

Haut Conseil de l'Évaluation de la Recherche et
de l'Enseignement Supérieur



DOCUMENT D'AUTOÉVALUATION
Équipe MoVe



Campagne d'évaluation 2023-2024 — Vague D

Table des matières

1	INFORMATIONS GÉNÉRALES SUR L'ÉQUIPE MOVE	3
1.1	Les thématiques scientifiques et leurs enjeux	3
	Vérification formelle : des fondements théoriques pour l'analyse de systèmes	3
	Vérification formelle : Applications et outils	4
	Ingénierie des modèles et Ingénierie des processus	5
	Modélisation de familles de systèmes : Gestion de la Variabilité logicielle	6
2	INTRODUCTION DU PORTFOLIO	7
3	AUTOÉVALUATION DU BILAN	8
3.1	Autoévaluation de l'équipe	8
	Domaine 2. Attractivité	8
	Domaine 3. Production scientifique	9
	Domaine 4. Inscription des activités de recherche dans la société	11
4	RÉFÉRENCES BIBLIOGRAPHIQUES EXTERNES	12
5	RÉFÉRENCES BIBLIOGRAPHIQUES SIGNIFICATIVES DE MOVE	13
A	ANNEXE — MEMBRES PERMANENTS AU 31/12/2022	16

1 INFORMATIONS GÉNÉRALES SUR L'ÉQUIPE MoVe

Nom de l'équipe : Modélisation et Vérification (MoVe)

Responsable de l'équipe

Tewfik Ziadi (depuis le 01/10/2020)

Béatrice Bérard et Tewfik Ziadi (jusqu'au 30/09/2020)

	2017	2018	2019	2020	2021	2022
PR	7	7	7	7	6	6
MCF HDR	5	5	6	6	6	6
MCF	10	10	9	9	9	9
DR	0	0	0	0	0	0
CR HDR	0	0	0	0	0	0
CR	0	0	0	0	0	0
Total permanents	22	22	22	22	21	21
Émérites	2	2	2	2	3	3
Doctorants	3	6	8	8	11	9
Ingénieurs CDD ou hors tutelles	1	0	0	0	0	0
Post-doc, ATER, etc.	0	2	2	0	1	0
Stagiaires	6	6	4	0	9	4
Total non permanents	10	14	14	8	21	13
Total avec émérites	34	38	38	32	45	37
Equivalent temps plein recherche	11.0	11.0	11.0	11.0	10.5	10.5

TABLE 1 – Personnels MoVe sur la période 2017-2022 (au 1er juillet de chaque année)

MoVe est la plus grosse équipe du LIP6. Cependant, il convient de signaler que, parmi nos permanents 5 sont personnels de l'Université Paris-Nanterre et 4 de l'IUT de Paris - Rives de Seine - Université Paris Cité. Cette caractéristique a un impact sur la vie de l'équipe car environ la moitié de nos membres sont localisés sur deux sites et ont souvent des responsabilités dans leur propre établissement.

1.1 Les thématiques scientifiques et leurs enjeux

L'équipe MoVe s'intéresse à la modélisation et la vérification de systèmes complexes répartis. La construction de tels systèmes nécessite des solutions pour maîtriser leur *complexité* et *diversité*. De plus, garantir leur *fiabilité* est une tâche cruciale puisque ces systèmes sont censés gérer des missions critiques. Pour maîtriser la complexité et la diversité de systèmes complexes, l'équipe MoVe s'appuie conjointement sur les fondements de *l'ingénierie de modèles* et *l'ingénierie de lignes de produits logiciels*. L'objectif est de modéliser ces systèmes selon différents aspects, allant de leur processus de développement jusqu'à la génération de code et la gestion de la variabilité. Quant aux aspects fiabilités et sûretés, ces derniers sont assurés en s'appuyant sur des fondements théoriques pour leur analyse. Ceci concerne particulièrement l'utilisation des techniques de *vérification* (comme le *Model Checking*), la *synthèse de contrôleur* et les *problèmes de satisfaisabilité booléenne*. L'équipe trouve une cohérence thématique dans la maîtrise des différentes dimensions de construction de systèmes complexes allant de la modélisation jusqu'à la vérification pour assurer leur fiabilité.

Vérification formelle : des fondements théoriques pour l'analyse de systèmes

Une partie de nos travaux des cinq dernières années a porté sur le développement de nouveaux modèles, et l'exploration de nouveaux problèmes de vérification capables de capturer des systèmes infinis, soit parce qu'ils modélisent l'écoulement quantitatif du temps, soit parce qu'ils représentent des systèmes distribués faisant interagir un nombre non borné d'agents.

Modèles temporisés et concurrents. L'équipe a participé à la proposition d'un modèle temporisé qui a été étendu aux contraintes polynomiales ([Bérard et al., 2021b]). Dans un article plus récent ([Bérard and Haddad, 2022]), nous avons proposé une optimisation de la complexité pour le problème d'accessibilité dans ce modèle. Nous avons également obtenu des résultats originaux de décidabilité pour des systèmes dynamiques avec des trajectoires très générales ([Bérard et al., 2018a]).

Systèmes paramétrés. Les travaux de l'équipe concernent également le problème de synthèse, qui consiste à dériver automatiquement un système sous la forme d'un modèle formel à partir de sa spécification donnée par une formule logique. Nous avons travaillé sur la synthèse de systèmes distribués paramétrés, c'est-à-dire qui se



comportent correctement quel que soit le nombre de processus qui le composent. Le problème étant indécidable en général, nous avons obtenu des résultats de décidabilité pour de nouveaux cas de systèmes paramétrés, que le système soit dynamique avec de nouveaux processus pouvant apparaître au cours de l'exécution ([Bollig et al., 2018]) ou non ([Sznajder et al., 2020]). Ce travail a été réalisé dans le cadre de la thèse de M. Lehaut. Enfin, notre collaboration avec l'équipe NPA se poursuit sur la vérification formelle d'algorithmes distribués pour des réseaux de robots collaboratifs (algorithmes auto-stabilisants). Nous avons proposé un modèle permettant de vérifier les premières propriétés de façon paramétrée sur ces algorithmes souvent difficiles à prouver manuellement ([Sangnier et al., 2017a, Sangnier et al., 2020]).

