Table des matières

Conférences invitées

Planetary Limits, Anti-Limits in Computer Systems And The Missing Scenarios

Florence Maraninchi * [1]

[1] VERIMAG (VERIMAG - IMAG) – CNRS : UMR5104, Institut National Polytechnique de Grenoble - INPG, Université Joseph Fourier - Grenoble I, Institut National Polytechnique de Grenoble (INPG) – Centre Équation - 2, avenue de Vignate - 38610 GIÈRES, France

Research in computer science and computer engineering includes several branches dedicated to the environmental impacts of ICT. Green-ICT consists in improving the performances of ICT itself (software, hardware, communication infrastruture) in order to reduce its impacts; Green-by-ICT promises to reduce the impacts of other sectors thanks to ICT. In this talk we will argue that this is not sufficient. Green-ICT optimizations are often (if not always) synonymous of massive rebound effects. Green-by-ICT is nothing more than a promise, at least until now. Moreover there are intrinsic anti-limits in the design principles that make it difficult, if not impossible, to stay within planetary limits. We should start studying other, less techno-optimistic, scenarios. A somewhat extreme hypothesis is that manufacturing new hardware will stop at some point in the future. We should therefore study the "fading-ICT" scenario, using the abundant ICT resources of today to prepare a future of scarcity.

*Intervenant

Deep Software Variability and Frictionless Reproducibility

Mathieu Acher * [1]

[1] Diversity-centric Software Engineering – Inria Rennes – Bretagne Atlantique, LANGAGE ET GÉNIE LOGICIEL – France

The ability to recreate computational results with minimal effort and actionable metrics provides a solid foundation for scientific research and software development. When people can replicate an analysis at the touch of a button using open-source software, open data, and methods to assess and compare proposals, it significantly eases verification of results, engagement with a diverse range of contributors, and progress. However, we have yet to fully achieve this; there are still many sociotechnical frictions.
Inspired by David Donoho's vision, this talk aims to revisit the three crucial pillars of frictionless reproducibility (data sharing, code sharing, and competitive challenges) with the perspective of deep software variability.

Our observation is that multiple layers - hardware, operating systems, third-party libraries, software versions, input data, compile-time options, and parameters - are subject to variability that exacerbates frictions but is also essential for achieving robust, generalizable results and fostering innovation. I will first review the literature, providing evidence of how the complex variability interactions across these layers affect qualitative and quantitative software properties, thereby complicating the reproduction and replication of scientific studies in various fields.

I will then present some software engineering and AI techniques that can support the strategic exploration of variability spaces. These include the use of abstractions and models (e.g., feature models), sampling strategies (e.g., uniform, random), cost-effective measurements (e.g., incremental build of software configurations), and dimensionality reduction methods (e.g., transfer learning, feature selection, software debloating).
I will finally argue that deep variability is both the problem and solution of frictionless reproducibility, calling the software science community to develop new methods and tools to manage variability and foster reproducibility in software systems.

*Intervenant

The How and Why of Higher-Order SMT for Prospective Users

Sophie Tourret * [1,2]

[1] Max-Planck-Institut für Informatik – Allemagne
[2] Laboratoire Lorrain de Recherche en Informatique et ses Applications – L'Institut National de Recherche en Informatique et e n Automatique (INRIA), CNRS – France

SMT solvers solve the satisfiability problem in first-order logic modulo theories such as equality, linear arithmetic on real and integers, as well as data structures such as strings and bit-vectors. As such, they are perfect to reason on programs and indeed they are well-known as a favorite backend of formal methods software. A few years ago, SMT solvers were extended to higher-order logic, however the work is not over yet. How far are we from efficient, usable higher-order SMT and why would you want it? This talk is about my answers to these questions.

*Intervenant

Atelier Préparation des concours

Atelier Préparation aux concours

Paul Temple * [1]

[1] LANGAGE ET GÉNIE LOGICIEL – Univ Rennes, CNRS, Inria, IRISA - UMR 6074 – France

Cet atelier vise à démystifier autant que possible le déroulement des concours. Sur un temps assez limité, nous proposons de balayer rapidement le déroulement des concours CNRS, Inria et Mcf afin que les doctorants et doctorantes en dernière année puissent se préparer au mieux.Cet atelier n'est pas limité à ce public, puisqu'en en parlant au plus tôt, les doctorants et doctorantes les plus jeunes peuvent également réfléchir à leur projet de recherche et en discuter au plus tôt. Pour compléter ces informations, Céline Comte (classée 1ère au concours CNRS en 2022) et Julier Cailler (ayant passé ses concours cette année) ont accepté de venir témoigner et partager cette expérience. Cet atelier se veut interactif au maximum pour répondre aux questions des plus jeunes grâce au retour d'expérience de chacun.

*Intervenant

Europe needs strong software research

Europe needs strong software research

Jean-Michel Bruel * [1]

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

Le GdR GPL œuvre pour que soit reconnue la place du logiciel dans les préoccupations des différents appels à projet nationaux, européens et internationaux, de plus en plus tournés vers les applications et non plus vers les fondamentaux.

L'Europe a été à l'avant-garde avec de nombreuses innovations dans le domaine des logiciels, mais risque désormais de prendre du retard. La taille et la complexité sans cesse croissante des logiciels, l'explosion de l'utilisation de l'IA, nécessitent de toute urgence de nouvelles techniques et de nouveaux principes pour relever les défis logiciels de l'avenir, pour préserver l'autonomie et la souveraineté de l'Europe et protéger des valeurs fondamentales telles que la confidentialité, la sécurité, l'équité et l'inclusion.

C'est autour de ces préoccupations que s'est tenue une session "Europe" dans les journées nationales cette année. Cette session était composée de deux interventions.

La première a porté sur les appels à projets européens en cours et à venir en lien avec le logiciel (plus précisément du Cluster 4 sur le numérique), présentés par Smail NIAR, du Point de Contact National "Horizon Europe" au Ministère de l'Enseignement Supérieur et de la Recherche.

La deuxième a porté sur Informatics Europe, association européenne bien connue, que Jean-Marc Jézéquel, qui vient d'en prendre la présidence, nous a présentée en insistant sur l'importance de joindre nos forces pour pouvoir influencer nos institutions et nos décideurs sur l'importance d'avoir une recherche forte en génie logiciel.

*Intervenant

Prix de thèse

Prix de thèse

Pascal Poizat * [1]

[1] LIP6 – U. Paris-Nanterre, University 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 2023, le jury est présidé par Pascal Poizat et est constitué des membres suivants : B. Baudry, S. Bliudze, I. Borne, S. Chabridon, S. Conchon, S. Costiou, F. Dabrowski, S. Ebersold, A. Giorgetti, Y.-G. Guéhéneuc, D. E. Khelladi, N. Kosmatov, N. Kushik, M. Lhommeau, S. Mosser, A. Noureddine, T. Polacsek, G. Salaün, P. Temple.

Le prix de thèse GPL 2023 est attribué à Xavier Denis pour sa thèse intitulée "Deductive Verification of Rust Programs" préparée à l'Université Paris-Saclay / LMF sous la direction de Claude Marché.

Les accessits sont attribués à

Sylvain Guérin pour sa thèse intitulée "FML : un langage de fédération de modèles pour l'interopérabilité sémantique de sources d'information hétérogènes" préparée à l'ENSTA Bretagne / Lab-STICC sous la direction d'Antoine Beugnard et Joël Champeau.

et à

Youcef Remil pour sa thèse intitulée "A Data Mining Perspective on Explainable AIOps with Applications to Software Maintenance" préparée à l'Université de Lyon / INSA Lyon sous la direction de Jean-François Boulicaut.

Félicitations aux lauréats!

*Intervenant

GT IDM

Automated co-evolution of metamodels and code

Zohra Kaouter Kebaili * [1], Djamel Eddine Khelladi [2], Mathieu Acher [1], Olivier Barais [3]

[1] Institut de Recherche en Informatique et Systèmes Aléatoires – CNRS, L'Institut National de Recherche en Informatique et e n Automatique (INRIA), Université de Rennes I – France
[2] Institut de Recherche en Informatique et Systèmes Aléatoires – CNRS – France
[3] Institut de Recherche en Informatique et Systèmes Aléatoires – CNRS, L'Institut National de Recherche en Informatique et e n Automatique (INRIA), Université de Rennes I – France

In Model-Driven Engineering (MDE), a metamodel is a cornerstone artifact. A metamodel represents a high-level abstraction layer. It is used as an input for the generation of several artifacts of lower abstraction level: constraints, transformations, model instances, and code.
The code generated from the models is then enriched to add further functionalities and tooling. Eclipse Modeling Framework (EMF) is an important example of this setup. Based on a metamodel, EMF generates Java code APIs, adapters, and an editor. This generated code is then enriched by developers to offer additional functionalities and tools, such as editors, compilers, etc.

As metamodels evolve between releases, the generated code is automatically updated. As a consequence, the additional developers' code is impacted and needs to be co-evolved accordingly.

In our work, we propose a new fully automatic code co-evolution approach with the evolution of the Ecore metamodel. The approach relies on 1) pattern matching of the additional code errors 2) the abstraction gap between the metamodel elements and the code errors.
We evaluated our approach on nine Eclipse Modeling Framework projects. Results show that we automatically co-evolved 771 errors due to metamodel evolution with 631 matched and applied resolutions. Our approach reached an average of 82% of precision and 81% of recall, varying from 48% to 100% for precision and recall, respectively. Moreover, to assess whether the co-evolution of the code is performed correctly or not, we rely on running generated test cases before and after the co-evolution. We observed that the percentage of passing, failing, and erroneous tests stayed almost the same, suggesting the behavioral correctness of the co-evolution.

