ont contribué à l'organisation de ces journées nationales, et tout particulièrement Olivier Barais, Benoit Combemale et Djamel E. Khelladi, sans qui ces journées n'auraient pas été possibles, les responsables de groupes de travail et les membres du comité de direction du GDR GPL. Nous remercions Nikolai Kosmatov qui a pris en main, une fois encore, les actes des journées nationales.

Mireille Blay-Fornarino et Catherine Dubois
Co-directrices du GDR « Génie de la Programmation et du Logiciel »

Table des matières

Conférences invités

Revisiting Program Analysis through the Security Lens

Sébastien Bardin * [1]

[1] CEA LIST – Univ. Paris-Saclay – France

Symbolic Execution emerged in the mid-2000 and was rapidly adopted by the research community as a tool of choice for bug hunting. In this talk, we consider security concerns and binary-level vulnerability issues. We will show some challenges symbolic execution faces in this field of application, and report on several results and achievements carried out within the BINSEC group to adapt Symbolic Execution to these challenges. We will especially focus on the problems of robust reachability (trying to define and find meaningful bugs) and adversarial reachability (considering an active code-level attacker).

*Intervenant

Software Supply Chain

Benoît Baudry * [1]

[1] KTH Royal Institute of Technology – Suède

Once an idealistic concept, software reuse is now a major success! Open source software, package managers, build systems all contribute to fueling large-scale reuse to develop robust applications. They are so successful that application binaries are now essentially composed of third-party code. This observation and a few high-profile attacks have let to the emergence of a new concept: the software supply chain.
This talk explores this new concept as well as the research opportunities that it opens, at the intersection of software engineering and software security. Code integrity and specialization, software composition analysis and reproducible builds are great challenges for suture software research.

*Intervenant

How to provide proof that software is bug-free? Verified compilation to the rescue

Sandrine Blazy * [1]

[1] IRISA – Université de Rennes – France

Deductive verification provides very strong guarantees that software is bug-free. Since the verification is usually done at the source level, the compiler becomes a weak link in the production of software. Verifying the compiler itself provides guarantees that no errors are introduced during compilation. This talk will illustrate this approach through CompCert, the first fully verified compiler for C that is actually usable on real source code and that produces decent target code on real-world architectures. More generally, this approach opens the way to the verification of software tools involved in the production and verification of software.

*Intervenant

Sustainable digitalization: Why we need to shift away from Big Tech business models

Hugues Ferreboeuf * [1]

[1] The Shift Project – - – France

The unsustainable growth of the digital environmental footprint over the last decade is due to the rapid growth of digital "volumes", which have outpaced efficiency gains. This growth is a result of the business models of the dominant players of the digital economy, namely the largest digital platforms (GAFAM, BATX) and their derivatives (Netflix, TikTok etc.). To make digitalization sustainable, a transition away from these digital superpowers is necessary, and alternative, sustainable business models must be adopted. The initiation and acceleration of this transition requires the implementation of appropriate public policies that will reduce the attractiveness of current Big Tech business models and promote and support the development of a new digital economy.

*Intervenant

Global Warming: The need for a new model of development and the key role of higher education

Jean Jouzel * [1]

[1] Scientific Working Group of the Intergovernmental Panel on Climate Change (IPCC) – - – France

If nothing was done to manage the increase of the greenhouse effect tied to human activity, we should see, at the end of this century, a minimum average 4 ∘C increase in temperature worldwide which will increase well beyond 2100. The impacts of such a " business as usual " scenario would be difficult if not impossible to handle. And these difficulties will hold true for a +3 ∘C climate change a level which could be reached in the current context of the Paris agreement. After briefly examining the causes and consequences of this ongoing global warming in the light of recent IPCC reports, we will conclude on the absolute need to keep global warming well below 2∘C, and much better around 1.5∘C, if we want today young generations be able to adapt to future climate change in the second part of this century and beyond. We will argue that research, innovation and creativity are essential for going towards this low carbon society but that this " ecological transition " also requires large dedicated teaching efforts in higher education and all along our life.

*Intervenant

Prix de Thèse

Prix de thèse du GDR GPL 2022

Pascal Poizat * [1]

[1] LIP6 – U. Paris-Nanterre, Université Paris IV - Paris Sorbonne – France

Créé en 2013 pour récompenser chaque année une excellente thèse préparée au sein du GDR GPL, le Prix de thèse du GDR GPL a pour objectif de promouvoir les travaux du GDR GPL auprès de la communauté informatique.

Le prix est décerné par un jury couvrant les thématiques du GDR GPL. Pour l'édition concernant les thèses soutenues en 2022, le jury est présidé par Pascal Poizat et est constitué des membres suivants : R. Ameur-Boulifa, E. Cariou, S. Chabridon, S. Costiou, J. Deantoni, Th. Degueule, D. Delahaye, A. Etien, J.-L. Giavitto, A. Giorgetti, L. Henrio, A. Hurault, N. Kosmatov, R. Laleau, M. Lhommeau, P.-E. Moreau, S. Mosser, A. Noureddine, S. Sadou.

Le prix de thèse 2022 a été délivré à Paulo Emilio DE VILHENA pour sa thèse intitulée "Preuves de programmes avec effect handlers" préparée à l'Université Paris Cité / INRIA sous la direction de François Pottier.

L'accesit a été délivré à Faezeh KHORRAM pour sa thèse intitulée "A testing framework for executable domain-specific languages" préparée à l'IMT Atlantique / LS2N sous la direction de Gerson Sunyé.

*Intervenant

Nouvelles du CNRS

Nouvelles du CNRS

Olivier Serre * [1]

[1] CNRS – INS2I – France

Olivier Serre présente les nouvelles du CNRS et répond aux questions des participants.

*Intervenant

GT IDM

Ongoing Work on Domain-Specific Modeling for Early Design Evaluation with the Help of Formal Methods for Correctness and Completeness Guarantees

Gurvan Le Guernic *† [1], Hubert Godfroy [2], Pierre Kimmel [2], Abdelghani Alidra [3], Antoine Beugnard [3,4]

[1] DGA Maîtrise de l'information Université de Rennes (DGA.MI) – Direction générale de l'Armement (DGA) – Route de Laillé. La Roche Marguerite - 35170 - Bruz, France
[2] Capgemini – Capgemini – France
[3] Département Informatique (IMT Atlantique - INFO) – IMT Atlantique – IMT Atlantique - Campus de Brest - Technopôle Brest-Iroise CS 8381829238 BREST Cedex 3, France
[4] Lab-STICC – Lab-STICC UMR CNRS 6285, Brest – France

The DGA is involved in the development of sensitive devices that may require the evaluation by the contracting authority (MOA) of early design decisions made by the prime contractor (MOE). The exchange of information for this evaluation is traditionally based on documents (bearing resemblance to Common Criteria documentation) and face-to-face meetings. DGA experiments on using Model-Based System Engineering and Formal Methods to improve the correctness and completeness of information exchanged at this stage, in order to improve the efficiency and quality of this early design evaluation. This talk will first quickly introduce the audience to the Network Pump of the NRL, a realistic use case representative of the type of sensitive devices dealt with by the DGA, and for which a large amount of information is openly accessible. It will then present the objectives and current state of an ongoing project by the DGA to develop a dedicated (domain specific) modeling environment for this task of early design evaluation. The current prototype allows modeling different "views" of the early design for which the tooling provides various algorithms providing guarantees regarding the completeness and correctness of those views. The talk will conclude by presenting the limitations and open questions in the current state of this work.

*Intervenant
†Auteur correspondant: gurvan.le_guernic@inria.fr

FaST: A Model-Driven Framework For Efficient Visualization Of Large-Scale Time Series

Manele Ait Habouche *† [1], Mickaël Kerboeuf [1], Guillou Goulven [1], Jean-Philippe Babau [1]

[1] Université de Bretagne Occidentale - UFR Sciences et Techniques (UBO UFR ST) – Université de Brest, Lab-STICC UMR CNRS 6285, Brest – 6, avenue Victor Le Gorgeu - CS93837 29238 Brest Cedex 3, France

Scientists who analyze physical phenomena typically work with data gathered from various sensors placed throughout the environment. Visualizing massive amount of data is crucial to the analysis process, but implementing a reliable and efficient visualization tool can be challenging, especially without the support of a large-scale platform.
We propose to take up this challenge with FaST (an efficient model-driven Framework for visualizing large-Scale Time series).

FaST is a model-driven framework that provides a complete solution for the storage, the querying and the visualization of time series in a big data context. It offers a dedicated language for data scientists to efficiently specify the solution's architecture and the data it has to handle. The deployment process is streamlined through code generation and server-side dockerization. The generated tool itself is optimized for performance through ad hoc optimizations. On the server-side, these optimizations involve pre-computation of views based on the Min-Max principle. On the client-side, they come from the anticipation of queries related to the data navigation abilities of the generated tool.
Our current work focuses on a data collection system that involves a USV (Unmanned Surface Vehicle) equipped with various sensors, which is designed to cater to a group of users seeking to monitor the sensors' data. To achieve this, we propose a generic model-driven framework based on the publish-subscribe pattern that enables efficient transmission of the emitted data to the users.