Sécurité. Les problèmes de sécurité ont été abordés sous l'angle des fuites d'information. Nous avons obtenu des résultats de décidabilité sur des modèles à base d'ordres partiels, souvent utilisés dans le contexte de systèmes répartis ([Bérard et al., 2017d, Bérard et al., 2018b]). Nos travaux ont également concerné l'étude spécifique de l'opacité, une propriété des systèmes qui consiste à empêcher ou à minimiser ces fuites d'informations. Dans le cadre qualitatif et pour des réseaux de Petri, nous avons mené une étude des différentes complexités pour la vérification de cette propriété ([Bérard et al., 2018c]). Nous avons obtenu des résultats pour des modèles probabilistes, qui permettent de mesurer cette opacité, c'est-à-dire la proportion de secret qui fuit d'un système ([Bérard et al., 2018d]). Nous avons étendu cette étude à des modèles plus complexes où les exécutions sont guidées par des stratégies, avec des objectifs de minimisation et de maximisation ([Bérard et al., 2017c]).

Contrôle et synthèse. Une thèse en collaboration avec IRT SystemX et le LSV (soutenue en novembre 2018) s'attaque à la vérification statistique de contrôleurs industriels pour véhicules autonomes, dont la conception est souvent réalisée avec l'outil Simulink. Pour cela, le doctorant a produit une sémantique formelle pour Simulink qu'il a ensuite implémentée dans le model-checker statistique Cosmos ([Barbot et al., 2018a]). D'un point de vue plus théorique, nous avons démarré une collaboration avec des membres du LMF sur la synthèse de systèmes distribués avec réseaux de communication dynamiques ([Bérard et al., 2020]).

Vérification formelle : Applications et outils

L'équipe MoVe s'appuie sur les fondements théoriques présentés ci-dessus pour la vérification formelle de systèmes. Un élément important de nos travaux en vérification formelle est notre investissement dans le développement de logiciels concrets permettant l'analyse et la vérification de systèmes concurrents complexes.

Solveurs SAT. Le problème de satisfaisabilité d'une formule booléenne (SAT) est un problème NP-complet très expressif mais pour lequel des stratégies permettant de résoudre de très grandes instances existent. Au cours des cinq dernières années nous avons investi en collaboration avec LRDE et l'équipe DELYS dans la plateforme générique et configurable PaInLeSS. Ce travail a été l'occasion d'impliquer 5 doctorants

En-dehors de la plateforme elle-même, nos contributions touchent les stratégies permettant de faire collaborer des solveurs dans une architecture multi-core et/ou distribuée, ainsi que l'exploitation des symétries pour réduire la complexité, ou l'étude fine des contraintes introduites afin d'optimiser les solutions. La plateforme PaInLeSS a montré ses qualités en remportant des récompenses dans la compétition internationale officielle autour des solveurs SAT.

Model-Checking Contest¹. Le développement d'outils efficaces et corrects, qui se comportent bien sur des benchmarks divers est une tâche difficile. La reproduction des expériences rapportées dans les articles comparant ou mesurant des outils est aussi un problème majeur. Notre équipe est fortement impliquée dans l'organisation de la compétition annuelle de model-checking (MCC) depuis 2011.

Outils pour le model-checking. Notre équipe a construit et continue de développer le model-checker ITS-tools, un outil permettant l'analyse du comportement (logique temporelle) de systèmes divers en appui sur un langage intermédiaire, et muni de procédures de décisions puissantes et efficaces. Au cours des cinq dernières années nous avons investi dans les développements de procédures de décision qui mêlent explorations pseudo-aléatoires, réductions structurelles évitant d'explorer des pans entiers du comportement, et des sur-approximations du système basées sur l'intégration de solveurs de contraintes généraux (comme Z3 pour le SMT). Grâce à ces progrès algorithmiques et aux efforts d'implantation, l'outil a pris depuis 2020 la tête de la compétition au MCC, et y obtient de nombreuses médailles d'or.

Outils pour l'analyse de BPMN. Notre équipe développe l'outil fBPMN, qui permet l'analyse de spécifications BPMN à l'aide d'une transformation vers une logique de premier ordre ensuite soumise à TLA+ (portfolio 6).

1. <https://mcc.lip6.fr>

Application pour les systèmes cyber-physiques.. L'équipe s'intéresse à l'utilisation des techniques d'analyse formelle pour la simulation de systèmes cyber-physiques. Ces systèmes se caractérisent non seulement par le grand nombre d'entités qui les composent, mais aussi par la présence en leur sein d'éléments embarqués temps réel en interface avec le monde physique (capteurs, actionneurs). L'équipe développe un prototype appelée BCM4Java (modèles à composants répartis) et son extension aux systèmes de contrôle cyber-physiques (BCM4Java-Cyphy) intégrant une bibliothèque de simulation en Java dédiée [Malenfant, 2018].