*Intervenant

Langages de modélisation auto-adaptables : Opportunités et Challenges

Gwendal Jouneaux * [1], Benoit Combemale [2], Olivier Barais [1], Gunter Mussbacher [3]

[1] Université de Rennes 1 – Université de Rennes I, INRIA-IRISA – France
[2] Diverse – Univ Rennes, CNRS, Inria, IRISA - UMR 6074 – France
[3] McGill University – Canada

Les systèmes logiciels évoluent aujourd'hui dans des environnements complexes et changeants, nécessitant une adaptation dynamique pour fournir au mieux les services. La communauté des systèmes auto-adaptables a fourni au fil du temps de nombreux frameworks et modèles architecturaux pour répondre à cette préoccupation. Toutefois, lorsque l'auto-adaptation est une préoccupation secondaire et ne relève donc pas du domaine d'expertise des spécialistes, les modèles architecturaux ne fournissent généralement pas de soutien et les frameworks sont limités aux langages pour lesquels ils ont été conçus. Dans le contexte des langages modélisation dédiés, la réimplémentation de ces cadriciels reste d'une complexité prohibitive. Pour répondre à ce problème, nous proposons le concept de langages auto-adaptables. Les langages auto-adaptables sont des langages ayant la capacité de modifier leur syntaxe (abstraite et/ou concrète) et leur sémantique en fonction du contexte (e.g. d'utilisation, d'exécution). Ces nouveaux langages offrent un certain nombre d'opportunités pour la création de systèmes intelligents et d'environnements de modélisation intelligents. Mais ils s'accompagnent d'un certain nombre de défis, incluant entre autres : le débogage et le test dans le cadre de sémantique adaptative et la détection d'opportunités d'évolutions de la syntaxe du langage. Dans cet exposé, je présenterai un aperçu du concept de langages auto-adaptables issu de mon travail de thèse. Puis, dans une seconde partie, je détaillerai les opportunités et challenges apportés par ces travaux.

*Intervenant

Engineering Digital Twin

Benoit Combemale * [1]

[1] Diverse – Univ Rennes, CNRS, Inria, IRISA - UMR 6074 – France

Digital twins promise tremendous potential for gaining insights, optimize operations, and improve decision-making for cyber-physical systems across various industries, including manufacturing, healthcare, transportation, and more. Recent developments show that Model-Driven Engineering (MDE) can play a central role in systematically leveraging the potential of digital twins, and many researchers from the MDE community have applied MDE technology to build digital twins in recent years. However, software engineering practices for engineering digital twins are only in their infancy. In this talk, I am presenting concrete case studies, reviewing recent contributions, as well as identifying and discussing some key remaining challenges to realize the vision.

*Intervenant

Jumeau numérique : une opportunité pour le GDR/GPL - Enquête sur la réalisation des JN

Antoine Beugnard * [1,2]

[1] Département Informatique (IMT Atlantique - INFO) – IMT Atlantique – IMT Atlantique - Campus de Brest - Technopôle Brest-Iroise CS 8381829238 BREST Cedex 3, France
[2] Lab-STICC – Lab-STICC UMR CNRS 6285, Brest – France

The digital twin is a central idea in the digitization of society. Digital twins can be seen as a new paradigm applied to systems design. However, the diversity of implementations, uses and solutions currently proposed, show that the digital twin ”object” is not yet well defined. This presentation will offer some thoughts on what a digital twin is, on its architecture and relationship with various tools. It will conclude with the presentation of a questionnaire designed to gather information on the way in which digital twins are currently being built. Le jumeau numérique est une idée centrale dans la numérisation de la société. Le jumeau numérique peut être considéré comme un nouveau paradigme appliqué à la conception de systèmes. Cependant, la diversité des mises en œuvre, des utilisations et des solutions actuellement proposées montre que " l'objet " jumeau numérique n'est pas encore bien défini. Cette présentation proposera quelques réflexions sur ce qu'est un jumeau numérique, sur son architecture et sa relation avec différents outils. Elle se terminera par la présentation d'un questionnaire destiné à recueillir des informations sur la manière dont les jumeaux numériques sont actuellement construits.

*Intervenant

Domain Specific Language Specialisation

Chiara Relevat * [1], Benoit Combemale [2], Gurvan Le Guernic [3]

[1] Institut de Recherche en Informatique et Systèmes Aléatoires – L'Institut National de Recherche en Informatique et e n Automatique (INRIA) – France
[2] Diverse – Univ Rennes, CNRS, Inria, IRISA - UMR 6074 – France
[3] 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

Domain-Specific Languages (DSLs) are languages whose syntax and semantics are tailored to address a specific problem space belonging to a specific application domain. Whenever end-users of a DSL repetitively have to work on a sub problem space of the problem space initially targeted by their DSL, they may benefit from using an even more tailored DSL targeting this specific sub problem space. However, current methods for creating DSLs are often not accessible to the vast majority of DSL end-users.
This talk introduces an envisioned specialisation process accessible to such DSL end-users that enables them to refine an existing DSL to address more specific subproblems. This process relies on a metalanguage that allows for the description of new specialised concepts' semantics using mostly the concepts of the initial DSL. This involves (1) promoting initial concepts to the metalanguage, and (2) combining them with the initial concepts of the metalanguage. In addition, as tooling is paramount to language usability and adoption, the proposed process ambition to automate the specialisation of the tooling of the initial DSL to the more specialised one. Finally, DSLs created through specialisation ought to be usable independently of their original DSL, while maintaining the ability to be themselves specialisable afterwards.
Then, the talk highlights the main challenges for such a specialisation process, and the roadmap to address them

*Intervenant

GT IE

Présentation de l'outil Maat Re

Patrick Tessier * [1]

[1] CEA-LIST – CEA-LIST, CEA LIST – France

à venir

*Intervenant

L'ingénierie des exigences et les hypothèses : de la construction incrémentale aux approches guidées par les données

Thomas Lambolais * [1]

[1] EuroMov - Digital Health in Motion – IMT Mines Alès, IMT - MINES ALES – France

à venir

*Intervenant

Exploring Goal Relationships in Satellite Assembly Line Design

Thomas Polacsek * [1], Anouk Chan *

1

[1] ONERA, Université de Toulouse [Toulouse] – ONERA – France

With the emergence of satellite constellations in recent years, satellite assembly lines have to respond to new demand : produce multiple copies of the same satellite at short production rates, manage multiple production rates for different constellations, but also continue to produce specific satellites in a single copy over longer time horizons. Designing versatile assembly lines that can assemble multiple types of satellites is a major challenge for manufacturers. Such lines must be designed within the existing infrastructure and with the specific required machines and tools, which must be shared between assembly activities for all types of satellites. In this work, we are interested in eliciting the goals of a versatile assembly chain from the perspective of goal-requirement modelling. More specifically, we want to determine which goals might conflict with each other. To do this, we propose to elicit the different goals and then use a constraint programming approach to compute if some goals are difficult to satisfy together.

*Intervenant

Exploration des relations entre les buts pour la conception d'une chaîne d'assemblage de satellites

Anouk Chan * [1], Thomas Polacsek *

1

[1] ONERA, Université de Toulouse [Toulouse] – ONERA – France

With the emergence of satellite constellations in recent years, satellite assembly lines have to respond to new demand : produce multiple copies of the same satellite at short production rates, manage multiple production rates for different constellations, but also continue to produce specific satellites in a single copy over longer time horizons. Designing versatile assembly lines that can assemble multiple types of satellites is a major challenge for manufacturers. Such lines must be designed within the existing infrastructure and with the specific required machines and tools, which must be shared between assembly activities for all types of satellites. In this work, we are interested in eliciting the goals of a versatile assembly chain from the perspective of goal-requirement modelling. More specifically, we want to determine which goals might conflict with each other. To do this, we propose to elicit the different goals and then use a constraint programming approach to compute if some goals are difficult to satisfy together.

*Intervenant

GT GLIA

Défi Inria LLM4Code

Mathieu Acher * [1,2]

[1] Institut de Recherche en Informatique et Systèmes Aléatoires – INSA Rennes – France
[2] Diversity-centric Software Engineering – L'Institut National de Recherche en Informatique et e n Automatique (INRIA) – France

Generative AI, in particular the recent Large Language Models_˜(LLMs), show great promise for software developments.
Specialized models are now able to perform an impressive variety of programming tasks: solving programming exercises, assisting software developers, or even generating mechanized proofs.
Yet, many challenges still need to be addressed to build reliable and productive LLM-based coding assistants: improving the quality of the generated code, increasing the developers' confidence in the generated code, enabling interaction with other software development tools (verification, test), and providing new capabilities (automated migration and evolution of software).
The goal of this Défi Inria is to leverage LLM capabilities to build code assistants that can enhance both reliability and productivity.
The challenge is organized along three work packages: 1) self-improving code generation, 2) evolution of existing software, and 3) interactive tools with AI-in-the-loop.

*Intervenant

Apprentissage automatique pour l'amélioration de la vérification formelle de code

Maykel Mattar * [1,2]

[1] Laboratoire de Sûreté et de sécurité des Logiciels – CEA/ DRT/LIST – France
[2] Institut de Recherche en Informatique et Systèmes Aléatoires – Université de Bretagne Sud (UBS) – France

Version française:

Dans le monde numérisé d'aujourd'hui, les logiciels jouent un rôle critique dans divers aspects de la vie humaine, de divertissement et de gestion à la finance, s'étendant à des secteurs essentiels tels que la santé et l'énergie. Cependant, cette dépendance croissante aux logiciels comporte également des risques significatifs, notamment des défaillances, des comportements indésirables et des cyber-attaques potentielles.