*Intervenant
†Auteur correspondant: Manele.AitHabouche@univ-brest.fr

Model-Aided Engineering of Cyber-Physical and Socio-Technical Systems

Thuy Nguyen *† [1]

[1] EDF – EDF Recherche et Développement – France

Les systèmes cyber-physiques (CPS) et socio-techniques (STS, avec également des aspects humains et organisationnels) ont des différences importantes avec le logiciel. Leur ingénierie doit en tenir compte sous peine de sérieuses déconvenues. En particulier, la spécification de leurs exigences ne peut pas se calquer sur celle des logiciels. Pour ces derniers, la spécification est une donnée d'entrée ou au mieux une première phase. Pour les CPS-STS, c'est une activité permamente du processus de conception, voire du cycle de vie. Une erreur grave mais fréquente consiste aussi à confondre spécification du sysème et spécification du logiciel. Enfin, alors qu'on parle d'Ingénierie Dirigée par les Modèles (IDM) en logiciel avec en bout de course, la génération automatique de code, ce ne peut pas être le cas pour les CPS-STS. Pour eux, il convient plutôt de parler d'Ingénierie Assistée par les Modèles, l'objectif premier des modèles étant d'aider à la vérification des exigences et solutions (par la simulation et la vérification formelle), tout en étant conscient qu'un modèle n'est qu'une approximation, et n'est pas toujours juste.

*Intervenant

†Auteur correspondant:

Collaborative Security-by-Design Platform with Model-Driven Engineering approach

Othmane El Karmy *† [1], Sophie Ebersold [1], Nan Messe [1], Mahmoud Nassar [2], Mahmoud El Hamlaoui [2]

[1] Université Toulouse - Jean Jaurès (UT2J) – Université de Toulouse, IRIT - UMR 5505 – 5 allées Antonio Machado - 31058 Toulouse Cedex 9, France
[2] Univ. Mohammed V de Rabat / ENSIAS – Maroc

Le développement de logiciels est exposé à des risques de sécurité croissants. Pour faire face à ces défis, il est important d'adopter une approche proactive et assurer une collaboration entre les développeurs, les experts en sécurité et les utilisateurs est essentielle pour renforcer la sécurité des systèmes, cela comprend la modélisation des menaces (threat modeling) pour identifier les vulnérabilités potentielles et les scénarios d'attaque, une approche basée sur l'utilisation de l'ingénierie dirigée par les modèles (MDE).

*Intervenant
†Auteur correspondant:

GT IE

Fouille des avis d'applications bilingues avec des modèles pré-entraînés et ChatGPT

Jialiang Wei * [1]

[1] IMT Mines Ales – IMT Mines Alès – France

Les avis d'utilisateurs sur les applications provenant des "stores" (Google Play Store, App Store...) peuvent être intéressants pour améliorer les exigences des logiciels, y compris de logiciels similaires. Un grand nombre d'avis précieux sont continuellement publiés, décrivant des rapports d'erreur et des demandes de fonctionnalités. L'utilisation efficace des avis des utilisateurs nécessite l'extraction d'informations pertinentes. En raison du volume important d'avis d'utilisateurs, l'analyse manuelle est ardue. Diverses approches basées sur le traitement du langage naturel ont été proposées pour l'extraction automatique d'avis d'utilisateurs. Dans cette exposé, nous allons présenter notre approche d'analyse des avis des utilisateurs basés sur les modèles pré-entraînés, et voir comme ceci peut enrichir ou corriger le cahier des cahier des charges initial d'une application similaire.

*Intervenant

Exigences et conception globale de l'avion et sa chaîne d'assemblage

Anouck Chan * [1]

[1] ONERA / DTIS / Université de Toulouse – ONERA – France

Chaque famille d'avion dispose d'un système industriel dédié. Cela est dû aux spécificités des techniques d'assemblage, matériaux et technologies qu'elles utilisent. Dès lors, il est nécessaire de concevoir un système industriel spécifique pour chaque famille d'avion.
Des travaux cherchent à proposer une aide à la conception de tels systèmes en s'appuyant sur des techniques d'intelligence artificielle et de recherche opérationnelle. Cependant, avant de concevoir ces outils, il est nécessaire d'éliciter, à la fois les objectifs, les enjeux et les contraintes de chaque acteur impliqué.
Pour cela nous avons utilisé une approche orientée buts afin d'éliciter les exigences d'un outil d'aide à la conception d'une chaîne d'assemblage d'avion. Fort de cette expérience, nous avons développé une approche générique, orientée buts, consistant à raffiner les buts abstraits de haut niveau d'une organisation, en buts satisfaisables par des acteurs de cette même organisation. Cette méthode est appliquée sur un cas industriel aéronautique.

*Intervenant

Les défauts dans la spécification du logiciel de commande du LGS (Landing Gear System)

Thuy Nguyen * [1]

[1] EDF – EDF – France

Il est humainement impossible d'écrire un logiciel complet sans aucune erreur, et une bonne partie du génie logiciel vise donc à détecter ces erreurs. Il en va de même pour la spécification des systèmes : l'expérience montre qu'elle contient presque toujours des défauts rédhibitoires qui ne sont révélés que tard dans le processus de développement, voire même en exploitation, avec des conséquences parfois catastrophiques. Avec la méthode BASAALT (Behaviour Analysis and Simulation All Along systems Life Time), la formalisation des exigences ne vise pas seulement à fournir une référence rigoureusement définie à la conception et la réalisation, mais aussi et surtout à détecter et corriger au plus tôt les défauts de forme et de fond de la spécification. Le but de la présentation est de montrer l'application de BASAALT à un cas public proposé par l'ONERA., la spécification du logiciel de contrôle d'un LGS (Landing gear System).

*Intervenant

Validations d'exigences au plus tôt

Jean-Michel Bruel * [1]

[1] IRIT – Centre National de la Recherche Scientifique - CNRS – France

Les exigences liées à un système d'intérêt peuvent porter sur le les objectifs de ses parties-prenantes, le système lui-même, son environnement, ou encore les attentes en termes de son cycle de vie (développement, exploitation, ...). Elles peuvent prendre de nombreuses formes (documents, modèles, ...), concerner de nombreuses disciplines différentes et à différents niveaux d'abstraction.
Pour les plus critiques d'entre elles, il peut être intéressant de représenter formellement les éléments clés qu'elles expriment, les relations qu'elles ont les unes avec les autres, etc. Cet ensemble d'éléments, reliées entre eux représentent une base de concepts permettant son analyse au plus tôt dans le cycle de vie, mais nécessite de posséder une ontologie des concepts que l'on cherche à identifier.
Le but de cette présentation est de montrer les pistes envisagées pour atteindre ce but sur un exemple concret, dans le cadre de la chaire industrielle Airbus CoCoVaD.

*Intervenant

GT GLIA

MLinter: Learning Coding Practices from Examples - Dream or Reality?

Corentin Latappy * [1]

[1] LaBRI – Université de Bordeaux (Bordeaux, France) – France

Coding practices are increasingly used by software companies. Their use promotes consistency, readability, and maintainability, which contribute to software quality. Coding practices were initially enforced by general-purpose linters, but companies now tend to design and adopt their own company-specific practices. However, these company-specific practices are often not automated, making it challenging to ensure they are shared and used by developers. Converting these practices into linter rules is a complex task that requires extensive static analysis and language engineering expertise. In this paper, we seek to answer the following question: can coding practices be learned automatically from examples manually tagged by developers? We conduct a feasibility study using CodeBERT, a state-of-the-art machine learning approach, to learn linter rules. Our results show that, although the resulting classifiers reach high precision and recall scores when evaluated on balanced synthetic datasets, their application on real-world, unbalanced codebases, while maintaining excellent recall, suffers from a severe drop in precision that hinders their usability.

*Intervenant

Performance prediction of configurable systems using machine learning

Paul Temple * [1]

[1] Irisa – Univ Rennes, Inria, CNRS, IRISA F-35000 Rennes – France

Modern software systems are being more and more configurable to adapt to a maximum of requirements coming from different users.
This is done by adding new functionalities to the software and letting users decide whether they need them; thus creating a configuration of the system tailored for their specific use.
Yet, being able to manage such a system and have a clear view of its capabilities is difficult as the number of possible configurations increase exponentially with the number of newly added functionalities.
A way to gain insights about these capabilities (e.g., execution time or memory consumption) is to use a Machine Learning (ML) model that can predict these capabilities without executing the configuration of the system.

*Intervenant

Benchmarks for ML4Code

Romain Robbes * [1]

[1] LaBri – LaBRI, Universite de Bordeaux, Bordeaux INP, CNRS, UMR 5800, Talence, France – France

We present two benchmarks and datasets that are designed to help the ML4Code community progress on goals that we think are important.