Une autre application pour les systèmes cyber-physiques concerne le système de guidage adaptatif et contextuel d'un agent ambiant dédié à la conduite efficace de véhicules autonomes [Chaouche et al., 2020, Ilié et al., 2022]. Cet agent embarqué de nom E-HoA (Embedded - Higher order Agent) raisonne suivant des principes symboliques d'intentionnalité dont l'expressivité est riche et intuitive. Des propriétés de consistance symbolique intentionnelle des exécutions spatio-temporelles et la capacité d'adaptation du modèle d'exécution à la volée sont démontrées. Son architecture logicielle est issue de coopérations transverses au LIP6 - MoVe et CIAN avec des chercheurs de l'Université de Constantine 2 en Algérie. En particulier, des mécanismes conjugués de caches de données et d'apprentissage permettent à l'agent de raisonner et de (ré-)agir, contextuellement et efficacement, en fonction des expériences de conduite sur le terrain.

Ingénierie des modèles et Ingénierie des processus

L'utilisation de modèles est au cœur de l'ingénierie logicielle et de l'ingénierie des processus. Les modèles supportent la conception, l'analyse et l'exécution des systèmes et applications informatiques. Ces modèles peuvent être définis de façon spécifique à un domaine ou un besoin ou avoir un cadre d'application plus général. Dans le premier cas, on se place dans ce qui concerne les modèles spécifiques au domaine (avec la définition et l'utilisation des DSL et DSML). Dans le second cas on se place souvent dans le cadre de normes et de standards internationaux tels que CMMI, UML ou BPMN.

Analyse de processus métiers.. L'équipe MoVe développe depuis plusieurs années des travaux portant sur la définition de techniques d'analyse de processus métiers définis à l'aide du standard BPMN [1]. Nous avons défini une sémantique formelle à un sous-ensemble de BPMN [Houhou et al., 2019].

Par rapport à l'existant, cette sémantique présente plusieurs avantages. Il s'agit d'une sémantique directe en logique du premier ordre (FOL), ce qui permet de cibler plusieurs cadres et outils de vérification (des expériences ont été faites avec TLA⁺, Alloy et leurs outils d'analyse), contrairement aux sémantiques de la littérature qui procèdent par traduction dans un cadre spécifique (typiquement, les réseaux de Petri). Notre sémantique prend en compte des éléments du standard BPMN dont la sémantique est reconnue complexe, car non locale (porte inclusive, interruption des sous-processus).

Enfin l'analyse des modèles de processus métiers peut se faire suivant différents modèles de la communication inter-processus (certaines propriétés peuvent être vérifiées pour un modèle de communication mais pas pour un autre). L'ensemble de ces travaux, liés à l'analyse des processus métiers BPMN, est implanté dans l'outil open-source fBPMN. Ces travaux ont ensuite servi de base à différentes extensions pour la prise en compte des aspects temporels du standard BPMN [Houhou et al., 2022, Houhou et al., 2021] ainsi que l'intégration au sein de modèles BPMN de contraintes et d'actions liées au contexte d'exécution des processus métier [Saddem-Yagoubi et al., 2021]. Au delà de l'analyse d'une version donnée d'un processus ou d'une collaboration inter-processus, nous avons aussi proposé des techniques outillées pour l'analyse de propriétés dans le cadre de l'évolution de processus métier [Krishna et al., 2019].

Recommandation de relecteurs dans les processus de développement de logiciels libres.. Dans le contexte des processus et des équipes de développement agiles, l'un des défis bien identifié consiste à trouver une allocation optimale des ressources [3, 4]. Une ressource désigne dans ce contexte des développeurs, testeurs, relecteurs de code, administrateurs ou managers. Dans une première étape nous nous sommes intéressés au moyen d'identifier et d'augmenter le nombre de relecteurs dans les projets de logiciels libres. Ce manque a déjà été reporté par plusieurs communautés [6]. Nous l'avons également confirmé en menant une étude sur l'historique de développement de 5 projets de logiciels libres i.e. , Kafka, Beam, Flink, Spark and Zookeeper, ce qui représentait plus de 58000 pull-request, 80000 commits et plus de 106000 fichiers. L'étude démontre clairement un manque flagrant de relecteurs et le fait que ce sont toujours les mêmes personnes qui effectuent les relectures. Dans le contexte de gros projets de logiciels libres, cela peut être très bloquant.

Pour pallier ce problème, nous avons conçu un système de recommandations basé sur la stratégie de filtrage collaboratif (CF) [5], et plus précisément, sur la factorisation de matrice. Pour une modification de code donnée, le système recherche d'abord les relecteurs pertinents sur la base de l'historique des révisions. Notre approche identifie ensuite comme nouveaux relecteurs possibles les développeurs dont l'historique de développement est

similaire à celui des relecteurs trouvés. La principale avancée de notre système par rapport aux travaux existants est sa capacité à recommander à la fois des relecteurs réguliers et de nouveaux relecteurs potentiels sur la base de leur historique de révisions et de livraisons.

Pour évaluer notre approche, nous avons utilisé deux ensembles de données dont la fréquence des changements de code, le nombre de commits, de fichiers et de contributeurs diffèrent. Le premier ensemble de données se compose de cinq projets hébergés sur GitHub : Beam, Flink, Kafka, Spark et Zookeeper. Le deuxième ensemble de données est composé de quatre communautés de logiciels qui hébergent leurs révisions sur Gerrit : Android, Openstack, QT et Libreoffice. Android est la plus grande des communautés sélectionnées avec plus de 3 000 000 de commits à travers les 112 sous-projets. Dans tous les cas, notre système de recommandation fonctionne avec succès, identifiant les relecteurs pertinents parmi l'ensemble des relecteurs réels et les relecteurs potentiels parmi les développeurs [Chueshev et al., 2020]. Ces travaux ont été réalisés en collaboration avec Julia Lawall, Directrice de Recherche au sein de l'équipe INRIA Whisper.