Frama-C est une plateforme de vérification de code source C basée sur des plugins qui mettent en œuvre diverses techniques d'analyse statique et dynamique. Frama-C est particulièrement adopté dans les domaines critiques de la sécurité par des organisations comme EDF, Thales, et de nombreux autres acteurs publics et privés.

Le plugin " Analyse de valeur évoluée " (EVA), utilise l'interprétation abstraite pour prouver l'absence de comportements non définis dans le code C. Ces comportements non définis peuvent conduire à des résultats erronés et, dans le pire des cas, à des vulnérabilités.

Les tendances récentes préconisent de remplacer les outils statiques par des modèles d'apprentissage automatique (ML). Cependant, des outils comme Frama-C privilégient une analyse rigoureuse, où même un faux négatif est inacceptable.

Une analyse EVA réussie sans alarme indique que le code respecte les normes C et qu'il n'y a pas de comportement inattendu, à l'exception d'éventuels faux positifs. Pour minimiser les faux positifs, EVA offre des paramètres configurables par l'utilisateur pour une analyse spécifique au scénario. Cependant, l'ensemble étendu de paramètres complique son utilisation, limitant l'accessibilité aux experts du domaine, réduisant ainsi le nombre de systèmes bénéficiant d'une analyse approfondie.

Au lieu de remplacer ces outils, ma thèse de doctorat vise à les améliorer en automatisant la sélection des paramètres à l'aide de techniques d'apprentissage automatique et profond, en améliorant l'efficacité et la précision de l'analyse, et en ouvrant la porte à davantage d'utilisateurs et de cas d'utilisation. Plus généralement, l'objectif est de augmenter EVA et Frama-c avec des techniques

*Intervenant

apprises automatiquement, permettant des analyses plus évolutives.

Le premier défi à relever est l'apprentissage automatique du déroulement des boucles, une technique employée par EVA pour conserver une meilleure précision lors de l'analyse d'une boucle. L'objectif est de trouver le juste équilibre entre la minimisation des faux positifs et le maintien d'un temps d'analyse efficace.

Ce défi illustre les problèmes plus généraux rencontrés dans les approches d'apprentissage automatique pour le code, tels que la traduction du code source dans un format adapté au modèle tout en préservant autant d'informations et de logique que possible, la sélection d'une approche de traitement appropriée et la validation des résultats dans le monde réel.* De plus, des défis spécifiques, tels que le déséquilibre des données à travers toutes les phases (formation, test et validation de la vérité terrain), étant donné que la probabilité d'avoir besoin de dérouler une boucle est intrinsèquement faible.

Cette présentation détaillera les techniques utilisées pour représenter et traiter les codes, suivie d'une évaluation comparative de diverses approches, y compris des modèles établis tels que XGBoost et des avancées récentes telles que les réseaux neuronaux en graphes (GNN) et les grands modèles de langage (LLM).

English Version:

In today's digitalized world, software plays a critical role in various aspects of human life, from entertainment and management to finance, extending to essential sectors such as healthcare and energy. However, this increasing reliance on software also brings about significant risks, including failures, undesired behaviors, and potential cyber-attacks.

Frama-C is a C source code verification platform that implements plugins with various static and dynamic analysis techniques. Frama-C is especially adopted in safety-critical domains by organizations like EDF, Thales, and many other public and private actors.

'Evolved Value Analysis' (EVA) plugin leverages abstract interpretation to prove the absence of undefined behaviors in C code. These undefined behaviors can lead to erroneous results and, in the worst case, vulnerabilities.

Recent trends advocate replacing static tools with machine learning (ML) models. However, tools like Frama-C prioritize rigorous analysis, where even a false negative is unacceptable.

A successful EVA analysis with no alarms indicates the code's adherence to the C standards and the absence of unintended behavior, barring potential false positives. To minimize false positives, EVA offers user-configurable parameters for scenario-specific analysis. However, the extensive parameter set complicates its use, restricting accessibility to domain experts, thus limiting the number of systems benefiting from thorough analysis.

Instead of replacing these tools, my PhD thesis aims to enhance them by automating parameter selection using machine and deep learning techniques, improving the analysis efficiency and precision, and opening the door for more users and use cases. More broadly, the objective is to augment EVA and Frama-c with automatically learned techniques, allowing more scalable analyses.

The first challenge being addressed is the automatic learning of loop unrolling, a technique employed by EVA for retaining better precision when analyzing a loop. The objective is to find the optimal balance between minimizing false positives and maintaining efficient analysis time.

This challenge exemplifies broader issues encountered in machine-learning approaches for code, like source code translation to a model-friendly format while preserving as much info and logic as possible, appropriate processing approach selection, and real-world validation of results. Additionally, specific challenges, such as data imbalance across all phases (training, testing, and ground truth validation), since the probability of needing to unroll a loop is inherently low.
This presentation will detail the techniques being developed for learning loop unrolling parametrization for EVA, from code processing and representation, followed by a benchmark of various approaches, including established models like XGBoost and recent advancements like Graph Neural Networks (GNNs) and Large Language Models (LLMs).

Code stylometry vs formatting and minification

Stefano Balla * [1], Stefano Zacchiroli [2], Gabbrielli Maurizio [1]

[1] Alma Mater Studiorum Università di Bologna = University of Bologna – Italie
[2] LTCI, Telecom Paris, Institut Polytechnique de Paris – LTCI, Télécom Paris, Institut Polytechnique de Paris – France

The automatic identification of code authors based on their programming styles-known as authorship attribution or code stylometry-has become possible in recent years thanks to improvements in machine learning-based techniques for author recognition. Once feasible at scale, code stylometry can be used for well-intended or malevolent activities, including: identifying the most expert coworker on a piece of code (if authorship information goes missing); fingerprinting open source developers to pitch them unsolicited job offers; de-anonymizing developers of illegal software to pursue them. Depending on their respective goals, stakeholders have an interest in making code stylometry either more or less effective. To inform these decisions we investigate how the accuracy of code stylometry is impacted by two common software development activities: code formatting and code minification. We perform code stylometry on Python code from the Google Code Jam dataset (59 authors) using a code2vec-based author classifier on concrete syntax tree (CST) representations of input source files. We conduct the experiment using both CSTs and ASTs (abstract syntax trees). We compare the respective classification accuracies on: (1) the original dataset, (2) the dataset formatted with Black, and (3) the dataset minified with Python Minifier. Our results show that: (1) CST-based stylometry performs better than AST-based (51.00%→68%), (2) code formatting makes a significant dent (15%) in code stylometry accuracy (68%→53%), with minification subtracting a further 3% (68%→50%). While the accuracy reduction is significant for both code formatting and minification, neither is enough to make developers non-recognizable via code stylometry.

*Intervenant

CNNGen & Towards Feature-based ML-enabled Behaviour Location

Paul Temple * [1]

[1] LANGAGE ET GÉNIE LOGICIEL – Univ Rennes, CNRS, Inria, IRISA - UMR 6074 – France

Je vais tenter de vous présenter des avancées de cette année réalisées avec l'Université de Namur.
CNNGen est un générateur d'architectures de Convolutional Neural Networks (CNN) pour la détection d'objets dans les images. La génération se fait de manière automatique grâce à une grammaire permettant de décrire la topologie du réseau. Ce travail fait parti de la thèse d'Antoine Gratia, doctorant à l'UNamur sous la supervision de Gilles Perrouin et Pierre-Yves Schobbens.
Towards Feature-based ML-enabled Behaviour Location fait suite à VaryMinions. VaryMinions cherchait à associer des traces d'éxécutions à des variants provenant d'une même ligne de produits. Cette évolution cherche plutôt à associer les features de la ligne de produits aux traces. Le processus devient moins lourd mais cela amène son lot de nouveaux challenges. Ce travail fait partie de la thèse de Sophie Fortz qui est maintenant post-doc à King's College (Londres)

*Intervenant

Options Matter: Documenting and Fixing Non-Reproducible Builds in Highly-Configurable Systems

Georges Aaron Randrianaina * [1]

[1] Univ Rennes, IRISA – Université de Rennes, Univ Rennes, CNRS, Inria, IRISA - UMR 6074 – France

A critical aspect of software development, build reproducibility, ensures the dependability, security, and maintainability of software systems. Although several factors, including the build environment, have been investigated in the context of non-reproducible builds, to the best of our knowledge the precise influence of configuration options in configurable systems has not been thoroughly investigated. This paper aims at filling this gap. This paper thus proposes an approach for the automatic identification of configuration options causing non-reproducibility of builds. It begins by building a set of builds in order to detect non-reproducible ones through binary comparison. We then develop automated techniques that combine statistical learning with symbolic reasoning to analyze over 20,000 configuration options. Our methods are designed to both detect options causing non-reproducibility, and remedy non-reproducible configurations, two tasks that are challenging and costly to perform manually. We evaluate our approach on three case studies, namely Toybox, Busybox, and Linux, analyzing more than 2,000 configurations for each of them. Toybox and Busybox come exempt from nonreproducibility. In contrast, 47% of Linux configurations lead to non-reproducible builds. The approach we propose in this paper is capable of identifying 10 configuration options that caused this non-reproducibility. When confronted to the Linux documentation, none of these are documented as non-reproducible. Thus, our identified non-reproducible configuration options are novel knowledge and constitutes a direct, actionable information improvement for the Linux community. Finally, we demonstrate that our methodology effectively identifies a set of undesirable option values, enabling the enhancement and expansion of the Linux kernel documentation while automatically rectifying 96% of encountered non-reproducible builds.