GLUE Code (Global and Local Understanding Evaluation of Code) is geared towards the development of models that use a global context beyond a code snippet. Indeed, one of our studies shows that 60% of method calls are project-specific, and 40% come from a distant context. GLUE Code is based on the JEMMA dataset of source code projects, a dataset of 50,000 projects (derived from 50K-C) that include significant post-processing to add source code representations, call graphs, and static analysis tool data. GLUE Code includes tasks that require a model to go beyond the current code snippet and include larger context (file, package, callers/callees). GLUE Code users can use JEMMA to assemble the context they need to solve the GLUE Code tasks. In this way, GLUE Code and JEMMA allow users to experiment with a variety of source code contexts.

Find more details on JEMMA at: https://arxiv.org/abs/2212.09132

RunBugRun is a large-scale, executable, and multi-lingual dataset to incentivize Automated Program Repair models to leverage runtime information in their design. RunBugRun is derived from CodeNet; it includes 450,000 (carefully curated) executable bug/fix pairs that can be validated via running tests. Generated patches can be compiled and executed. RunBugRun includes bug/fix pairs in C, C++, Python, Java, JavaScript, Go, Ruby, and PHP. The bug/fix pairs are also labeled with respect to the kind of changes they include. Initial results on two baselines show both that there is room for future improvement, and the potential of transfer learning from common to uncommon languages.

Find more details on RunBugRun at: https://github.com/giganticode/run_bug_run

Based on joint work of:
Anjan Karmakar, Julian Prenner, Miltiadis Allamanis

*Intervenant

GT CLAP

RT-DFI : Optimizing Data-Flow Integrity for Real-Time Systems

Nicolas Bellec *† [1], Guillaume Hiet‡ [2], Simon Rocicki§ [3], Frédéric Tronel¶ [2], Isabelle Puaut‖ [1]

[1] Pushing Architecture and Compilation for Application Performance (PACAP) – Inria Rennes – Bretagne Atlantique, ARCHITECTURE – Campus de Beaulieu 35042 Rennes cedex, France
[2] Centrale Supélec – SUPELEC – France
[3] Architectures matérielles spécialisées pour l'ère post loi-de-Moore (TARAN) – Inria Rennes – Bretagne Atlantique, ARCHITECTURE – Campus de beaulieu 35042 Rennes cedex, France

The emergence of Real-Time Systems with increased connections to their environment has led to a greater demand in security for these systems. Memory corruption attacks, which modify the memory to trigger unexpected executions, are a significant threat against applications written in low-level languages. Data-Flow Integrity (DFI) is a protection that verifies that only a trusted source has written any loaded data. The overhead of such a security mechanism remains a major issue that limits its adoption.
In this presentation, we present RT-DFI, a new approach that optimizes Data-Flow Integrity to reduce its overhead on the Worst-Case Execution Time. We model the number and order of the checks and use an Integer Linear Programming solver to optimize the protection on the Worst-Case Execution Path. Our approach protects the program against many memory-corruption attacks, including Return-Oriented Programming and Data-Only attacks. Moreover, our experimental results show that our optimization reduces the overhead by 7% on average compared to a state-of-the-art implementation.

*Intervenant
†Auteur correspondant: nicolas.bellec@inria.fr
‡Auteur correspondant: guillaume.hiet@centralesupelec.fr
§Auteur correspondant:
¶Auteur correspondant: frederic.tronel@inria.fr
‖Auteur correspondant: isabelle.puaut@irisa.fr

Polymorphic Types with Polynomial Sizes

Jean-Louis Colaço *† [1], Baptiste Pauget [2], Marc Pouzet [3]

[1] ANSYS – Ansys inc. – France
[2] Inria Paris-Rocquencourt – Institut National de Recherche en Informatique et en Automatique – INRIA Rocquencourt : Domaine de Voluceau, Rocquencourt B.P. 105 78153 le Chesnay Cedex, France
[3] Ecole normale supérieure – PSL (Paris) – L'Institut National de Recherche en Informatique et e n Automatique (INRIA) – France

We present a compile-time analysis for tracking the size of data-structures in a statically typed and strict functional language. This information is valuable for static checking and code generation. Rather than relying on dependent types, we propose a type-system close to that of ML: polymorphism is used to define functions that are generic in types and sizes; both can be inferred. This approach is convenient, in particular for a language used to program critical embedded systems, where sizes are indeed known at compile-time. By using sizes that are multivariate polynomials, we obtain a good compromise between the expressiveness of the size language and its properties (verification, inference).

*Intervenant
†Auteur correspondant:

Co-optimizing Dataflow Graphs and Actors with MLIR

Pedro Ciambra *† [1,2], Mickaël Dardaillon [1], Maxime Pelcat [1], Yviquel Hervé [2]

[1] Institut d'Électronique et des Technologies du numéRique (IETR) – Université de Rennes, Institut National des Sciences Appliquées - Rennes, CentraleSupélec, Centre National de la Recherche Scientifique, Nantes Université - pôle Sciences et technologie – Campus de Beaulieu Bâtiment 11D 263 Av.Général Leclerc-CS 74205 35042 Rennes Cedex, France

[2] Computer Systems Laboratory [Campinas] (LSC - UNICAMP) – LSC - Computer Systems Laboratory Av. Albert Einstein, 1251 - Cidade Universitária Campinas - SP - Brazil 13083-852, Brésil

Dataflow programming is considered a good solution for the implementation of parallel signal processing applications. However, the strict separation between kernel and coordination codes limits the variety of possible optimizations and the compatibility with state-of-the-art compiler frameworks. We present a prototype static dataflow compiler, built with the LLVM MLIR framework, that overcomes these limitations and enables a previously impossible combination of optimization strategies that leverages information from the dataflow topology. Initial results show 30% wall time improvement and 53% memory usage improvement on a video processing workload.

*Intervenant

†Auteur correspondant: Pedro.Ferrazoli-Ciambra@insa-rennes.fr

GT Debugging

Détection des anomalies d'ordonnancement dans un système temps réel

Blandine Djika * [1]

[1] Laboratoire des sciences et techniques de l'information, de la communication et de la connaissance (Lab-STICC) – Ecole Nationale d'Ingénieurs de Brest, Université de Bretagne Sud, Université de Brest, École Nationale Supérieure de Techniques Avancées Bretagne, Institut Mines-Télécom [Paris], Centre National de la Recherche Scientifique, Université Bretagne Loire, IMT Atlantique – Technopole Brest Iroise CS 83818 29238 BREST cedex 3, France

Nos travaux portent sur les anomalies d'ordonnancement dans les systèmes temps réel. Dans un système temps réel, les tâches doivent être exécutées de sorte qu'elles respectent des contraintes temporelles telles que des échéances. Pour ce faire, les acteurs du domaine valident le comportement temporel des tâches lors des phases amonts de la conception du système. Toutefois, sous certaines conditions, il peut arriver que les échéances des tâches ne soient finalement pas respectées à l'exécution. C'est notamment le cas lors d'évènements contre intuitifs comme l'augmentation des ressources du système. On parle alors d'anomalies d'ordonnancement. Dans cet exposé, nous décrivons un modèle d'analyse permettant la détection de ces anomalies d'ordonnancement. Nous montrons également comment exploiter ce modèle pour le développement d'un outil de monitoring sur POSIX/RTEMS appelé MONANO. MONANO permet de détecter ce type d'anomalies à l'exécution.

*Intervenant

Finding Faults of Executable Models: Manually and Automatically

Faezeh Khorram * [1]

[1] Huawei Technologies – Huawei Technologies, Huawei Technologies – France

When a model represents the dynamic aspects of a system (a.k.a behavioral model), testing it becomes a necessitate to ensure it represents the correct behavior. If test cases fail, it alerts the existence of faults in the model, hereafter proper means are needed to localize the faults. In this presentation, I will talk about both manual and automatic fault localization techniques in the context of executable models and their test cases. In particular, the challenges of adopting the existing debugging techniques from the software testing area to the model testing area will be discussed. Finally, I will present a generic solution to tackle the challenges along with demonstrating the developed solution in an Eclipse environment.