Modélisation de familles de systèmes : Gestion de la Variabilité logicielle

MoVe s'intéresse à la modélisation et à la construction de familles de systèmes en utilisant les fondements de l'ingénierie de lignes de produits logiciels [2]. Une ligne de produits logiciels est définie comme une famille de systèmes appartenant au même domaine et qui sont caractérisés par des éléments communs mais aussi par des points de variation. La gestion de cette variabilité est le principal défi scientifique pour la construction de lignes de produits. Plusieurs contributions ont été proposées par l'équipe pour l'inférence de la variabilité et l'optimisation de l'espace de configuration.

Inférence de la variabilité logicielle.. MoVe développe des méthodes pour l'inférence de la variabilité logicielle à partir d'artefacts logiciels similaires existants. Les contributions de l'équipe dans ce contexte sont intégrées dans le logiciel libre BUT4Reuse² qui a été développé comme un cadre générique et extensible pour supporter plusieurs types d'artefacts logiciels (le code source, les modèles, etc.).

Durant les 5 dernières années, nous avons étendu BUT4Reuse avec de nouvelles méthodes pour l'identification de services réutilisables à partir de code source de familles de systèmes [Shatnawi et al., 2020]. Nous avons aussi intégré à BUT4Reuse de nouveaux adaptateurs pour l'inférence de la variabilité dans les applications mobiles. Cela a abouti à une étude empirique sur 70 familles d'applications [Shatnawi et al., 2019]. Une nouvelle extension à BUT4Reuse a été proposée pour l'inférence de la variabilité à partir de familles d'architectures logicielles [Kerdoudi et al., 2019b, Kerdoudi et al., 2022]. Dans le cadre d'une collaboration avec des collègues de l'Université de San Jorge en Espagne, nous avons proposé une méthode pour inférer la variabilité à partir de modèles. La méthode s'appuie sur des concepts de l'ingénierie dirigée par les modèles et les algorithmes génétiques. La méthode a été publiée dans la revue IEEE TSE [Pérez et al., 2022]. Pour permettre l'évaluation systématique des techniques d'inférence de la variabilité, nous avons proposé deux benchmarks. Le premier, EFLBench, repose sur les distributions d'Eclipse [Martinez et al., 2018d] et le second, ArgoUML-SPL, s'appuie sur l'outil libre ArgoUML [Martinez et al., 2018a]. Les deux benchmarks sont diffusés à la communauté scientifique du domaine. Le benchmark ArgoUML-SPL est utilisé depuis plusieurs années comme challenge dans le contexte de la conférence SPLC³.

Analyse et optimisation de modèles de variabilité.. Dans le cadre d'une thèse CIFRE avec Renault, nous nous sommes intéressés à l'analyse de l'espace de configurations du modèle de variabilité des véhicules de Renault. Nous avons proposé d'exploiter les symétries, qui représentent des parties isomorphes dans l'espace de configurations, pour optimiser le système de gestion de variabilité de Renault. Les premiers résultats ont été publiés dans la conférence SPLC 2020. [Xu et al., 2021].

2. <https://but4reuse.github.io/>

3. <https://variability-challenges.github.io/index.html>

2 INTRODUCTION DU PORTFOLIO

Nous avons choisi à travers les éléments de portfolio des éléments permettant de montrer la diversité des thèmes de recherche de l'équipe.

Chaque élément disposant d'une fiche explicative détaillée, nous listons simplement ici ces éléments :

1. **Élément 1 (publication)** : article intitulé "Approaches to Co-Evolution of Metamodels and Models : A Survey", par Regina Hebig ; Djamel Eddine Khelladi ; Reda Bendraou. Publication dans la revue IEEE Trans Software Eng 2017.
2. **Élément 2 (Jeu de données consolidées)** : Le Model Checking Contest.
3. **Élément 3 (logiciel ou bibliothèque logicielle)** : Outil de verification formelle ITS-Tools.
4. **Élément 4 (publication)** : article intitulé "A novel approach for Software Architecture Product Line Engineering", par Mohamed Lamine Kerdoudi , Tewfik Ziadi, Chouki Tibermacine, Salah Sadou. Publication dans la revue Journal of Systems and Software, vol. 186, pp. 111191, (Elsevier).
5. **Élément 5 (publication)** : article intitulé "Parameterized Synthesis for Fragments of First-Order Logic over DataWords", par Béatrice Bérard, Benedikt Bollig, Mathieu Lehaut, Nathalie Sznajder. Publication dans les actes de FOSSACS 20.
6. **Élément 6 (publication)** : article intitulé "A First-Order Logic Semantics for Communication-Parametric BPMN Collaborations", par Sara Houhou, Souheib Baair, Pascal Poizat, Philippe Quéinnec. Publication à la 17ème International Conference on Business Process Management (BPM 2019).
7. **Élément 7 (logiciel ou bibliothèque logicielle)** : bibliothèque de développement automatique de SAT solveurs parallèles Painless.

3 AUTOÉVALUATION DU BILAN

3.1 Autoévaluation de l'équipe

Domaine 2. Attractivité

Référence 1. L'unité est attractive par son rayonnement scientifique et s'insère dans l'espace européen de la recherche.

Organisation de conférences internationale. L'équipe a organisé la 23e édition de la conférence ACM SPLC 2019 (Systems and Software Product Line Conference) avec plus de 150 participants. Il s'agit de la conférence principale dans le domaine de l'ingénierie de lignes de produits logiciels. Un membre de l'équipe était co-general chair de cette édition. Un autre membre était co-general chair des conférences ICSSP 2021 (International Conference on Software and System Processes) et ACM/IEEE ICGSE 2021 (International Conference on Global Software Engineering). L'équipe a organisé aussi deux éditions de la conférence Petri Nets (2020, 2021). Un membre de l'équipe était le co-general chair de ces deux éditions. En 2018, l'équipe a également organisé la journée nationale du GT IDM du GDR GPL.