*Intervenant

A Performance Study of LLM-Generated Code on Leetcode

Tristan Coignion * [1], Clément Quinton [2], Romain Rouvoy [3]

[1] Inria Lille - Nord Europe – Université de Lille - Sciences et Technologies – France
[2] Self-adaptation for distributed services and large software systems (SPIRALS) – Univ. Lille, CNRS, Inria, Centrale Lille, UMR 9189 CRIStAL – F-59000 Lille, France
[3] Univ. Lille, CNRS, Inria – Inria Lille - Nord Europe, Centre de Recherche en Informatique, Signal et Automatique de Lille - UMR 9189 – France

This study evaluates the efficiency of code generation by LLMs and measures their performance against human-crafted solutions using a dataset from Leetcode.
We compare 18 LLMs, considering factors such as model temperature and success rate, and their impact on code performance.
This research introduces a novel method for measuring and comparing the speed of LLM-generated code, revealing that LLMs produce code with comparable performance, irrespective of the adopted LLM.
We also find that LLMs are capable of generating code that is, on average, more efficient than the code written by humans.
The paper further discusses the use of Leetcode as a benchmarking dataset, the limitations imposed by potential data contamination, and the platform's measurement reliability.
We believe that our findings contribute to a better understanding of LLM capabilities in code generation and set the stage for future optimizations in the field.

*Intervenant

Modelling for citizens with citizens. Building accessible and reliable software for agent-based modelling

Oleksandr Zaitsev * [1,2]

[1] Savoirs, ENvironnement et Sociétés – CIRAD, CIRAD – France
[2] Département Environnements et Sociétés – CIRAD – France

Agent-based models (ABM) are computer systems that simulate the actions and interactions between autonomous agents. At UMR SENS, we practice an inclusive approach to modelling (ComMod), which involves local stakeholders in model design, simulation analysis, and even decision-making. Cormas is an agent-based modelling platform that we develop at UMR SENS, which is highly interactive and particularly well-suited for the ComMod approach. My talk will consist of two parts. First, I will present the ComMod approach and Cormas platform. Then, I will discuss some open research questions related to developing accessible and reliable agent-based modelling software. What are the lessons learned from over 25 years of Cormas development? Why do we need a meta-model for ABM? Do we need to debug and test models? Is there software evolution in the context of ABM? How can we make modelling more interactive through board games and chat applications?

*Intervenant

IA et métier, séparation des préoccupations au cœur du logiciel

Sylvain Lejambe * [1]

[1] Université Savoie Mont Blanc – LISTIC USMB, LOCIE laboratory - Polytech Annecy-Chambéry – France

The challenge of designing self-adaptive systems is crucial in the rapidly evolving technological landscape, where applications need to continually adapt to changes and user demands. Starting from scratch each time is not feasible due to the complexity and resource demands; hence, developers frequently rely on frameworks to build upon existing technologies rather than duplicating efforts. The Wise Object Framework (WOF) assists developers in building self-adaptive systems by adding adaptability to any object-oriented software. The WOF dynamically creates agents (WiseObjects) to manage actual software objects and equips them with capabilities for logging, analysis, and self-correction. Developers can automatically generate WOs for each class instance simply by annotating the relevant source class. The WOF ensures a clear separation of concerns, maintaining the business application unchanged while integrating new intelligent components. The WOF also features a unique dual-state mechanism, where WOs operate in "Awake" and "Dream" states. In the Awake state, WOs perform standard operations and interact with the system. In the Dream state, they engage in introspection, analyzing performance, and planning adaptations without affecting real-world processes. In this paper, we introduce the Wise Object Framework (WOF) and analyze its strengths and weaknesses by transforming a basic banking authorization application into an adaptive credit card fraud detection system.

*Intervenant

GT Yoda & CLAP

Fast Choreography of Cross-DevOps Reconfiguration with Ballet

Jolan Philippe * [1]

[1] IMT Atlantique – IMT Atlantique, Inria, LS2N, UBL, F-44307 Nantes, France – France

In the context of Edge Computing or Cyber-Physical Systems, cross-functional, and cross-geographical DevOps teams are in charge of automating deployments, configuration, and management (i.e., reconfiguration) of complex, large-scale, highly dynamic, and geo-distributed service-oriented software systems. In this context, DevOps teams cannot reasonably manually coordinate their reconfiguration operations in a global manner. Furthermore, as disconnection is the norm in these paradigms, a central entity responsible for reconfiguration should be avoided, and the set of changes to apply should be as fast as possible. This talk presents Ballet, a fast tool to automate decentralized choreographies (i.e., coordination) of cross-DevOps reconfiguration. We show a gain of 42.6% for a deployment scenario and 24% for an update scenario on an OpenStack case study.

*Intervenant

Génération automatique de code haute performance prévisible: de l'algèbre des tableaux au code vectorisé et multicoeur

Gaétan Hains * [1]

[1] Université Paris Est Créteil – Université Paris-Est Créteil Val-de-Marne (UPEC) – France

High-performance architectures have complex features so that the reliable production of parallel software remains beyond the reach of most Computer Science graduates. Compilers alone cannot guarantee the highest performance and multiple APIs with complex performance features are difficult to master.
As a first step towards more comprehensive solutions we are building key elements of a precompiler system that will automatically produce predictable, scalable and high-performance code from declarative tensor expressions. In this paper we summarize and analyze a large set of timing experiments of matrix multiplication variants that are mapped to vectorized and multithread code. The analysis covers two high-end target architectures and exhaust a whole space of code, compiler, pragma and parallelism parameters. Our analysis shows how the best choice of parameters is produced from a small set of tests that can converge in a matter of seconds and then predict performance of larger instances to within 25% or much less. Inefficient choices of parameters is also shown to be reliably predicted from small tests, so that our design for a precompiler is guaranteed to be a realistic and portable tool. The generality of our Mathematics of Arrays tensor algebra, and very broad applicability of tensor operations (signal processing, scientific computing, AI, etc) supports our claim that these experiments and design can be generalized to a general purpose parallel programming tool.

*Intervenant

GT Debugging

Testing Framework for scientific computing : proposing new software testing approaches for reliable computationally intensive software systems

Ewen Brune * [1]

[1] Inria – L'Institut National de Recherche en Informatique et e n Automatique (INRIA) – France

-

*Intervenant

Scopeo: an object-centric debugging approach for exploring object-oriented programs

Valentin Bourcier * [1]

[1] INRIA – Inria Lille-Nord Europe – France

-

*Intervenant

Debugging Activity Blueprint: vizualisations to understand how developers debug

Alexandre Bergel * [1]

[1] RelationalAI – Suisse

-

*Intervenant

Object-Centric Debugging

Steven Costiou * [1]

[1] inria – Inria Lille - Nord Europe, CNRS – France

-

*Intervenant

Un protocole à Meta-Object pour l'implémentation de debuggers centrés objets

Rémi Dufloer * [1]

[1] inria – Inria Lille - Nord Europe, CNRS – France

-

*Intervenant

GT Logiciel Éco-Responsable

Software Frugality in an Accelerating World: the Case of CI/CD

Quentin Perez * [1]

[1] INSA Rennes – L'Institut National de Recherche en Informatique et e n Automatique (INRIA) – France

Software Frugality in an Accelerating World: the Case of CI/CD

*Intervenant

Analyse des compromis entre performance et consommation d'énergie des frameworks Java de mapping objet-relationnel

Alexandre Bonvoisin * [1]

[1] Inria – L'Institut National de Recherche en Informatique et e n Automatique (INRIA) – France

Analyse des compromis entre performance et consommation d'énergie des frameworks Java de mapping objet-relationnel

*Intervenant

Rapport d'activité et bilan 2023-2024 du GT Logiciel Eco-Responsable

Adel Noureddine [1], Olivier Le Goaer [1], Florence Maraninchi * [2], Romain Rouvoy [3]

[1] Université de Pau – Laboratoire d'Informatique LIUPPA, Anglet, EA3000 – France
[2] Université de Grenoble – Verimag, CNRS, Grenoble – France
[3] Université de Lille – L'Institut National de Recherche en Informatique et e n Automatique (INRIA) – France

Rapport d'activité et bilan 2023-2024 du GT Logiciel Eco-Responsable

*Intervenant

GT VL

A manual categorization of new quality issues on automatically-generated tests

Geraldine Galindo-Gutierrez [1], Maximiliano Narea Carvajal [2], Alison Fernandez Blanco [2], Nicolas Anquetil * [3], Juan Pablo Sandoval Alcocer [2]

[1] Bolivian Catholic University – Bolivie
[2] Pontificia Universidad Catolica de Chile – Chili
[3] Centre de Recherche en Informatique, Signal et Automatique de Lille - UMR 9189 – Université des Sciences et Technologies de Lille - Lille I, L'Institut National de Recherche en Informatique et e n Automatique (INRIA), CNRS – France

Diverse studies have analyzed the quality of automatically generated test cases by using test smells as the main quality attribute. But recent work reported that generated tests might suffer from a number of quality issues not considered previously. In this paper, we report on a manual analysis of an external dataset consisting of 2,340 automatically generated tests. As a result, we propose a taxonomy of 13 new quality issues grouped in four categories. We also report on the frequency of these new quality issues within the dataset and present eight recommendations that test generators may consider to improve the quality and usefulness of the automatically generated tests. Our results suggest that (i) test quality should be evaluated not only on the tests themselves, but considering also the tested code; and (ii) automatically generated tests present flaws that are unlikely to be found in manually created tests and thus require specific quality checking tools.