*Intervenant

Prototypage IHM pour la défense : déboguage et correctifs distribués à chaud et sans interruption de système collaboratifs en cours d'exécution

Pierre Laborde * [1]

[1] Thales DMS – THALES – France

Nous réalisons des prototypes d'interface homme-machine (IHM) pour différents domaines de la défense (Aérien, Terrestre, Maritime, etc.) dans le but de concevoir de futurs systèmes de mission. Pour mettre au point ces prototypes nous allons jusqu'à mettre en situation les utilisateurs avec des scénarios d'usages au travers d'évaluations ergonomiques. Les utilisateurs sont alors immergés dans une séance qui leur permet de réaliser les tâches de leur quotidien comme s'ils avaient le futur système entre leurs mains. Nos spécialistes UX (User Experience) et IHM observent alors les séances au travers des différents scénarios pour mieux récolter les retours utilisateurs, les problèmes, identifier les manques mais aussi mettre en avant et tester des nouvelles capacités ou ergonomies. Cependant, ces prototypes ne sont pas exempts de défauts : il y a parfois des bogues et des choses que nous n'avons pas anticipées. Ces séances sont longues, rares et difficiles à mettre au point, nous avons donc besoin de faire travailler au maximum les utilisateurs sans les interrompre et sans perturber le bon déroulé des scénarios. Il nous faut pouvoir résoudre tous les problèmes qui surviennent (dans la mesure du possible) pendant que le système est utilisé. Si le système doit être arrêté pour un problème, nous perdons parfois des heures de mise en situation et de contexte utilisateur. Nous avons mis au point des outils et des processus qui nous permettent d'intervenir directement pendant l'exécution, sans interruption majeure, et ceci alors que le système est utilisé de manière collaborative par plusieurs personnes en même temps. Nous expliquons dans cette présentation notre démarche au travers d'un exemple récent d'évaluation ergonomique sur prototype de système de surveillance maritime à bord d'un avion simulé avec un équipage de 5 personnes.

*Intervenant

Protocol-Based Interactive Debugging for Domain-Specific Languages

Josselin Enet * [1]

[1] NaoMod – LS2N, UMR CNRS 6004, – France

Interactive debuggers are established tools used by developers to understand programs and localize faults. They are equally valuable in the context of model-driven development, when working on executable behavioral models. However, development costs of interactive debuggers for Domain-Specific Languages (DSLs) can be significant. In order to mitigate these costs, several reusable DSL-agnostic debugging solutions have been proposed. We argue that the applicability of these solutions is limited by being tied to a fixed set of debugging services, a specific language engineering approach, or a particular user interface. In this paper, we present a novel approach to provide interactive debugging services for executable DSLs through a reusable generic architecture. We propose a protocol allowing a generic interactive debugger to communicate with heterogeneous DSL runtimes, both for controlling the execution and for configuring the debugger with domain-specific breakpoints. The proposed debugger can itself be controlled using a reinterpretation of the Debug Adapter Protocol (DAP), for an effortless integration in existing Integrated Development Environments (IDEs) that support it. Using a prototype implementation based on JSON-RPC and two heterogeneous DSL runtimes, we demonstrate that our approach provides an off-the-shelf reusable interactive debugger that supports meaningful domain-specific breakpoints, and that can be used with minimal effort within a standard IDE such as Visual Studio Code.

*Intervenant

Comment faciliter le processus de debugging en tracant la compilation

Bruno Mateu * [1]

[1] IMT Atlantique – IMT Atlantique, IMT Atlantique, IMT Atlantique – France

Un processus de compilation moderne comporte plusieurs centaines de passes qui peuvent chacune contenir plusieurs transformations. Ces transformations sont appliquées sur le code de manière sélective, selon les options de configurations données et des propriétés du code en entrée. À un code source et une configuration donnée (contenant éventuellement une graine pour le générateur de nombres aléatoires), le processus de compilation exécuté sera toujours le même. Cependant, une modification - même légère - de la configuration ou du code peut modifier la liste des transformations appliquées sur le code. Les propriétés du code produit dépendent essentiellement des passes appliquées.
Pour cette raison retracer l'histoire d'une instruction telle qu'elle apparaît dans le binaire, c'est-à-dire à quelle(s) ligne(s) de code source elle correspond et quelles sont les transformations subies par celle(s)-ci, est difficile. S'il est possible de connaître la liste des passes appelées lors d'un processus de compilation, on ne peut pas savoir quelles sont les transformations qu'une passe a réalisé sans comparer la représentation interne en entrée et en sortie.

Pour débugger un code, une première étape peut être de désactiver toute optimisation ou obfuscation, afin de travailler sur un code machine le plus proche possible du code source. Cependant, dans certains cas, le bug détecté cessera d'appartaitre une fois ces options désactivées. Dans ce cas, il peut s'agir d'un bug à l'intérieur d'une passe d'optimisation ou d'obfuscation, et il relève donc des développeurs du compilateur de le résoudre. Mais il peut également s'agir d'un bug dans le code source qui ne se manifeste que lorsqu'il interagit avec certainse optimisations et/ou obfuscations.

Dans les deux cas, pour identifier le bug, accéder à l'histoire des instructions pour mieux comprendre comment chaque instruction du code machine a été formée à partir du code source serait un atout, à la fois pour l'utilisateur du compilateur pour l'aider à identifier le bug, et pour le développeur du compilateur, pour l'aider dans le processus de debugging du compilateur lui-même, en particulier lorsque le code produit par le compilateur n'est pas correct.
Dans cette présentation, je souhaite présenter le problème détaillé ci-dessus, présenter les travaux que j'ai effectué sur le compilateur LLVM pour tracer les transformations, puis présenter deux cas d'usages-type de bugs pour lesquels la trace produite peut faciliter le processus de debug.

*Intervenant

GT VL

HyperAST: Enabling Efficient Analysis of Software Histories at Scale

Quentin Le Dilavrec * [1]

[1] Institut de Recherche en Informatique et Systèmes Aléatoires (IRISA) – Université de Rennes, Institut National des Sciences Appliquées - Rennes, Université de Bretagne Sud, École normale supérieure - Rennes, Institut National de Recherche en Informatique et en Automatique, CentraleSupélec, Centre National de la Recherche Scientifique, IMT Atlantique – Avenue du général LeclercCampus de Beaulieu 35042 RENNES CEDEX, France

Abstract Syntax Trees (ASTs) are widely used beyond compilers in many tools that measure and improve code quality, such as code analysis, bug detection, mining code metrics, refactoring. With the advent of fast software evolution and multistage releases, the temporal analysis of an AST history is becoming useful to understand and maintain code.

However, jointly analyzing thousands versions of ASTs independently faces scalability issues, mostly combinatorial, both in terms of memory and CPU usage. In this paper, we propose a novel type of AST, called HyperAST, that enables efficient temporal code analysis on a given software history by: 1/ leveraging code redundancy through space (between code elements) and time (between versions); 2/ reusing intermediate computation results. We show how the HyperAST can be built incrementally on a set of commits to capture all multiple ASTs at once in an optimized way. We evaluated the HyperAST on a curated list of large software projects. Compared to Spoon, a state-of-the-art technique, we observed that the HyperAST outperforms it with an order-of-magnitude difference from × 6 up to × 8076 in CPU construction time and from × 12 up to × 1159 in memory footprint. While the HyperAST requires up to 2h 22min and 7.2GB for the biggest project, Spoon requires up to 93h and 31min and 2.2TB. The gains in construction time varied from to and the gains in memory footprint varied from to . We further compared the task of finding references of declarations with the HyperAST and Spoon. We observed on average precision and recall without a significant difference in search time.

*Intervenant

Guiding Feature Models Synthesis from User-Stories: An Exploratory Approach

Thomas Georges * [1]

[1] Laboratoire d'Informatique de Robotique et de Microélectronique de Montpellier (LIRMM) – Centre National de la Recherche Scientifique, Université de Montpellier – 161 rue Ada - 34095 Montpellier, France

Throughout the software lifecycle, a huge amount of knowledge is accumulated around the source code. In our work, we focus on agile software requirements, more specifically on user stories, and on issues and merge requests of the version control platforms, opened for implementing user stories. In this paper, we present a method that leverages the use of this knowledge to guide an SPL migration. In addition to user stories and the source code itself, we exploit domain ontologies to enrich and better organize this knowledge. We consider merge requests in version control systems as the hub between user stories (requirements) and the source code (implementation). In this work, we aim to synthesize feature models by combining several approaches. Natural language processing and clustering of user stories are used to identify features (NLP step). Formal concept analysis is used to hierarchically classify them (FCA step). Logical rules generated by analyzing the results of NLP and FCA steps are used to refine feature constraints. We implemented and evaluated this method on a dataset from our industrial partner. The obtained results showed the efficiency of our method in properly synthesizing feature models towards an SPL migration of our partner's code base.

*Intervenant

Une théorie des organisations communautaires de maintenance de paquets

Théo Zimmerman * [1]

[1] LTCI, Télécom Paris, Institut Polytechnique de Paris – LTCI, Télécom Paris, Institut Polytechnique de Paris – 19 Place Marguerite Perey 91120 Palaiseau, France

Dans de nombreux écosystèmes de langages de programmation, les développeurs dépendent de plus en plus de dépendances externes en open source, disponibles via des gestionnaires de paquets. Les paquets clés qui ne sont pas maintenus présentent un risque pour les projets qui en dépendent ainsi que pour les écosystèmes. Par conséquent, des initiatives communautaires peuvent émerger au sein des écosystèmes pour résoudre ce problème en adoptant les paquets clés ayant des problèmes de maintenance. Dans mon exposé, je présenterai les résultats de l'article coécrit avec Jean-Rémy Falleri et récemment accepté pour publication dans Empirical Software Engineering intitulé "A Grounded Theory of Community Package Maintenance Organizations". Le but de celui-ci était de construire une théorie de ces organisations (CPMO), notamment leur émergence et leur mode de fonctionnement. Pour ce faire, nous avons utilisé une méthodologie qualitative appelée Grounded Theory. Nous avons analysé des documents existants provenant de plusieurs CPMO, tels que des documentations et des discussions sur des forums publics, complétés par des entretiens avec des initiateurs de CPMO. Je parlerai également de mon application de ce modèle d'organisation à l'écosystème Coq, avec la création de l'organisation Coq-community, et de l'expérience acquise par ce biais.