Participation à des comités de programme de conférences internationales. Les membres de l'équipe sont invités régulièrement au comité de programme de conférences internationales de référence dans le domaine de la vérification formelle de systèmes (ex. ; Petri Nets, FORMATS), de l'ingénierie de logiciels et des processus (ex. ; ASE, ICSSP, ACM SAC-SE, BPM ICSOC, et ICECCS), de l'ingénierie des lignes de produits logiciels (ex. ; SPLC, ICSR). Ils participent aussi comme éditeurs associés dans des revues internationales comme Science of Computer Programming.

Participation à des comités de pilotage de conférences internationales. Les membres de l'équipe MoVe sont membres de 5 comités de pilotage de conférences internationales : SPLC, ICSSP, ETAPS, Petri Nets, et ICAS.

Prix et invitations. Plusieurs contributions de l'équipe ont été primées par des prix. Les outils ITS-Tools et PalnLeSS sont régulièrement récompensés dans les compétitions autour respectivement de la vérification (le Model Checking Contest) et des solveurs SAT (International SAT Competition).

ITS-Tools a gagné au total plusieurs médailles depuis 2017 dont 8 médailles d'or dans les différentes catégories de model checking (Accessibilité, limites supérieures, et propriétés globales, etc.). Les solveurs SAT mis en œuvre à l'aide de PalnLeSS ont obtenu d'excellents résultats lors des récentes compétitions SAT dont la première place dans le parallel track en 2018, 2021 et 2022. L'article concernant la vérification de modèles de processus a eu le prix du meilleur article de la conférence BPM 2020. L'article proposant l'exécution des modèles de processus [Bendraou et al., 2007a] a été sélectionné en 2017 comme l'article le plus influent depuis 10 ans dans la conférence internationale APSEC. Les membres de l'équipe sont régulièrement invités à des exposés et à des tutoriels associés à des conférences et workshops dont ICECCS 2019, TOOLS 2019, ICSR 2020, et MOROVER 2017.

Logiciels. L'équipe a une visibilité concernant la diffusion de logiciels libres. Ce volet est détaillé à la référence 4.

Coopérations internationales. Les membres de l'équipe collaborent avec des collègues à l'international : en Espagne, en Allemagne, en Suède, au Canada, et en Algérie. Ces collaborations se traduisent par des invitations et des visites. L'équipe a pu bénéficier, par exemple, à deux reprises de la visite en tant que professeur invité de Vijay Ganesh de l'université de Waterloo. Francisca Pérez, professeure à l'université San Jorge, a passé un séjour de deux mois au sein de l'équipe en 2019.

Responsabilités niveau recherche. L'équipe participe au pilotage national et local de la recherche. Un membre de l'équipe est président du jury du prix de thèse du GDR GPL (depuis 2021). Un autre membre était co-animateur de l'action transversale ALROB des GRD GPL et Robotique (jusqu'à 2019). Un membre de l'équipe est directeur du LIP6 (depuis 2019). Une autre membre est co-animatrice de l'axe "Théorie et outils mathématiques pour l'informatique" du LIP6. Des membres de l'équipe participent à la commission de soutien à la publication du LIP6 (depuis 2014). Au niveau international, un membre de l'équipe est secrétaire général de l'association ISSPA (International Software and Systems Process Association) qui s'intéresse particulièrement aux processus en tant que domaine de recherche, de développement, d'application et d'amélioration.

Référence 2. L'unité est attractive par la qualité de sa politique d'accompagnement des personnels.

Pour les doctorants en thèse CIFRE, l'équipe recommande une journée spécifique de présence par semaine afin de préserver un moment permettant les échanges au sein de l'équipe. Les séminaires d'équipes sont souvent organisés le jour de présence des doctorants CIFRE.

L'équipe a radicalement changé la gestion de son budget en la confiant à une commission interne. L'objectif était de créer une gestion collégiale et d'impliquer plus les collègues avec un esprit bienveillant. L'équipe MoVe continue d'accueillir ses anciens membres permanents comme émérites en leur fournissant toutes les ressources nécessaires (bureaux, équipement, .etc.).

Référence 3. L'unité est attractive par la reconnaissance de ses succès à des appels à projets compétitifs.

Participation aux projets compétitifs. L'équipe a participé au montage d'un grand projet européen ITEA3 RE-VaMP2 (2017-2020). Le projet a regroupé 29 partenaires dans 5 pays. Le projet s'intéresse aux problématiques d'extraction de lignes de produits à partir d'artefacts logiciels existants. En plus du montage, l'équipe a eu un rôle majeur comme coordinateur scientifique. Dans le cadre d'appel ANR, l'équipe MoVe participe aux deux projets : 1) PARDI (2016-2021) portant sur la vérification de systèmes distribués paramétrés ; 2) SAPPORO (2020-2024) dont l'objectif est d'établir un cadre formel pour la vérification de protocoles distribués pour les réseaux de robots mobiles, qui coopèrent pour exécuter des tâches complexes. Les budgets de ces projets sont essentiellement consacrés au recrutement de doctorants et de pos-doctorants de l'équipe avec une partie utilisée pour le financement de missions. Le projet REVaMP2 a financé 3 post-doctorants et PARDI a financé une thèse.

Financement industriels. Les thèses CIFRE avec Renault, RedFabriQ et SAP France sont une autre source importante de financement de thèses. D'autres financements de la part des organismes de recherche comme le CEA et IRT SystemX ont aussi permis le financement de deux thèses (thèses de Emma Effa et Huang Yining).

Référence 4. L'unité est attractive par la qualité de ses équipements et de ses compétences techniques.