*Intervenant

Lightweight Syntactic API Usage Analysis with UCov

Gustave Monce * [1]

[1] Laboratoire Bordelais de Recherche en Informatique – L'Institut National de Recherche en Informatique et e n Automatique (INRIA), Université de Bordeaux (Bordeaux, France), Centre national de la recherche scientifique - CNRS (France), Bordeaux INP, G – France

Designing an effective API is essential for library developers as it is the lens through which clients will judge its usability and benefits, as well as the main friction point when the library evolves. Despite its importance, defining the boundaries of an API is a challenging task, mainly due to the diverse mechanisms provided by programming languages that have non-trivial interplays. In this paper, we present a novel conceptual framework designed to assist library maintainers in understanding the interactions allowed by their APIs via the use of syntactic usage models. These customizable models enable library maintainers to improve their design ahead of release, reducing friction during evolution. The complementary syntactic usage footprints and coverage scores, inferred from client code using the API (e.g., documentation samples, tests, third-party clients), enable developers to understand in-the-wild uses of their APIs and to reflect on the adequacy of their tests and documentation. We present an implementation of these models for Java libraries in the tool UCov and demonstrate its capabilities on three Java libraries exhibiting diverse styles of interaction: Jsoup, Apache commons-cli, and Spark. Our case study shows that UCov provides valuable information regarding API design and fine-grained analysis of client code to identify under-tested and under-documented library code.

*Intervenant

Polyglot programming: static analysis and test

Philémon Houdaille * , Djamel Eddine Khelladi [1], Benoit Combemale [2], Gunter Mussbacher [3]

[1] Institut de Recherche en Informatique et Systèmes Aléatoires – CNRS – France
[2] Diverse – Univ Rennes, CNRS, Inria, IRISA - UMR 6074 – France
[3] McGill University – Canada

Le dévelopement logiciel moderne implique des projets de taille et complexité toujours plus importantes. Pour répondre à la variété des problèmes à résoudre au sein d'un même logiciel, une solution possible est la programmation polyglotte, un style de programmation dont le principe est de mélanger et coordonner différents langages informatiques. Cependant malgré la popularité grandissante de cette pratique, la plupart des travaux existants se concentrent sur des problématiques d'exécution de programmes polyglottes. Cette présentation se focalise donc sur un autre aspect essentiel à l'activité de la programmation, l'outillage développeur, sous deux angles principaux : l'analyse statique et le test dynamique en boîte blanche, dans un context de programmation polyglotte.

*Intervenant

GT GLSec

Apprentissage automatique pour l'amélioration de la vérification formelle de code

Maykel Mattar * [1,2]

[1] Laboratoire de Sûreté et de sécurité des Logiciels – CEA/ DRT/LIST – France
[2] Institut de Recherche en Informatique et Systèmes Aléatoires – Université de Bretagne Sud (UBS) – France

Version française:

Dans le monde numérisé d'aujourd'hui, les logiciels jouent un rôle critique dans divers aspects de la vie humaine, de divertissement et de gestion à la finance, s'étendant à des secteurs essentiels tels que la santé et l'énergie. Cependant, cette dépendance croissante aux logiciels comporte également des risques significatifs, notamment des défaillances, des comportements indésirables et des cyber-attaques potentielles.

Frama-C est une plateforme de vérification de code source C basée sur des plugins qui mettent en œuvre diverses techniques d'analyse statique et dynamique. Frama-C est particulièrement adopté dans les domaines critiques de la sécurité par des organisations comme EDF, Thales, et de nombreux autres acteurs publics et privés.

Le plugin " Analyse de valeur évoluée " (EVA), utilise l'interprétation abstraite pour prouver l'absence de comportements non définis dans le code C. Ces comportements non définis peuvent conduire à des résultats erronés et, dans le pire des cas, à des vulnérabilités.

Les tendances récentes préconisent de remplacer les outils statiques par des modèles d'apprentissage automatique (ML). Cependant, des outils comme Frama-C privilégient une analyse rigoureuse, où même un faux négatif est inacceptable.

Une analyse EVA réussie sans alarme indique que le code respecte les normes C et qu'il n'y a pas de comportement inattendu, à l'exception d'éventuels faux positifs. Pour minimiser les faux positifs, EVA offre des paramètres configurables par l'utilisateur pour une analyse spécifique au scénario. Cependant, l'ensemble étendu de paramètres complique son utilisation, limitant l'accessibilité aux experts du domaine, réduisant ainsi le nombre de systèmes bénéficiant d'une analyse approfondie.

Au lieu de remplacer ces outils, ma thèse de doctorat vise à les améliorer en automatisant la sélection des paramètres à l'aide de techniques d'apprentissage automatique et profond, en améliorant l'efficacité et la précision de l'analyse, et en ouvrant la porte à davantage d'utilisateurs et de cas d'utilisation. Plus généralement, l'objectif est de augmenter EVA et Frama-c avec des techniques

*Intervenant

apprises automatiquement, permettant des analyses plus évolutives.

Le premier défi à relever est l'apprentissage automatique du déroulement des boucles, une technique employée par EVA pour conserver une meilleure précision lors de l'analyse d'une boucle. L'objectif est de trouver le juste équilibre entre la minimisation des faux positifs et le maintien d'un temps d'analyse efficace.

Ce défi illustre les problèmes plus généraux rencontrés dans les approches d'apprentissage automatique pour le code, tels que la traduction du code source dans un format adapté au modèle tout en préservant autant d'informations et de logique que possible, la sélection d'une approche de traitement appropriée et la validation des résultats dans le monde réel.* De plus, des défis spécifiques, tels que le déséquilibre des données à travers toutes les phases (formation, test et validation de la vérité terrain), étant donné que la probabilité d'avoir besoin de dérouler une boucle est intrinsèquement faible.

Cette présentation détaillera les techniques utilisées pour représenter et traiter les codes, suivie d'une évaluation comparative de diverses approches, y compris des modèles établis tels que XGBoost et des avancées récentes telles que les réseaux neuronaux en graphes (GNN) et les grands modèles de langage (LLM).

English Version:

In today's digitalized world, software plays a critical role in various aspects of human life, from entertainment and management to finance, extending to essential sectors such as healthcare and energy. However, this increasing reliance on software also brings about significant risks, including failures, undesired behaviors, and potential cyber-attacks.

Frama-C is a C source code verification platform that implements plugins with various static and dynamic analysis techniques. Frama-C is especially adopted in safety-critical domains by organizations like EDF, Thales, and many other public and private actors.

'Evolved Value Analysis' (EVA) plugin leverages abstract interpretation to prove the absence of undefined behaviors in C code. These undefined behaviors can lead to erroneous results and, in the worst case, vulnerabilities.

Recent trends advocate replacing static tools with machine learning (ML) models. However, tools like Frama-C prioritize rigorous analysis, where even a false negative is unacceptable.

A successful EVA analysis with no alarms indicates the code's adherence to the C standards and the absence of unintended behavior, barring potential false positives. To minimize false positives, EVA offers user-configurable parameters for scenario-specific analysis. However, the extensive parameter set complicates its use, restricting accessibility to domain experts, thus limiting the number of systems benefiting from thorough analysis.

Instead of replacing these tools, my PhD thesis aims to enhance them by automating parameter selection using machine and deep learning techniques, improving the analysis efficiency and precision, and opening the door for more users and use cases. More broadly, the objective is to augment EVA and Frama-c with automatically learned techniques, allowing more scalable analyses.

The first challenge being addressed is the automatic learning of loop unrolling, a technique employed by EVA for retaining better precision when analyzing a loop. The objective is to find the optimal balance between minimizing false positives and maintaining efficient analysis time.

This challenge exemplifies broader issues encountered in machine-learning approaches for code, like source code translation to a model-friendly format while preserving as much info and logic as possible, appropriate processing approach selection, and real-world validation of results. Additionally, specific challenges, such as data imbalance across all phases (training, testing, and ground truth validation), since the probability of needing to unroll a loop is inherently low.
This presentation will detail the techniques being developed for learning loop unrolling parametrization for EVA, from code processing and representation, followed by a benchmark of various approaches, including established models like XGBoost and recent advancements like Graph Neural Networks (GNNs) and Large Language Models (LLMs).

Un métamodèle outillé pour assister l'ingénierie logicielle dans la protection de la vie privée des utilisateurs

Selena Lamari * [1,2]

[1] Laboratoire d'Informatique de Robotique et de Microélectronique de Montpellier – CNRS, Université de Montpellier – France
[2] Laboratoire de Recherche pour le Developpement des Systemes Informatises [Blida] – Algérie

Alors que les systèmes logiciels deviennent de plus en plus personnalisés pour les utilisateurs, la collecte et l'exploitation des données personnelles ont atteint des niveaux sans précédent. Cette évolution souligne un problème critique sur la protection de la vie privée des utilisateurs. Le respect de réglementations en matière de protection des données personnelles, comme le règlement général européen sur la protection des données (RGPD), est devenu une nécessité juridique. Malgré les obligations légales, il subsiste un manque de méthodes universellement acceptées pour intégrer la gestion de la vie privée dans les applications. Pour aider les développeurs, notre objectif principal est d'introduire une approche d'ingénierie logicielle outillée qui intègre des fonctionnalités de protection des données personnelles dans les applications existantes, les rendant ainsi respectueuses de la vie privée. La particularité de notre proposition réside dans son caractère actionnable, qui va au-delà d'une simple description déclarative de la régulation. Notre travail consiste à présenter les fondements de cette approche : le métamodèle PRIAM (Privacy Assessment Model) et ses artefacts associés. PRIAM modélise le RGPD en capturant les concepts décrits dans le règlement. Ce métamodèle sert de base à divers artefacts (user stories génériques, schéma de BD) qui s'adaptent à toute application spécifique qui nécessite de tenir compte du respect de la vie privée. Notre approche comprend un langage spécifique au domaine (DSL) dérivé du métamodèle PRIAM, des user stories adaptées à l'application spécifique et une base de données qui stocke à la fois les données spécifiques à l'application et à l'utilisateur nécessaires à la mise en œuvre des droits des utilisateurs en matière de respect de la vie privée.