*Intervenant

On the Benefits and Limits of Incremental Build of Software Configurations: An Exploratory Study

Georges Aaron Randrianaina * [1]

[1] Institut de Recherche en Informatique et Systèmes Aléatoires (IRISA) – Université de Rennes, Institut National des Sciences Appliquées - Rennes, Université de Bretagne Sud, École normale supérieure - Rennes, Institut National de Recherche en Informatique et en Automatique, CentraleSupélec, Centre National de la Recherche Scientifique, IMT Atlantique – Avenue du général LeclercCampus de Beaulieu 35042 RENNES CEDEX, France

Software projects use build systems to automate the compilation, testing, and continuous deployment of their software products. As software becomes increasingly configurable, the build of multiple configurations is a pressing need, but expensive and challenging to implement. The current state of practice is to build independently (*a.k.a.*, clean build) a software for a subset of configurations. While incremental build has been studied for software evolution and relatively small changes of the source code, it has surprisingly not been considered for software configurations. In this exploratory study, we examine the benefits and limits of building software configurations incrementally, rather than always building them cleanly. By using five real-life configurable systems as subjects, we explore whether incremental build works, outperforms a sequence of clean builds, is correct *w.r.t.* clean build, and can be used to find an optimal ordering for building configurations. Our results show that incremental build is feasible in 100% of the times in four subjects and in 78% of the times in one subject. In average, 88.5% of the configurations could be built faster with incremental build while also finding several alternatives faster incremental builds. However, only 60% of faster incremental builds are correct. Still, when considering those correct incremental builds with clean builds, we could always find an optimal order that is faster than just a collection of clean builds with a gain up to 11.76%.

*Intervenant

GT GLSec

Collaborative Security-by-Design Platform with Model-Driven Engineering approach

Othmane El Karm * [1]

[1] IRIT – Université Toulouse 3 - ERL5294 CNRS – France

Le développement de logiciels expose à des risques de sécurité croissants. Pour faire face à ces défis, il est important d'adopter une approche proactive et assurer une collaboration entre les développeurs, les experts en sécurité et les utilisateurs est essentielle pour renforcer la sécurité des systèmes, cela comprend la modélisation des menaces (threat modeling) pour identifier les vulnérabilités potentielles et les scénarios d'attaque, ainsi que l'utilisation de l'ingénierie dirigée par les modèles (MDE) pour améliorer la sécurité des logiciels.

*Intervenant

Améliorer la confiance dans la chaine d'approvisionnement du logiciel avec les gestionnaires de paquets fonctionnels et la compilation reproductible

Julien Malka * [1]

[1] Telecom Paris – Télécom ParisTech – France

Dans cet exposé, je présenterai les concepts de gestionnaires de paquets fonctionnels et de compilation reproductible et montrerai le lien avec la sécurité de la chaîne d'approvisionnement du logiciel open source. Je ferai un tour d'horizon des questions de recherches qui seront abordées dans mon projet de thèse, avec un attachement particulier pour les travaux déjà débutés dans le cadre d'un stage.

*Intervenant

Feature-based software architecture analysis to identify safety and security interactions

Oum El Kheir Aktouf * [1]

[1] LCIS – Institut National Polytechnique de Grenoble - INPG – France

In the automotive domain, feature-based software architecture is a widely used software design method to produce cost efficient and reusable software. With increasing complexity of automotive systems and the shift towards automated driving, safety and security measures become even more crucial for these systems. However, safety and security functionalities can undermine each other if they interact in unintended ways. We propose the novel method FIISS for automatic identification of interactions between safety and security features in UML models. We evaluate our implementation of the method by applying it to a real-world component for autonomous driving. We show that the method is able to identify unintended interactions while providing only few false positive findings. Thus, we see that our method can be applied to real-world UML system designs without modifying the underlying models and without applying specialized UML profiles.

*Intervenant

GT HiFi

Stimulus : un langage de programmation synchrone à contrainte appliqué à la simulation d'exigences temps-réel fonctionnelles

Bertrand Jeannet *† [1]

[1] 3DS – Dassault Systèmes – France

Les langages synchrones tels que Lustre et Lucid Synchrone ont été conçus pour programmer des systèmes temps-réels de contrôle-commande de manière sûre et efficace. Pour exprimer des propriétés sur ces programmes a ensuite été introduite la notion d'observateur synchrone. Enfin, pour simuler et tester ces programmes, le besoin a été identifié de modéliser des environnements non-déterministes pour stimuler le programme sous test avec des entrées réalistes, ce qui a mené notamment au langage à contraintes Lutin et à la machine d'exécution Lurette.
Stimulus synthétise ces travaux : il combine les contraintes de Lutin et les automates hiérarchiques de Lucid Synchrone, et il introduit une notion d'observateur permettant de transformer tout fragment de programme générant des signaux en reconnaisseur de propriété sur ces signaux. Dans ce cadre, exécuter un programme consiste, à chaque pas d'exécution synchrone, à faire évoluer l'état des automates et à résoudre les contraintes actives sur les signaux à l'aide d'un solveur.
L'application privilégiée qui a guidé nos travaux est la simulation des exigences fonctionnelles d'un système temps-réel, que nous illustrerons.

*Intervenant
†Auteur correspondant: Bertrand.JEANNET@3ds.com

10 ans de "Precision Tuning"

Matthieu Martel *† [1]

[1] Université Perpignan Via Domitia (UPVD) – Lamps – France

Le réglage de la précision (precision tuning) consiste à trouver, pour les variables flottantes d'un programme, les formats minimaux permettant de garantir une précision donnée pour les résultats. Depuis dix ans, ce sujet à reçu une grande attention et de nombreux outils ont été développés. Dans cet exposé, nous présenterons les principaux résultats de cette décennie en insistant particulièrement sur l'outil POP qui propose une approche originale fondée sur la résolution de contraintes pour résoudre ce problème.

*Intervenant

†Auteur correspondant: matthieu.martel@univ-perp.fr

GT LVP AFADL

Une logique de séparation de haut niveau pour l'espace de tas en présence d'un glaneur de cellule

Alexandre Moine *† [1], Arthur Charguéraud [2], François Pottier [1]

[1] Inria – Cambium – France
[2] Inria, Université de Strasbourg, CNRS – ICube – France

Ce texte est un résumé de notre article "A High-Level Separation Logic for Heap Space under Garbage Collection" présenté à POPL en 2023.

*Intervenant
†Auteur correspondant: alexandre.moine@inria.fr

Nondeterministic, Recursive, and Impure Programs in Coq

Ludovic Henrio * [1], Nicolas Chappe [1], Yannick Zakowski [1], Paul He [2], Steve Zdancewic [2]

[1] Univ Lyon, EnsL, UCBL, CNRS, Inria, LIP – F-69342, LYON Cedex 07 – France
[2] University of Pennsylvania – États-Unis

This extended abstract summarizes the article presented at POPL this year: "Choice Trees: Representing Nondeterministic, Recursive, and Impure Programs in Coq" by N Chappe, P He, L Henrio, Y Zakowski, S Zdancewic - Published in Proceedings of the ACM on Programming Languages, 2023.

It introduces a new formalism designed specifically to facilitate the definition of and reasoning about nondeterministic computations in Coq's dependent type theory. The key idea is to update Xia, et al.'s interaction trees framework with native support for nondeterminisic "choice nodes" that represent internal choices made during computation.

*Intervenant

Génération automatique de tests d'égalité corrects en Coq, en pratique

Benjamin Grégoire [1], Jean-Christophe Léchenet *† [1], Enrico Tassi [1]

[1] Université Côte d'Azur – INRIA – France

Cet article est un résumé étendu de l'article "Practical and Sound Equality Tests, Automatically : Deriving eqType Instances for Jasmin's Data Types with Coq-Elpi", accepté à CPP 2023 (12th ACM SIGPLAN International Conference on Certified Programs and Proofs).

*Intervenant

†Auteur correspondant: jean-christophe.lechenet@inria.fr

GT MTV2 AFADL

On race detection in distributed systems using state models

Evgenii Vinarskii *† [1,2]

[1] Télécom SudParis – SAMOVAR – France
[2] Institut Polytechnique de Paris – Institut Polytechnique de Paris – France

This paper summarises the results of two papers devoted to race detection in distributed systems focusing on two complementary strategies: model checking and model-based test generation. We discuss how these approaches have been applied to the composition of an SDN controller and a switch resulting in the detection of races in the SDN framework. This study aims to provide the valuable insights for researchers and practitioners interested in race detection in distributed systems.