Les compétences techniques de l'équipe ont permis d'accroître sa visibilité à travers le développement de plusieurs logiciels qui sont diffusés à l'extérieur. Comme indiqué au domaine 3, référence 1, l'équipe développe et maintient de nombreux logiciels comme ITS-Tools (portfolio 3). En plus, le Model Checking Contest (portfolio 2), à travers son jeu de données consolidées, démontre aussi les compétences techniques de l'équipe pour implémenter et maintenir une infrastructure logicielle complexe.

Domaine 3. Production scientifique

	2017	2018	2019	2020	2021	2022
Articles (revues)	0.72	0.45	0.45	0.63	0.57	0.57
Communications (conférences)	2.00	1.90	1.63	1.18	0.47	0.57

TABLE 2 – Publications par ETPR par an entre 2017 et 2022

Référence 1. La production scientifique de l'unité satisfait à des critères de qualité.

Fondements scientifiques des résultats. Les thèmes de recherche autour de la vérification de systèmes se basent sur des approches formelles qui montrent le bien fondé des résultats. D'autres résultats de l'équipe font l'objet d'une validation expérimentale importante et reproductible. C'est par exemple le cas pour les travaux sur l'inférence de variabilité avec l'outil BUT4Reuse. L'article [Shatnawi et al., 2019] montre une validation expérimentale sur le code source de 70 familles d'application mobiles. Comme nous l'avons déjà mentionné, l'équipe a participé à l'élaboration de deux benchmarks [Martinez et al., 2018a], [Martinez et al., 2018d] qui sont aussi utilisés pour la validation expérimentale de nos résultats.

Publications. L'équipe MoVe publie ses résultats dans des conférences et revues de référence : les conférences PETRI NET, FoSSaCS, et TACAS pour la vérification formelle, les conférences SPLC et ICSR pour les travaux autour de la variabilité logicielle, et la conférence BPM pour l'ingénierie des processus. L'équipe porte une grande attention à la qualité des conférences et de revues visées. Les travaux de l'équipe dans les aspects théoriques de

MoVe, Évolution des publications (2017–2022)

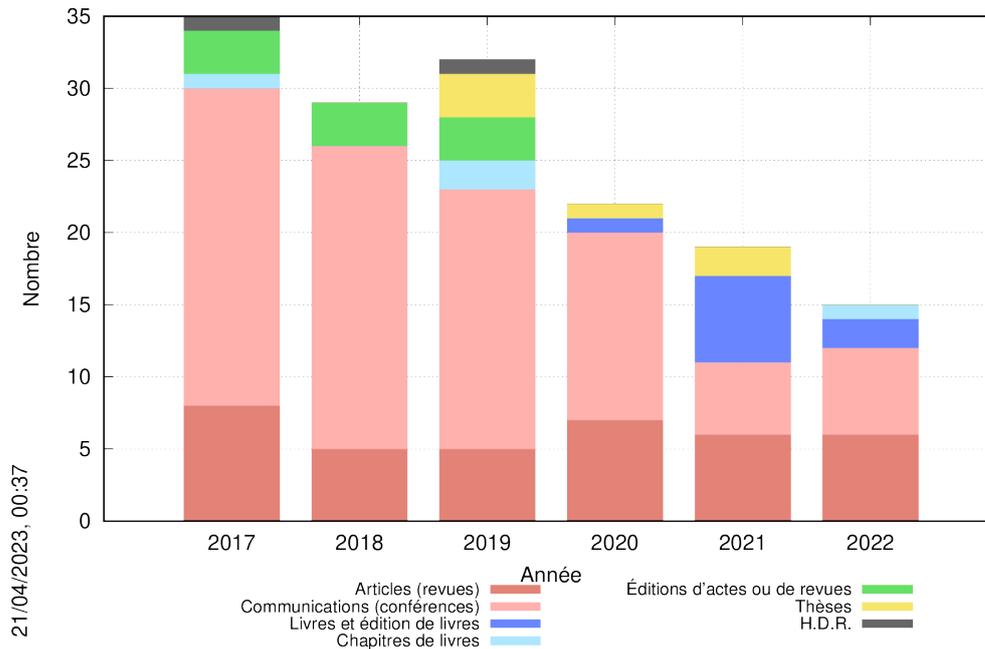


FIGURE 1 – Évolution des publications entre 2017 et 2022

la vérification (portfolio 5) sont publiés dans la FoSSaCS (une conférence classée A). Nous avons publié nos travaux concernant la modélisation des processus (portfolio 1) [Hebig et al., 2017] et la gestion de la variabilité [Pérez et al., 2022] dans la revue majeure IEEE TSE du génie logiciel (une revue classée A*). Plusieurs de nos publications impliquent régulièrement des co-auteurs d'institutions reconnues en France et à l'étranger. Les résultats sur l'extension de BUT4Reuse pour les architectures logicielles (portfolio 4) illustrent ce type de collaboration avec des collègues du LIRMM, du ENSIBS et de l'université de Biskra (Algérie).

Productions de Logiciels. Comme indiqué dans la section *Thématiques scientifiques et leurs enjeux*, plusieurs logiciels sont développés par l'équipe pour le support de la recherche : ITS-Tools, fBPMN, et PalnLeSS, et BUT4Reuse.

Ces outils sont diffusés avec leurs documentations et leurs benchmarks sur Github. Comme cela a déjà été mentionné :

- ▶ les outils ITS-Tools et PalnLeSS ont été primés dans les compétitions de *model checking* et des solveurs SAT (track parallèle),
- ▶ l'outil BUT4Reuse a été exploité dans le cadre du projet européen REVaMP2 par des partenaires industriels [Sadovykh et al., 2017, Grüner et al., 2019].

En plus d'accroître la visibilité de l'équipe, les logiciels développés par l'équipe ont permis le montage de nombreuses collaborations avec d'autres équipes du LIP6 (ex. [Vallade et al., 2020a] avec DELYS) mais aussi des collaborations avec d'autres laboratoires (ex. IRISA [Pinchinat et al., 2019] et LIRMM [Kerdoudi et al., 2022]).