*Intervenant

Formally verified hardening of C programs against fault injection

Sylvain Boulme [1], David Monniaux [2], Basile Pesin * [3], Marie-Laure Potet [4]

[1] VERIMAG – Université Grenoble Alpes, CNRS, Institut polytechnique de Grenoble (Grenoble INP) – France
[2] VERIMAG – Université Grenoble Alpes, CNRS, Institut polytechnique de Grenoble (Grenoble INP) – France
[3] VERIMAG – Université Grenoble Alpes, CNRS, Institut polytechnique de Grenoble (Grenoble INP) – France
[4] VERIMAG – Université Grenoble Alpes, CNRS, Institut polytechnique de Grenoble (Grenoble INP) – France

Fault attacks allow malicious actors to modify the behavior of a program by physically injecting a fault in the hardware. They typically target sensitive applications such as cryptography services, authentication or boot-loader and firmware updater. They can be defended against by adding countermeasures, that is control flow checks and redundancies, either in the hardware, or in the software running on it. In particular, software countermeasures may be added automatically during compilation.

In this talk, we will describe a formally verified implementation of this approach in the CompCert verified compiler for the C language. We proposed a toolkit to implement countermeasures as transformations of a middle-end representation of CompCert, RTL. We applied this toolkit to two existing countermeasures that protect the control flow of the program. We proved that these countermeasures are correct, that is, they do not change the observable behavior of the program during an execution without fault injection. We then modeled the effect of a fault on the behavior of the program as an extension of the semantic model of RTL. We used this new model to formally prove the efficacy of the countermeasure: all attacks are caught. In addition to this formal reasoning, we evaluated the protected program using Lazart, a tool for symbolic fault injection.

*Intervenant

GT LVP - AFADL

Guided Equality Saturation

Thomas Koehler * [1], Andrés Goens [2], Siddharth Bhat [3], Tobias Grosser [4], Phil Trinder [5], Michel Steuwer [6]

[1] Inria – Universite de Strasbourg, CNRS, ENGEES, ICube UMR 7357 – France
[2] University of Amsterdam – Pays-Bas
[3] University of Edinburgh – Royaume-Uni
[4] University of Cambridge [UK] – Royaume-Uni
[5] University of Glasgow – Royaume-Uni
[6] Technische Universität Berlin – Allemagne

Je propose un résumé long de notre papier accepté à POPL 2024: https://doi.org/10.1145/3632900
La réécriture de termes est une technique simple mais puissante, avec des utilisations variées comme la compilation ou la preuve de théorèmes. Pour la preuve de théorème, chaque réécriture est une étape de preuve; pour la compilation, chaque réécriture est une étape d'optimisation. Il est possible d'écrire des séquences de réécriture manuellement, mais ceci n'est pas adapté pour des séquences trop longues. Les techniques automatiques, comme la simplification gloutonne ou la saturation d'égalité, fonctionnent sans intervention humaine. Cependant, elles ne passent pas à l'échelle d'espaces de recherche vastes et complexes, ce qui limite la complexité des tâches pour lesquelles l'automatisation est efficace : une augmentation de la taille des termes ou de la longueur de la séquence de réécriture peut entraîner un echec.

Nous proposons la saturation d'égalité guidée, une technique de réécriture de termes semi-automatique qui passe la technique entièrement automatisée de saturation d'égalité à l'échelle grâce à l'assistance humaine. Le programmeur fournit un guide intermédiaire, et la réécriture est divisée en deux étapes plus simples : de la source au guide, et du guide à la cible. Une tâche de réécriture complexe peut nécessiter plusieurs guides, ce qui résulte en une séquence d'étapes de saturation d'égalité. Un guide n'a pas besoin d'être un terme complet, pouvant aussi être une esquisse qui contient des éléments indéfinis qui seront instantiés par saturation d'égalité. De telles esquisses peuvent être beaucoup plus concises que le terme complet.

Nous démontrons la généralité et l'efficacité de la saturation d'égalité guidée avec deux études de cas. Premièrement, dans le compilateur Shine du langage fonctionnel Rise, la saturation d'égalité non guidée ne parvient pas à effectuer des optimisations avancées étant donné une heure et 60Go de mémoire. La saturation d'égalité guidée effectue ces mêmes optimisations en quelques secondes et moins d'1 Go, grâce à 3 guides. Chaque guide est une esquisse 10 fois plus petite que le programme complet. Deuxièmement, nous introduisons une tactique pour l'assistant de preuve Lean 4 basée sur la saturation d'égalité guidée, permettant d'écrire des preuves dans un style similaire à celui trouvé dans un manuel : une série de calculs qui omettent des détails et sautent des étapes. Notre tactique conclut en fractions de secondes au lieu de minutes, par rapport à la saturation d'égalité non guidée, et trouve des preuves complexes qui devaient auparavant être réalisées manuellement.

*Intervenant

Amélioration des raisonneurs du langage B avec des techniques SAT et SMT

Vincent Trélat * [1]

[1] Université de Lorraine, CNRS, Inria, LORIA – Université de Lorraine, CNRS, Inria, LORIA, F-54000, Nancy, France – France

Les récents progrès en déduction automatisée ont permis l'extension des techniques de déduction au premier ordre à la logique d'ordre supérieur (HOL). L'objectif principal de ma thèse est de rendre ces avancées disponibles pour la méthode B, augmentant ainsi de manière significative le degré d'automatisation des preuves.

*Intervenant

Mieux automatiser la vérification déductive avec des stratégies de preuve dans Frama-C/WP

Loïc Correnson [1], Allan Blanchard [1], Adel Djoudi [2], Nikolai Kosmatov * [3]

[1] Université Paris-Saclay, CEA, List – Université Paris-Saclay, CEA, List, F-91120, Palaiseau, France – France
[2] Thales Digital Identity Security – Thales Digital Identity and Security, Meudon, France – France
[3] Thales Research and Technology – THALES – France

Ces dernières années, la vérification déductive a été appliquée avec succès dans de nombreuses études. Les outils modernes arrivent à prouver automatiquement la grande majorité des buts de preuve (ou conditions de vérification). Un but resté non prouvé nécessite une preuve interactive: soit avec un script (partiel) de preuve (qui indique quelques premiers pas de preuve avant de le confier aux solveurs automatiques), soit une preuve interactive complète (par exemple, dans un assistant de preuve Coq). Le besoin de réaliser des preuves interactives reste un obstacle majeur sur le chemin vers une plus large application de la vérification déductive dans le cadre industriel. Cet article présente une nouvelle extension de l'outil Frama-C/WP pour les stratégies de preuve afin de mieux automatiser la preuve pour les conditions de vérification non prouvées par les solveurs automatiques. Cette soumission est un résumé étendu de l'article court (outil) publié à TACAS 2024.

*Intervenant

GT MTV2 - AFADL

Guiding Symbolic Execution with A-star

Theo De Castro Pinto * [1,2], Antoine Rollet [3], Grégoire Sutre [3], Ireneusz Tobor [4]

[1] Laboratoire Bordelais de Recherche en Informatique – CNRS, Univ. Bordeaux – France
[2] SERMA Technologies – - – France
[3] Laboratoire Bordelais de Recherche en Informatique – CNRS – France
[4] SERMA Technologies – - – France

Symbolic execution is widely used to detect vulnerabilities in software. The idea is to symbolically execute the program in order to find an executable path to a target instruction. For the analysis to be fully accurate, it must be performed on the binary code, which makes the well-known issue of state explosion even more critical. In this paper, we introduce a novel exploration strategy for symbolic execution aiming to limit the number of explored paths. Our strategy is inspired from the A$\star$ algorithm and steered towards least explored parts of the program. We compare our approach, using the Binsec tool, to three other classical strategies: depth-first (DFS), breadth-first (BFS) and non-uniform random (NURS). Our experiments on real-size programs show that our approach is promising.

*Intervenant

Sécurisation de services via des techniques de healing et d'encapsulations

Jarod Sue * [1], Sébastien Salva [1]

[1] Laboratoire d'Informatique, de Modélisation et d'Optimisation des Systèmes – UMR CNRS 6158, Clermont Auvergne University, UCA – France

Les compositions de services sont largement adoptées dans l'industrie depuis au moins une décennie.
Force est de constater que l'intérêt pour leur développement est souvent inversement proportionnel à leur sécurisation, ce qui expose les développeurs et services à devenir les proies d'attaquants.
Cet article présente une approche pour aider les entreprises à rendre résilient des services ayant diverses faiblesses (weaknesses) de sécurité grâce une nouvelle technique de guérison (software healing). De ce fait, cette approche semi-supervisée ne modifie pas le code original mais utilise des techniques d'encapsulation d'un service pour proposer un service mieux sécurisé.
Celui-ci propose donc la définition d'opérateurs de healing de sécurité composé de 2 mitigations, une construite via un générateur de "prompt" pour IA génératives, et une seconde de repli se basant sur des solutions connues mais plus limitative quand à l'accès au service.
Cet article propose également un algorithme de healing utilisant des ensembles de test et les opérateurs de healing.
L'approche sera illustrée avec un exemple de services et de weakness.
Finalement, sachant que le travail présenté est en cours d'étude, l'article présentera des perspectives de travail inhérentes à la confiance qui peut être apportée aux solutions de healing générées par IA et à l'évaluation de notre algorithme.

*Intervenant

Le projet TAGAda: Tests Automatiquement Générés pour Ada

Delphine Longuet * [1], Claire Dross [2]

[1] Thales Research Technology – - – France
[2] AdaCore – - – France

Ce papier présente le projet RAPID TAGAda entre Thales Research & Technologies et AdaCore qui a commencé en juin 2021 et se termine en mai 2024. Son objectif est de développer un outil de génération automatique de tests pour le langage Ada, en combinant ces différentes stratégies de génération afin d'atteindre une couverture du code maximale. La problématique principale réside dans la collaboration des différentes techniques de génération de tests (génération guidée par les types, fuzzing, exécution symbolique) en vue d'améliorer la couverture du code et des contrats par les tests. L'objectif est de tirer le meilleur parti de chacune de ces techniques pour le langage Ada.