*Intervenant
†Auteur correspondant: vinarskii.evgenii@telecom-sudparis.eu

Un support efficace des critères de couverture de test avancés pour Klee

Nicolas Berthier [1], Steven De Oliveira [1], Nikolai Kosmatov * [2], Delphine Longuet [3], Romain Soulat [4]

[1] OCamlPro – OCamlPro – France
[2] Thales Research Technology – Thales Research Technology – France
[3] Thales Research Technology – Thales Research Technology – France
[4] Thales Research Technology – Thales Research – France

Les techniques de génération automatique de tests ont fait des progrès significatifs pendant les vingt dernières années. Un des succès les plus remarquables dans ce domaine est l'exécution symbolique dynamique (DSE), une technique de génération de tests qui combine l'exécution symbolique et l'exécution concrète du programme sous test. Des travaux récents ont proposé un mécanisme générique pour la spécification des critères de couverture de test à l'aide d'objectifs de test élémentaires, appelés étiquettes de couverture (ou (coverage) labels), ainsi que des solutions pour une génération de tests efficace pour les labels. Cependant, ces techniques n'ont jamais été intégrées dans des générateurs de tests publiquement disponibles. Notre but est de démontrer qu'un support efficace des labels peut être intégré dans Klee, un générateur de tests populaire et ouvert, basé sur la DSE. La version de l'outil réalisée, appelée Klee4labels, est publiquement disponible. Nos expérimentations confirment les avantages de la technique proposée. Cette soumission est un résumé long de l'article publié à SAC-SVT 2023.

*Intervenant

Energy Büchi Problems

Sven Dziadek [1], Uli Fahrenberg * [1], Philipp Schlehuber-Caissier [1]

[1] LRE, EPITA – LRE, EPITA – France

We show how to efficiently solve energy Büchi problems in finite weighted automata and in one-clock weighted timed automata. Solving the former problem is our main contribution and is handled by a modified version of Bellman-Ford interleaved with Couvreur's algorithm. The latter problem is handled via a reduction to the former relying on the corner-point abstraction. All our algorithms are freely available and implemented in a tool based on the open source frameworks TChecker and Spot.

*Intervenant

Pairwise Testing Revisited for Structured Data with Constraints

Hélène Waeselynck *† [1]

[1] LAAS-CNRS – Laboratoire d'Analyse et d'Architecture des systèmes – Toulouse, France

Pairwise testing (PT) exercises the interactions of pairs of input parameters. The approach is classically defined for a flat set of parameters, the number of which is fixed. Such a definition does not fit well with applications that process structured data like XML and JSON documents. This paper revisits the PT concepts to accommodate hierarchical data structures. The choices and pairs are created by considering the multiplicity of data instances, their access paths and common ancestors. The revised PT approach is implemented on top of on a recent data generation tool, TAF. TAF mixes random sampling and constraint solving to produce diverse data from XML-based models. Our PT implementation interacts with TAF by inserting pair coverage constraints into the models. It monitors overall coverage progress by XPath queries on the data returned by TAF. The approach is demonstrated for two data models: a 3D scene for an agricultural robot, and a population of taxpayers for a tax management system.

*Intervenant
†Auteur correspondant:

AFADL

Vérification de propriétés interactives sur des systèmes réactifs interactifs

Cécile Marcon * [1], Xavier Thirioux [2], Celia Picard [3], Cyril Allignol [3]

[1] Institut Supérieur de lÁéronautique et de lÉspace – Institut Supérieur de l'Aéronautique et de l'Espace (ISAE), Institut supérieur de l'aéronautique et de l'espace [ISAE] – France
[2] ISAE-SUPAERO – Institut Supérieur de l'Aéronautique et de l'Espace – France
[3] enac – Ecole Nationale de l'Aviation Civile - ENAC – France

Les systèmes interactifs sont des systèmes informatiques qui échangent avec un utilisateur humain. Ils peuvent être décrits grâce à un UIDL (User Interface Description Language). Nous cherchons à apporter des garanties sur de tels systèmes. À cette fin, nous voulons exprimer et vérifier formellement des propriétés qui portent sur des aspects interactifs de programmes décrits à l'aide d'un UIDL. Nous disposons pour cela d'un UIDL formel, BIGUIL, dont la sémantique est décrite à l'aide des bigraphes. Nous voulons explorer l'outillage de BIGUIL pour garantir ces propriétés par construction.

Mots-Clés: interactive properties, interactive systems, formal expression

*Intervenant

Une approche pour inférer les expressions de calcul géométrique en modélisation à base topologique

Romain Pascual *† [1,2], Pascale Le Gall [3,4], Hakim Belhaouari [5,6], Agnès Arnould [5,6]

[1] MICS – MICS, CentraleSupélec, Université Paris-Saclay – France
[2] CentraleSupélec, Université Paris-Saclay – CentraleSupelec, Saclay, France. – France
[3] MICS – MICS – France
[4] CentraleSupelec – CentraleSupelec, Saclay, France. – France
[5] XLIM CNRS UMR 7252 – XLIM CNRS UMR 7252 – France
[6] université de Poitiers – Université de Poitiers, Université de Poitiers – France

La conception d'opérations de modélisation géométrique repose sur leur implantation dans un langage de programmation. Même si cette tâche peut être simplifiée en exploitant une description de haut niveau de ces opérations, la difficulté de les implanter contraste avec l'apparente simplicité de la description d'une opération sur un exemple. Nous proposons une méthode d'inférence d'opérations à partir d'un exemple représentatif. Plus précisément, nous nous plaçons dans le domaine de la modélisation géométrique à base topologique qui propose une représentation d'objets nD par décomposition en cellules topologiques (sommets, arêtes, faces, volumes, etc.) sur lesquelles sont ajoutées des informations géométriques. Ce domaine admet une spécification de la topologie par des structures combinatoires qui peuvent être représentées à l'aide de graphes, de sorte qu'une opération se formalise comme une règle de transformation de graphes. Dans cet article, nous complétons notre algorithme d'inférence du calcul topologique avec une approche pour la déduction des expressions de calcul géométrique. Notre approche exploite deux idées principales : des abstractions topologiques des expressions géométriques pour assurer la généricité des calculs retrouvés et une représentation comme un problème de satisfaction de contraintes de l'expression recherchée.

*Intervenant
†Auteur correspondant: romain.pascual@centralesupelec.fr

Cybersécurité pour les systèmes embarqués critiques à base d'Intelligence Artificielle

Céline Bellanger * [1]

[1] enac – Ecole Nationale de l'Aviation Civile - ENAC – France

L'intelligence artificielle est de plus en plus utilisée dans les systèmes embarqués critiques, et particulièrement dans l'aéronautique. Elle peut remplacer des fonctions existantes comme la stabilisation ou le guidage ; ou ouvrir de nouvelles possibilités comme la réalisation de toutes les phases de vol de façon autonome notamment grâce à la reconnaissance d'images. Cependant, elle peut être la cible de nouveaux types de cyberattaques, spécifiques à l'intelligence artificielle. Dans ce document, nous présentons nos travaux en cours et à venir pour améliorer la sûreté des réseaux de neurones utilisés dans les systèmes embarqués. Nous nous appuierons sur des méthodes formelles pour étudier les propriétés temporelles des signaux issus des réseaux de neurones. L'approche visera à 1. définir les méthodes et outils pour évaluer les propriétés exprimées en Signal Temporal Logic (STL), 2. caractériser les propriétés d'intérêts dans cette logique STL et 3. étudier la validité de contrôleurs à base de réseaux de neurones vis-à-vis de ces propriétés.

*Intervenant

Approche Formelle Dirigée par les Modèles pour la Collaboration de DSLs

Salim Chehida * [1,2], Akram Idani [1,3], Mario Cortes-Cornax [1,4], German Vega [1,5]

[1] LIG – LIG – France
[2] CNRS – CNRS, CNRS : UMR8568, CNRS, CNRS : UMR6074, CNRS, CNRS : UMR5593, CNRS : ERL3189, CNRS : UMR7104, CNRS : UMR5244, CNRS : UMR2205 – France
[3] Grenoble-INP – Institut polytechnique de Grenoble (Grenoble INP) – France
[4] UGA – , Universitey of Grenoble Alpes (UGA) – France
[5] CNRS – Centre national de la recherche scientifique - CNRS (France) – France

Ce travail présente une extension de la plate-forme {} permettant de faire collaborer des modèles issus de différents langages dédiés domaines (ou DSLs). {} est un atelier de conception formelle de DSLs qui repose sur la méthode B. Il permet de spécifier en B la syntaxe abstraite du langage ainsi que sa sémantique d'exécution. Cependant, l'outil se limite à un seul langage à la fois. Aussi, son applicabilité à des contextes réalistes est-elle restreinte car un système informatique est souvent conçu au travers d'une panoplie de modèles. Dans ce travail nous proposons de l'étendre en vue de couvrir la composition et la coordination de plusieurs DSLs. Ces aspects sont d'abord capturés via un modèle BPMN (Business Process Model and Notation) et ensuite traduits dans CSP (Communication Sequential Process) afin de définir un cadre rigoureux favorisant l'animation et la vérification. Notre approche a été appliquée avec succès sur une étude de cas réelle fournie par RTE, le gestionnaire du réseau de transport d'électricité français.