Référence 2. La production scientifique de l'unité est proportionnée à son potentiel de recherche et correctement répartie entre ses personnels.

La production d'articles est en baisse ces dernières années mais les réalisations logicielles représentent une force de l'équipe et contribuent à son rayonnement (domaine 3, référence 1). La baisse de production pour la période 2020-2022 est liée à plusieurs facteurs. Comme cela a déjà été mentionné, l'équipe a réussi à augmenter le nombre de doctorants ces dernières années. Cependant, 6 des nouveaux doctorants n'ont été recrutés qu'en 2021 et 2022. Il est donc trop tôt pour observer l'impact de ces nouveaux recrutements sur la production scientifique. Un autre facteur concerne le départ en 2020 à la retraite d'une membre de l'équipe et la prise de la responsabilité de directeur du laboratoire d'un autre membre. Aussi, la crise sanitaire du COVID a impacté considérablement la production scientifique de MoVe.

Référence 3. La production scientifique de l'unité respecte les principes de l'intégrité scientifique, de l'éthique et de la science ouverte. Elle est conforme aux directives applicables dans ce domaine.

L'équipe Move est fortement impliquée dans la démarche "open science" prônée par les tutelles du laboratoire. Sa participation se fait au travers des nombreux logiciels qu'elle développe et offre à la communauté. Elle se traduit également par la mise à disposition, l'enrichissement et la maintenance d'un jeu de données permettant de "benchmarker" les outils de model checking.

Tout cela démontre que nous sommes actifs sur le volet de la reproductibilité de nos résultats de recherche.

Domaine 4. Inscription des activités de recherche dans la société

Référence 1. L'unité se distingue par la qualité et la quantité de ses interactions avec le monde non-académique.

Comme présenté ci-dessus, l'équipe participe à plusieurs projets avec des partenaires industriels. Trois thèses en cours sont réalisées dans le monde de l'industrie dans le cadre de financement CIFRE.

Référence 2. L'unité développe des produits à destination du monde culturel, économique et social.

Les membres de l'équipe sont fortement impliqués en termes de responsabilités pédagogiques à Sorbonne Université, mais aussi à l'université Paris Cité et à Université Paris Nanterre. A Sorbonne Université, une membre de l'équipe est responsable adjointe de la licence informatique. Une autre membre est co-responsable du parcours SAR du Master d'Informatique (M1–M2). En plus de son rôle comme membre du conseil d'administration de Sorbonne Université, un membre de l'équipe est responsable du service général de la formation initiale. À l'université de Paris Cité, un membre de l'équipe est directeur des études de l'année spéciale du BUT Informatique en formation initiale et continue. À l'université Paris Nanterre, un membre de l'équipe est directeur adjoint du département informatique. Un autre membre est directeur adjoint de l'UFR et membre du CFVU.

Référence 3. L'unité partage ses connaissances avec le grand public et intervient dans des débats de société.

L'équipe participe régulièrement aux éditions de la Fête de la Science organisées par Sorbonne Université pour la vulgarisation des sciences. Par exemple, un membre de l'équipe a présenté en 2022 un stand autour de la programmation sur Smartphone⁴.

4. <https://psl.eu/agenda/fete-de-la-science-lespgg-0>

4 RÉFÉRENCES BIBLIOGRAPHIQUES EXTERNES

- [1] Business Process Model and Notation, version 2.0.2. Technical report, OMG, January 2014. <http://www.omg.org/spec/BPMN/2.0.2/>.
- [2] Sven Apel, Don S. Batory, Christian Kästner, and Gunter Saake. *Feature-Oriented Software Product Lines - Concepts and Implementation*. Springer, 2013.
- [3] Oleksii Kononenko, Olga Baysal, and Michael W. Godfrey. Code review quality : How developers see it. In *2016 IEEE/ACM 38th International Conference on Software Engineering (ICSE)*, pages 1028–1038, 2016.
- [4] Ehsan Mirsaeedi and Peter C. Rigby. Mitigating turnover with code review recommendation : Balancing expertise, workload, and knowledge distribution. In *2020 IEEE/ACM 42nd International Conference on Software Engineering (ICSE)*, pages 1183–1195, 2020.
- [5] Paul Resnick and Hal R. Varian. Recommender systems. *Communication ACM*, 40 :56–58, March 1997.
- [6] Motahareh Bahrami Zanjani, Huzefa Kagdi, and Christian Bird. Automatically recommending peer reviewers in modern code review. *IEEE Transactions on Software Engineering*, 42(6) :530–543, 2016.