*Intervenant

AFADL

Valider un système composé de modèles indépendants

Jean-Pierre Jacquot * [1]

[1] LORIA – LORIA Nancy - CNRS, LORIA Nancy - CNRS : Université de Lorraine – France

La modélisation en B-événementiel de systèmes formés de composants indépendants, homogènes ou non, n'est pas un exercice simple. Les opérations de décomposition et recomposition de modèles sont définies théoriquement mais leur usage en pratique est difficile. RODIN offre peu de support, de même que pour les CPS. Nous décrivons ici une approche pragmatique pour valider un système modélisé en plusieurs composants indépendants en utilisant l'outil JeB. Nous proposons un protocole basé sur les websockets qui permet l'échange de variables entre deux modèles formels. Ce travail en cours utilise l'étude de cas du Respirateur proposée par la conférence ABZ2024.

*Intervenant

Combiner la Vérification Déductive avec l'analyse de forme

Teo Bernier * [1], Yani Ziani *

[2], Nikolai Kosmatov *

[2], Frédéric Loulergue [3]

[1] Thales Research Technology – THALES – France
[2] Thales Research Technology – THALES – France
[3] Univ. Orléans, INSA CVL, LIFO EA 4022 – Univ. Orléans – France

Des outils de vérification déductive ont pu être utilisés avec succès dans de nombreuses études de cas pour prouver un large panel de propriété de sûreté, sécurité, ainsi que des propriétés fonctionnelles.

De tels outils rencontrent souvent des difficultés à procéder à des preuves automatiques dans des codes manipulant des {structures de données récursives (par exemple les listes chaînées, les arbres, etc.), en particulier, du fait des modèles mémoire complexes dont ils ont besoin. L'utilisateur doit alors guider la preuve par des lemmes prouvés interactivement, des assertions, etc.

Les outils d'interprétation abstraite basés sur la logique de séparation et l'analyse de forme peuvent raisonner efficacement sur de telles structures, mais ne peuvent typiquement pas gérer de larges classe de propriétés. L'article résumé ici présente de nouvelles idées et les premiers résultats dans une tentative de combinaison de ces
deux techniques pour en garder le meilleur de chacune.

*Intervenant

Compilation avec l'interprétation abstraite

Dorian Lesbre * [1], Matthieu Lemerre [2]

[1] Université Paris-Saclay, CEA, List – Université Paris-Saclay, CEA, LIST, Université Paris-Saclay, CEA List – France
[2] CEA List, Université Paris-Saclay – Université Paris-Saclay,Sorbonne Universités – France

Compiling with abstract interpretation, accepté à PLDI 2024. Ce papier présente une technique pour transformer un interpréteur abstrait en compilateur à l'aide d'une algèbre libre sur la signature des domaines abstraits. Les passes de compilation sont encodées par des foncteurs sur les domaines abstraits. Leur correction correspond à une simulation avant, et leur complétude à une simulation arrière. Nous utilisons cette technique pour compiler le code analysé vers une variante de SSA. Cette transformation permet d'améliorer la précision de notre analyseur : notamment, utiliser un domaine numérique non-relationnel basé sur la forme SSA est toujours plus précis qu'un domaine standard, sans pour autant en augmenter le coût. De plus, un tel domaine permet d'analyser du code machine avec la même précision que l'analyse du code source, sans souffrir de la perte d'information due à la compilation. Cette technique aide à combiner des transformations de programmes et des analyses sémantiques en une seule passe, ce qui est plus précis que de les faire séquentiellement.

*Intervenant

Utilisation conjointe de SysML et Reo en vue de modeliser et de valider les CPS

Perla Tannoury * [1], Ahmed Hammad France [2]

[1] FEMTO-ST Institute – Université de Bourgogne Franche-Comté (UBFC)-UTBM – France
[2] FEMTO-ST Institute – Université de Bourgogne Franche-Comté (UBFC)-UTBM – France

Les systèmes cyber-physiques (CPS) sont cruciaux dans la santé, les villes intelligentes et les véhicules autonomes, mais leur modélisation est complexe en raison de la diversité de leurs composants et interactions. Notre approche, SysReo, fusionne SysML et Reo pour une modélisation complète des CPS, avec une extension, Timed SysReo, pour gérer les contraintes de temps, offrant ainsi des solutions plus robustes et efficaces pour la conception et la validation des CPS.

*Intervenant

Model-Based Fuzz Testing for GNSS Receiver

Haag Nina * [1], Daniel Prun [2], Antoine Blais [3]

[1] Fédération ENAC ISAE-SUPAERO ONERA – Université de Toulouse, CNRS, INSA, ISAE-SUPAERO, Mines Albi, UPS, Toulouse France – France
[2] Fédération ENAC ISAE-SUPAERO ONERA – Université de Toulouse, CNRS, INSA, ISAE-SUPAERO, Mines Albi, UPS, Toulouse France – France
[3] Fédération ENAC ISAE-SUPAERO ONERA – PRES Université de Toulouse – France

GNSS receivers are vital for aircraft navigation system reliability and safety. However, traditional test methods recommended by norms have limitations in verifying their expected properties. This article presents a thesis aimed at enhancing the testing process by integrating Model-Based Testing (MBT) and Fuzz Testing approaches. Our approach involves leveraging formal models, including behavioral models (e.g., Finite State Machines, Sequence diagrams) and static ones (e.g., OCL, BNF-based grammar descriptions), to generate relevant test cases through a dedicated mutation process. We intend to validate our approach by developing a dedicated framework for GNSS receiver verification, focusing on the critical RAIM (Receiver Autonomous Integrity Monitoring) function.

*Intervenant

Spécification et Vérification de propriétés Typestates avec Frama-C

Sebastien Patte * [1]

[1] Université Paris-Saclay, CEA, List – Université Paris-Saclay, CEA, List, Laboratoire National Henri Becquerel (LNE-LNHB) – France

Les logiciels pour systèmes critiques ont besoin de garanties fortes de sûreté et de sécurité, un bogue pouvant avoir de lourdes conséquences dans ce contexte. C'est pourquoi nous avons besoin de méthodes de vérification puissantes, comme les méthodes formelles, pour s'assurer de l'absence de bogues. Frama-C est une plateforme open-source d'analyse formelle de code C, développée au CEA. Elle est accompagnée par ACSL, un langage de spécification formel basé sur les contrats de fonctions.

Récemment, un greffon Frama-C a été réalisé pour spécifier et vérifier des propriétés Typestates, qui restreignent l'ensemble des opérations possibles sur une structure de données, en fonction de son état actuel. Ces propriétés ne sont pas directement ramenables à des contrats de fonctions, vu que le Typestate d'un objet évolue au cours des appels. Le greffon instrumente le programme original avec du code fantôme et
des contrats ACSL, afin d'utiliser les analyseurs standard de Frama-C.

L'objectif principal de la thèse est de proposer une formalisation de cette instrumentation et de prouver sa correction : si un programme instrumenté est valide alors le programme d'origine respecte sa spécification Typestates. Dans ce but, utilisons le méta-langage Skel qui permet d'écrire des sémantiques de langages de programmation. Dans un premier temps, le sous-ensemble du langage C considéré est assez limité (un seul point de retour par fonction, pas de conditionnelles ni de boucles ...). L'instrumentation et ses langages d'entrée/sortie sont formalisés en Skel. Ensuite, nous générons une formalisation Coq grâce à l'outil Necrocoq. Actuellement, certaines preuves ont déjà été menées, par exemple sur les fonctions auxiliaires permettant de manipuler l'état mémoire, la fraîcheur des noms, l'absence de pointeurs pendouillants.

*Intervenant

Implémentation des Bigraphes dans Coq

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

[1] ISAE-SUPAERO – Institut Supérieur de l'Aéronautique et de l'Espace – France
[2] ISAE-SUPAERO – Institut Supérieur de l'Aéronautique et de l'Espace – France
[3] Laboratoire de recherche ENAC – Ecole Nationale de l'Aviation Civile - ENAC – France

Les bigraphes sont un modèle mathématique introduit par Robin Milner. Ils peuvent être utilisés pour représenter des systèmes concurrents et distribués.
Nous avons implémenté une sémantique formelle des bigraphes dans l'assitant de preuve Coq. Cet article présente l'implémentation choisie et comment instancier un bigraphe dans notre bibliothèque.

*Intervenant

Approche Dirigée par les Modèles pour la Sécurité Auto-adaptative des Systèmes Cyber-physiques

Salim Chehida * [1], Eric Rutten [2], Guillaume Giraud [3], Stéphane Mocanu [4]

[1] Inria Research Centre Grenoble Rhône-Alpes – Université de Grenoble Rhône-Alpes – France
[2] Inria Research Centre Grenoble Rhône-Alpes – Université de Grenoble Rhône-Alpes – France
[3] Réseau de Transport d'Electricité - RTE – RTE – France
[4] Laboratoire d'Informatique de Grenoble – Université Grenoble Alpes – France

Ce travail propose approche pour la sécurité auto-adaptative dans les systèmes cyber-physiques (CPS) : architecture logicielle, méthode de conception, intégration avec la prise de décision basée sur les modèles. Notre approche permet l'évaluation du risque de sécurité (SRA), en tenant compte les aspects de qualité de service (QoS). Nous formalisons le problème de décision à résoudre à chaque cycle de la boucle de contrôle d'auto-adaptation en termes de modélisation et de résolution par programmation de contraintes (CP). Nous validons notre approche en l'appliquant aux réseaux électriques intelligents, plus particulièrement à une étude de cas industrielle fournie par RTE, le gestionnaire du réseau de transport d'électricité français.