*Intervenant

Débogage Multivers de Modèles UML

Matthias Pasquier * [1], Ciprian Teodorov [2,3], Frédéric Jouault [4], Matthias Brun [4], Loïc Lagadec [2,3]

[1] Ertosgener – Ertosgener – France
[2] Lab-STICC CNRS UMR 6285 – Lab-STICC CNRS UMR 6285 – France
[3] ensta bretagne – ENSTA Bretagne, ENSTA Bretagne, Ensta-Bretagne – France
[4] ESEO – ESEO – France

Pour faciliter la phase de spécification, nous présentons un débogueur multivers pour les machines à état UML ainsi que deux extensions à cette approche : Les breakpoints temporels permettent l'utilisation de multiples logiques temporelles pour explorer le modèle. La réduction permet un certain contrôle sur l'espace d'état exploré pour permettre le passage à l'échelle.

*Intervenant

Vérification de modèles relationnels et temporels avec Pardinus

Nuno Macedo [1], Julien Brunel [2,3], David Chemouil * [2,3], Alcino Cunha *

4

[1] INESC TEC, Faculty of Engineering of the University of Porto – Portugal
[2] ONERA DTIS – ONERA DTIS – France
[3] Université de Toulouse – Université de Toulouse, Université de Toulouse, Université deToulouse – France
[4] INESC TEC, University of Minho – Portugal

Nous résumons ici un article paru dans le Journal of Automated Reasoning. Celui-ci présente Pardinus, une extension du model finder relationnel Kodkod au moyen de la logique temporelle linéaire (avec opérateurs du passé). Pardinus inclut un moteur de bounded model-checking basé sur SAT ainsi qu'un moteur de model-checking commplet basé sur SMV, les deux permettant d'itérer sur les instances (ou contre-exemples) d'une spécification. Il offre aussi une stratégie d'analyse " décomposée " parallèle qui améliore l'efficacité des deux moteurs d'analyse.

*Intervenant

Décider la contextualité de configurations quantiques avec un solveur SAT

Axel Muller * [1]

[1] Université de Franche-Comté, CNRS, institut FEMTO-ST – F-25000 Besançon – France

Cet article présente mes travaux de recherche en première année de thèse, à l'institut FEMTO-ST à Besançon, sous la direction de M. Alain Giorgetti et le co-encadrement de M. Frédéric Holweck.

*Intervenant

Posters et DEMO

Animation of formal specifications of information systems with RoZ and JazaGUIv3 (demo)

Yves Ledru *† [1], German Vega * ‡ [1]

[1] Validation de Systèmes, Composants et Objets logiciels (VASCO) – Laboratoire d'Informatique de Grenoble – Laboratoire LIG - Bâtiment IMAG - 700 avenue Centrale, CS 40700 - 38058 Grenoble cedex 9, France

RoZ is a tool that translates a class diagram, enhanced with Z annotations, into a Z specification. This demo will present how this Z specification can be animated with the latest version of JazaGUI. JazaGUI is based on the jaza animator enhanced with a graphical user interface. This interface displays animation scenarios as a tree of method calls. It generates object diagrams from the current state of the animation, and uses the "why" feature of jaza to help explain animations which led to a failure. The tool is used to teach formal methods to Master's students.

Working group : HiFi (and IDM)

Mots-Clés: animation of specifications, integrated formal methods, validation, HiFi

*Intervenant

†Auteur correspondant: Yves.Ledru@imag.fr

‡Auteur correspondant: german.vega@univ-grenoble-alpes.fr

Revealing contextuality of quantum configurations with a SAT solver

Axel Muller *† [1], Metod Saniga [2], Alain Giorgetti [1], Henri De Boutray [3], Frédéric Holweck [4,5]

[1] Université de Franche-Comté, CNRS, institut FEMTO-ST – F-25000 Besançon – France
[2] Astronomical Institute of the Slovak Academy of Sciences – 05960 Tatranska Lomnica, Slovaquie
[3] ColibrITD – Paris – France
[4] ICB, UMR 6303, CNRS, University of Technology of Belfort-Montbéliard, UTBM – 90010 Belfort – France
[5] Department of Mathematics and Statistics, Auburn University – Auburn, AL, États-Unis

We present a use of a SAT solver to decide the quantum contextuality and evaluate the contextuality degree (a way to quantify contextuality) for a variety of point-line geometries located in binary symplectic polar spaces of small rank. With this code we were not only able to recover, in a more efficient way, all the results of a recent paper by de Boutray et al (J. Phys. A: Math. Theor. 55 475301, 2022), but also arrived at a bunch of new noteworthy results. This poster describes the approach, and presents the results for a number of subspaces of symplectic polar spaces whose rank ranges from two to seven, as well as the proofs that were found with the help of these results. Working group: LVP

Mots-Clés: Quantum geometry, Multi qubit observables, Quantum contextuality, Contextuality degree

*Intervenant
†Auteur correspondant: axel.muller@femto-st.fr

HyperAST: Analyser efficacement de grands historiques de code

Quentin Le Dilavrec *† [1], Djamel Eddine Khelladi * ‡ [1], Arnaud Blouin * § [1], Jean-Marc Jézéquel * ¶ [1]

[1] Institut de Recherche en Informatique et Systèmes Aléatoires (IRISA) – Université de Rennes, Institut National des Sciences Appliquées - Rennes, Université de Bretagne Sud, École normale supérieure - Rennes, Institut National de Recherche en Informatique et en Automatique, CentraleSupélec, Centre National de la Recherche Scientifique, IMT Atlantique – Avenue du général LeclercCampus de Beaulieu 35042 RENNES CEDEX, France

Cette démonstration présente l'HyperAST (1), une approche d'analyse d'historiques de code performante, se fondant sur la redondance du code à travers le temps et l'espace, ainsi que sur les possibilités d'analyses partielles de code. Actuellement, l'analyse des historiques de code se fait en mode "batch": chaque version (ou commit) est traitée (en grande partie) indépendamment les unes des autres pour calculer un ensemble de métriques ; à la fin de l'analyse ces métriques sont utilisées pour observer l'évolution de la base de code au cours du temps. Notre approche propose de traiter plus finement l'historique de code, au niveau de l'AST, et de partager les éléments identiques dans et entre chaque version. Cette démonstration vise à expliquer ce principe au travers de trois scénarios. Groupes de travail : VL et CLAP

Mots-Clés: Mining Software Repositories, Temporal Code Analysis, Code Quality

*Intervenant
†Auteur correspondant: quentin.le-dilavrec@irisa.fr
‡Auteur correspondant: djamel-eddine.khelladi@irisa.fr
§Auteur correspondant: arnaud.blouin@irisa.fr
¶Auteur correspondant: jezequel@irisa.fr

Analyse statique incrémentale pour la vérification de programmes par interprétation abstraite

Mamy Razafintsialonina *† [1]

[1] Université Paris-Saclay, CEA, List, Palaiseau – Sorbonne Université, CNRS, LIP6, Paris – Université Paris-Saclay,Sorbonne Universités – France

Nous présentons une approche pour améliorer l'efficacité de l'analyse statique de programmes C, fondée sur l'incrémentalité et appliquée au greffon EVA de la plateforme Frama-C. Notre approche comprend deux techniques. La première est le résumé de fonction qui permet de réutiliser les résultats d'analyses sauvegardés pour des fonctions avec des entrées similaires. La seconde est la réutilisation d'invariants de boucle, qui permet d'accélérer l'analyse des boucles en commençant les itérations à partir des invariants précédemment inférés. Ce travail contribue au groupe de travail LVP du GDR GPL. Groupe de travail : LVP

Mots-Clés: Analyse statique, Interprétation abstraite, Analyse incrémentale, Langages et vérification de programmes

*Intervenant

†Auteur correspondant: ny-andrianinamamy.razafintsialonina@cea.fr

An extensible production-level debugger

Adrien Vanègue *† [1], Steven Costiou * ‡ [1]

[1] Inria Lille - Nord Europe – Institut National de Recherche en Informatique et en Automatique – Parc Scientifique de la Haute Borne 40, avenue Halley Bât.A, Park Plaza 59650 Villeneuve dÁscq, France

Debuggers are difficult to build and evaluate.
In order to solve this problem, , the Pharo debugger needs to be modular, extensible, stable, production-level and easy to adopt.
Thus, the Pharo debugger design allows to easily extend the debugger via new debugging commands and new debugger extensions, with modular pieces that are interchangeable such as the GUI, the API implementation and the interpreter used to perform steps.
As example of debugging tools, Chest is a debugger extension that allows to share objects between several program executions, in order to, for example, compare objects across different programs.
As another example, JumpToCaret is a debugging command that allows to jump back and forth in the code under debug, while keeping the same exact program state, in order to debug if some piece of code was executed.
In the end, the Pharo debugger, which has real users, satisfies all requirements to be an extensible production-level debugger. Working Group : Debugging

Mots-Clés: Debugging, Pharo language, Tools

*Intervenant
†Auteur correspondant: adrien.vanegue@inria.fr
‡Auteur correspondant: steven.costiou@inria.fr

FML : un langage d'assemblage de modèles pour l'interopérabilité sémantique de sources d'information hétérogènes

Sylvain Guérin *† [1], Antoine Beugnard * ‡ [2], Joël Champeau * § [1]

[1] École Nationale Supérieure de Techniques Avancées Bretagne (ENSTA Bretagne) – ENSTA Bretagne – 2 rue François Verny, 29806 Brest cedex 9, France

[2] IMT Atlantique (IMT Atlantique) – Institut Mines-Télécom [Paris] – Campus Brest : Technopôle Brest-Iroise CS 8381829238 BREST Cedex 3 -Campus Nantes : 4, rue Alfred Kastler- La chantrerie 44300 NANTES -Campus Rennes : 2 Rue de la Châtaigneraie, 35510 CESSON SEVIGNE, France

Les auteurs présentent la problématique de la fédération de modèles ainsi que les enjeux liés à l'interopérabilité sémantique de sources d'information autonomes et hétérogènes. Le langage de modélisation FML répond à la problématique de la conceptualisation et de l'exécution de cette fédération. Ce langage est mis en oeuvre au sein de l'infrastructure Openflexo, dont les auteurs proposent plusieurs scénarii de démonstration. Groupe de travail : IDM

Mots-Clés: Fédération de modèles, Ingénierie Dirigée par les Modèles, Free Modelling

*Intervenant

†Auteur correspondant: sylvain.guerin@ensta-bretagne.fr

‡Auteur correspondant: antoine.beugnard@telecom-bretagne.eu

§Auteur correspondant: joel.champeau@ensta-bretagne.fr

Safe Dynamic Reconfiguration of Concurrent Component-based Applications

Salman Farhat *† [1], Simon Bliudze * ‡ [1], Laurence Duchien * § [1], Olga Kouchnarenko * ¶ [2]

[1] Inria Lille - Nord Europe – Institut National de Recherche en Informatique et en Automatique – Parc Scientifique de la Haute Borne 40, avenue Halley Bât.A, Park Plaza 59650 Villeneuve dÁscq, France
[2] Franche-Comté Électronique Mécanique, Thermique et Optique - Sciences et Technologies (UMR 6174) (FEMTO-ST) – Université de Technologie de Belfort-Montbeliard, Ecole Nationale Supérieure de Mécanique et des Microtechniques, Centre National de la Recherche Scientifique, Université de Franche-Comté – 32 avenue de l'Observatoire 25044 BESANCON CEDEX, France

Cloud applications and cyber-physical systems are becoming increasingly complex, requiring frequent reconfiguration to adapt to changing needs and requirements. Existing approaches compute new valid configurations either at design time, at runtime, or both. However, these approaches can lead to significant computational or validation overheads for each reconfiguration step. We propose a component-based approach that avoids computational and validation overheads using a representation of the set of valid configurations as a variability model. More precisely, our approach leverages feature models to automatically generate, in a component-based formalism called JavaBIP, run-time variability models that respect the feature model constraints. Produced run-time variability models enable control over application reconfiguration by executing reconfiguration requests in such a manner as to ensure the (partial) validity of all reachable configurations. We evaluate our approach on a simple web application deployed on the Heroku cloud platform. Experimental results show that the overheads induced by generated run-time models on systems involving up to 300 features are negligible, demonstrating the practical interest of our approach.

Mots-Clés: Concurrent Component, based Systems, Variability Models, Self, Configuration, Dynamic Reconfiguration

*Intervenant
†Auteur correspondant: salman.farhat@inria.fr
‡Auteur correspondant: simon.bliudze@inria.fr
§Auteur correspondant: laurence.duchien@inria.fr
¶Auteur correspondant: olga.kouchnarenko@univ-fcomte.fr

From processes to automata: compactification theorem

Benoît Ballenghien *† [1]

[1] Laboratoire Méthodes Formelles (LMF) – Institut National de Recherche en Informatique et en Automatique, CentraleSupélec, Université Paris-Saclay, Centre National de la Recherche Scientifique, Ecole Normale Supérieure Paris-Saclay – 4, avenue des Sciences, 91190, Gif-sur-Yvette, France

When working on concurrency, synchronization can be very hard to deal with. This paper aims to outline the construction of a compactification theorem which simplifies a lot some proofs about the synchronization of a finite (but unbounded) number of normalizable processes in the framework HOL-CSP.

Mots-Clés: formal verification, isabelle, HOL, CSP, proof

*Intervenant

†Auteur correspondant: benoit.ballenghien@universite-paris-saclay.fr

A Collaborative Security-by-Design approach using Model-Driven Engineering

Othmane El Karmy *† [1], Sophie Ebersold * ‡ [1], Nan Messe * § [1], Mahmoud El Hamlaoui * ¶ [1], Mahmoud Nassar * ‖ [1]

[1] Institut de recherche en informatique de Toulouse (IRIT) – Université Toulouse Capitole, Université Toulouse - Jean Jaurès, Université Toulouse III - Paul Sabatier, Centre National de la Recherche Scientifique, Institut National Polytechnique (Toulouse), Toulouse Mind Brain Institut – 118 Route de Narbonne, F-31062 Toulouse Cedex 9, France

In software development, a lack of collaboration between security experts and software engineers can result in vulnerabilities and weaknesses in software systems. This can lead to severe consequences such as data breaches, system crashes, and financial losses. Even threat modeling often lack detailed procedures and reference models for brainstorming sessions, making them sub-optimal and requiring significant effort. Therefore, a collaborative security-by-design platform is needed to actively involve all participants from the beginning, providing guidance and formalized processes for threat modeling to develop more secure and reliable software systems.

Mots-Clés: Model Driven Engineering, security by design, threat modeling, Collaborative platform

*Intervenant
†Auteur correspondant:
‡Auteur correspondant: sophie.ebersold@irit.fr
§Auteur correspondant: nan.messe@irit.fr
¶Auteur correspondant: mahmoud.elhamlaoui@ensias.um5.ac.ma
‖Auteur correspondant: mahmoud.nassar@ensias.um5.ac.ma

Interoperability and formal semantic proofs

Amélie Ledein *† [1]

[1] Deduction modulo, interopérabilité et démonstration automatique (DEDUCTEAM) – Inria Saclay - Ile de France, Laboratoire Méthodes Formelles – ENS Paris-Saclay, France

K is a semantical framework for formally describing the semantics of programming languages. It is also an environment that offers various tools to help programming with the languages specified in the formalism. It is for example possible to execute programs and to check some properties on them, using the KProver tool.

Dedukti is a logical framework allowing the interoperability of proofs between different formal proof tools. It is based on the lambda-calculus modulo theory, an extension of the type theory by adding rewriting rules in the conversion relation. The flexibility of this logical framework allows to encode many theories like 1st order logic or simple type theory.

Thanks to the logical framework Dedukti, the objective describes in this poster is to verify formal proofs about the semantics of programming languages, described in the semantical framework K, and to reuse of such proofs in different proof tools. Working group : LVP

Mots-Clés: Programming language, Semantics, Logical framework, Interoperability, Type theory, Rewriting, K framework, Dedukti

*Intervenant

†Auteur correspondant: amelie.ledein@inria.fr

Simplify interactions with models in MDE through instrumentation of model-based applications

Asbathou Biyalou-Sama *† [1]

[1] Centre de Recherche en Informatique, Signal et Automatique de Lille - UMR 9189 (CRIStAL) – Centrale Lille, Université de Lille, Centre National de la Recherche Scientifique – Université de Lille - Campus scientifique - Bâtiment ESPRIT - Avenue Henri Poincaré - 59655 Villeneuve d'Ascq, France

Since the adoption of Model-Driven Engineering (MDE), interaction with models has been an important question. The interactions with models are
actually mainly provided via modeling tools which are complex. We propose an approach in which we make model modification actions (MMA) available on
the user interface of generated applications by instrumenting these applications, in order to simplify interactions with models in MDE compilation chains. We
also pay attention to simplicity during the instrumentation of applications. Working group: IDM

Mots-Clés: MDE, Modeling, Interaction, Instrumentation, Compilation chain

*Intervenant
†Auteur correspondant: asbathou.biyalousama@univ-lille.fr