*Intervenant

Posters et Démos

Assistance à l'ingénierie de logiciels pour mieux protéger la vie privée des utilisateurs

Selena Lamari * [1,2]

[1] Laboratoire d'Informatique de Robotique et de Microélectronique de Montpellier – CNRS, Université de Montpellier – France

[2] Laboratoire de Recherche pour le Developpement des Systemes Informatises [Blida] – Algérie

Alors que les systèmes logiciels deviennent de plus en plus personnalisés pour les utilisateurs, la collecte et l'exploitation des données personnelles ont atteint des niveaux sans précédent. Cette évolution souligne un problème critique sur la protection de la vie privée des utilisateurs. Le respect de réglementations en matière de protection des données personnelles, comme le règlement général européen sur la protection des données (RGPD), est devenu une nécessité juridique. Malgré les obligations légales, il subsiste un manque de méthodes universellement acceptées pour intégrer la gestion de la vie privée dans les applications. Pour aider les développeurs, notre objectif principal est d'introduire une approche d'ingénierie logicielle outillée qui intègre des fonctionnalités de protection des données personnelles dans les applications existantes, les rendant ainsi respectueuses de la vie privée. La particularité de notre proposition réside dans son caractère actionnable, qui va au-delà d'une simple description déclarative de la régulation. Notre travail consiste à présenter les fondements de cette approche : le métamodèle PRIAM (Privacy Assessment Model) et ses artefacts associés. PRIAM modélise le RGPD en capturant les concepts décrits dans le règlement. Ce métamodèle sert de base à divers artefacts (user stories génériques, schéma de BD) qui s'adaptent à toute application spécifique qui nécessite de tenir compte du respect de la vie privée. Notre approche comprend un langage spécifique au domaine (DSL) dérivé du métamodèle PRIAM, des user stories adaptées à l'application spécifique et une base de données qui stocke à la fois les données spécifiques à l'application et à l'utilisateur nécessaires à la mise en œuvre des droits des utilisateurs en matière de respect de la vie privée.
Notre approche se veut non-intrusive sur l'application du fournisseur des traitements. Pour cela, nous avons élaboré un protocole de contrôle d'accès (gestion de l'authentification et des autorisations) et de gestion des consentements externalisé, s'appuyant sur le paradigme ABAC (Attribute-Based Access Control) avec le patron d'architecture des side-cars pour impacter au minimum l'application du fournisseur.

*Intervenant

Lightweight Syntactic API Usage Analysis with UCov

Gustave Monce * [1]

[1] Laboratoire Bordelais de Recherche en Informatique – L'Institut National de Recherche en Informatique et e n Automatique (INRIA), Université de Bordeaux (Bordeaux, France), Centre national de la recherche scientifique - CNRS (France), Bordeaux INP, G – France

Designing an effective API is essential for library developers as it is the lens through which clients will judge its usability and benefits, as well as the main friction point when the library evolves. Despite its importance, defining the boundaries of an API is a challenging task, mainly due to the diverse mechanisms provided by programming languages that have non-trivial interplays.

In this poster, we illustrate the UCov paper submitted previously at the ICPC 2024 conference. We present a novel conceptual framework designed to assist library maintainers in understanding the interactions allowed by their APIs via the use of syntactic usage models. These customizable models enable library maintainers to improve their design ahead of release, reducing friction during evolution. The complementary syntactic usage footprints and coverage scores, inferred from client code using the API (e.g., documentation samples, tests, third-party clients), enable developers to understand in-the-wild uses of their APIs and to reflect on the adequacy of their tests and documentation. We present an implementation of these models for Java libraries in the tool UCov and demonstrate its capabilities on three Java libraries exhibiting diverse styles of interaction: Jsoup, Apache commons-cli, and Spark. Our case study shows that UCov provides valuable information regarding API design and fine-grained analysis of client code to identify under-tested and under-documented library code.

*Intervenant

Un outil pour la manipulation d'hamiltoniens ... et de comptage des couplages parfaits

Mathieu Nguyen * [1,2]

[1] Laboratoire Méthodes Formelles – CNRS, L'Institut National de Recherche en Informatique et e n Automatique (INRIA), Ecole Normale Supérieure de Paris - ENS Paris, UMR 3347 CNRS, U1021 Inserm, Université Paris Saclay, Centre Universitaire, F-91405, Orsay, France., Centrale Supélec – France

[2] Institut de Recherche en Informatique Fondamentale – CNRS, Université Paris Cité – France

Nous présentons un outil capable de représenter et de manipuler, de manière graphique, des hamiltoniens qui sont notamment utilisés dans le cadre de l'informatique quantique. Le calcul à base d'hamiltonien, à l'instar de celui plus habituel des circuits quantiques, a la particularité de nécessiter une matrice de taille exponentielle en la taille du système pour le décrire, même si le système est décrit opérationnellement de façon polynomiale. La représentation graphique permet une visualisation et une manipulation de cette description opérationnelle, permettant entre autres l'optimisation avant implémentation sur une machine physique. Via un lien récemment établi entre les diagrammes ZW et le calcul de couplages parfaits dans des graphes, cet outil peut également servir au calcul de cette quantité. L'outil est entièrement développé par le premier auteur, et est basé sur un framework développé en partie par le second. Cet outil est en cours d'implémentation, et sera rendu public lorsque plus mature.

*Intervenant

Technical Infrastructure Monitoring: Modernising a 20 year system for CERN Technical Services

Thomas Georges * [1,2]

[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
[2] European Organization for Nuclear Research – Suisse

This poster presents an overview of the Technical Infrastructure Monitoring (TIM) software ecosystem at CERN and its modernisation efforts. The project focuses on upgrading a 20-year-old system that is critical for the safe operation of the Large Hadron Collider (LHC) and other essential services. The aim is to move to a modern, manageable and cost-effective infrastructure, while minimising downtime and maintaining high availability.

Efforts are focused on improving data acquisition, developing user-friendly graphical interfaces, and improving data transformation and monitoring tools. Standardisation of processes is also a priority to ensure consistency and reliability.

In the short term, the project aims to modernise DevOps practices and upgrade the infrastructure. Medium-term goals include consolidating system configurations, updating user interfaces, and migrating to a Kubernetes-based infrastructure.

By addressing the challenges of legacy technologies and meeting increasing efficiency demands, this research will ensure the continuous and secure operation of CERN's technical services, supporting the ongoing functionality and safety of its experimental physics efforts.

*Intervenant

Qontextium : estimation du degré de contextualité de configurations quantiques

Axel Muller * [1], Alain Giorgetti [2]

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

Nous présentons le logiciel Qontextium, qui permet de générer desconfigurations quantiques et d'estimer leur degré de contextualité, en utilisantun solveur SAT. Plus précisément, ces configurations quantiques sont desgéométries symplectiques finies, dont les points sont en bijection avec desvecteurs de bits de longueur paire. Le calcul du degré de contextualité seréduit à celui du nombre maximal d'équations qui peuvent être satisfaites dansun système linéaire en arithmétique binaire. L'outil est téléchargeable ici : https://quantcert.github.io/contextualityDegree

*Intervenant

Méthodologie pour maintenir l'évolution sécurisée des modèles système

Chahrazed Boudjemila * [1]

[1] Laboratoire des sciences et techniques de l'information, de la communication et de la connaissance – IMT anlantique – France

La multi-modélisation est une approche dans le domaine de l'ingénierie dirigée par les modèles (MDE) pour le développement des systèmes complexes en utilsant un ensemble de modèles hétérogènes. Ces modèles sont définis en utilisant différents langages de modélisation et construits avec divers outils. Ces modèles représentent différents aspects du système qui sont souvent dépendants.
Cependant, les modèles d'un système subissent des changements pour s'adapter à de nouvelles exigences, s'ajuster à des changements de déploiement, améliorer les fonctionnalités, et corriger des bogues . Maintenir la consistance de ces modèles hétérogènes reste un défi surtout dans le développement des systèmes critiques qui requirent de la sécurité. En effet, pour que les exigences de sécurité soient prises en compte selon le principe du "security-by-design", il faut ajouter des aspects de sécurité dans les modèles représentant le système. Ces exigences de sécurité doivent être maintenues malgré les changements apportés aux modèles.
Dans ce contexte, on aborde les deux défis suivants : 1) Comment établir des liens entre des modèles hétérogènes ? 2) Comment détecter et assurer que les changements apportés à un modèle ne causent pas d'inconsistance de sécurité avec les autres modèles représentant le système ?

Pour relever ces défis, nous proposons une méthodologie qui permet de créer et de maintenir la consistance de la sécurité des modèles hétérogènes même après leur évolution.
Cette méthodologie est divisée en deux phases : la première phase consiste à créer une fédération de sécurité dans laquelle les dépendances, relatives à la sécurité, entre les différents modèles hétérogènes sont réifiées. Des règles de sécurité exprimant les exigences de consistance en matière de sécurité entre les modèles hétérogènes sont associées aux correspondances. Ces correspondances établissent un lien entre deux éléments appartenant à deux modèles différent.
Ensuite, dans une seconde phase, nous décrivons comment utiliser cette fédération de sécurité pour signaler d'éventuelles d'inconsistance lorsque les modèles changent. Pour cela, à chaque fois qu'un modèle est modifié, les règles de sécurité associées aux correspondances impactées par la modification sont évaluées pour vérifier la consistance.

*Intervenant

TAGAda

Delphine Longuet * [1]

[1] Thales Research Technology – - – France

Cette démo présenté l'outil de génération de tests développé dans le projet TAGAda.

*Intervenant