5 RÉFÉRENCES BIBLIOGRAPHIQUES SIGNIFICATIVES DE MoVe

- [Barbot et al., 2018a] Barbot, B., Bérard, B., Duploux, Y., and Haddad, S. (2018a). Integrating Simulink Models into the Model Checker Cosmos. In *39th International Conference on Applications and Theory of Petri Nets and Concurrency*, volume 10877 of *Lecture Notes in Computer Sciences*, pages 363–373, Bratislava, Slovakia. Springer.
- [Bendraou et al., 2007a] Bendraou, R., Combemale, B., Crégut, X., and Gervais, M.-P. (2007a). Definition of an eXecutable SPEM 2.0. In *14th Asia-Pacific Software Engineering Conference (APSEC)*, pages 390–397, Nagoya, Japan. IEEE Computer Society.
- [Bérard et al., 2020] Bérard, B., Bollig, B., Bouyer, P., Függer, M., and Sznajder, N. (2020). Synthesis in Presence of Dynamic Links. In *GandALF'20 - 11th International Symposium on Games, Automata, Logics, and Formal Verification*, Brussels (online), Belgium. 27 pages, 5 figures.
- [Bérard et al., 2018a] Bérard, B., Bouyer, P., and Jugé, V. (2018a). Finite Bisimulations for Dynamical Systems with Overlapping Trajectories. In Ghica, D. and Jung, A., editors, *27th EACSL Annual Conference on Computer Science Logic (CSL 2018)*, volume 119, pages 26 :1–26 :17, Birmingham, United Kingdom. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.
- [Bérard et al., 2018b] Bérard, B., Haar, S., and Héluët, L. (2018b). Hyper Partial Order Logic. In *FSTTCS 2018 - Foundations of Software Technology and Theoretical Computer Science*, volume 122 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 20 :1–20 :21, Ahmedabad, India. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.
- [Bérard et al., 2018c] Bérard, B., Haar, S., Schmitz, S., and Schwoon, S. (2018c). The Complexity of Diagnosability and Opacity Verification for Petri Nets. *Fundamenta Informaticae*, 161(4) :317–349.
- [Bérard and Haddad, 2022] Bérard, B. and Haddad, S. (2022). Revisiting Reachability in Polynomial Interrupt Timed Automata. *Information Processing Letters*, 174 :106208.
- [Bérard et al., 2017c] Bérard, B., Haddad, S., and Lefauchaux, E. (2017c). Probabilistic Disclosure : Maximisation vs. Minimisation. In *FSTTCS 2017*, pages 13 :1–13 :14, Kanpur, India.
- [Bérard et al., 2021b] Bérard, B., Haddad, S., Picaronny, C., Safey El Din, M., and Sassolas, M. (2021b). Polynomial interrupt timed automata : Verification and expressiveness. *Information and Computation*, 277 :104580.
- [Bérard et al., 2017d] Bérard, B., Héluët, L., and Mullins, J. (2017d). Non-interference in partial order models. *ACM Transactions on Embedded Computing Systems (TECS)*, 16(2) :44 :1–44 :34.
- [Bérard et al., 2018d] Bérard, B., Kouchnarenko, O., Mullins, J., and Sassolas, M. (2018d). Opacity for linear constraint Markov chains. *Discrete Event Dynamic Systems*, 28(1) :83–108.
- [Bollig et al., 2018] Bollig, B., Lehaut, M., and Sznajder, N. (2018). Round-Bounded Control of Parameterized Systems. In Lahiri, S. K. and Wang, C., editors, *16th International Symposium on Automated Technology for Verification and Analysis (ATVA 2018)*, volume 11138 of *Proceedings of the 16th International Symposium on Automated Technology for Verification and Analysis (ATVA 2018)*, pages 370–386, Los Angeles, California, United States. Springer.
- [Chaouche et al., 2020] Chaouche, A.-C., Ilie, J.-M., and Pêcheux, F. (2020). Dealing with Failures for Execution Consistency in Context-aware Systems. *Procedia Computer Science*, 177 :212 – 219.
- [Chueshev et al., 2020] Chueshev, A., Lawall, J., Bendraou, R., and Ziadi, T. (2020). Expanding the Number of Reviewers in Open-Source Projects by Recommending Appropriate Developers. In *ICSME 2020 - International Conference on Software Maintenance and Evolution*, Adélaïde / Virtual, Australia.
- [Grüner et al., 2019] Grüner, S., Burger, A., Abukwaik, H., El-Sharkawy, S., Schmid, K., Ziadi, T., Paule, A., Suda, F., and Viehl, A. (2019). Demonstration of a Toolchain for Feature Extraction, Analysis and Visualization on an Industrial Case Study. In *2019 IEEE 17th International Conference on Industrial Informatics (INDIN)*, pages 459–465, Helsinki, Finland. IEEE.
- [Hebig et al., 2017] Hebig, R., Khelladi, D. E., and Bendraou, R. (2017). Approaches to Co-Evolution of Metamodels and Models : A Survey. *IEEE Transactions on Software Engineering*, 43(5) :396–414.
- [Houhou et al., 2019] Houhou, S., Baarir, S., Poizat, P., and Quéinnec, P. (2019). A First-Order Logic Semantics for Communication-Parametric BPMN Collaborations. In *International Conference on Business Process Management (BPM 2019)*, volume 11675 of *Lecture Notes in Computer Science book series (LNCS)*, pages 52–68, Vienna, Austria.

- [Houhou et al., 2021] Houhou, S., Baair, S., Poizat, P., and Quéinnec, P. (2021). A Direct Formal Semantics for BPMN Time-Related Constructs. In *ENASE 2021 - 16th International Conference on Evaluation of Novel Approaches to Software Engineering*, pages 138–149, online, Czech Republic.
- [Houhou et al., 2022] Houhou, S., Baair, S., Poizat, P., Quéinnec, P., and Kahloud, L. (2022). A First-Order Logic Verification Framework for Communication-Parametric and Time-Aware BPMN Collaborations. *Information Systems*, 104 :101765.
- [Ilié et al., 2022] Ilié, J.-M., Chaouche, A.-C., and Pêcheux, F. (2022). A Reinforcement Learning Integrating Distributed Caches for Contextual Road Navigation. *International Journal of Ambient Computing and Intelligence*, 13(1) :1–19.
- [Kerdoudi et al., 2019b] Kerdoudi, M. L., Ziadi, T., Tibermacine, C., and Sadou, S. (2019b). Recovering Software Architecture Product Lines. In *ICECCS 2019 - 24th International Conference on Engineering of Complex Computer Systems*, pages 226–235, Nansha, Guangzhou, China. IEEE.
- [Kerdoudi et al., 2022] Kerdoudi, M. L., Ziadi, T., Tibermacine, C., and Sadou, S. (2022). A novel approach for Software Architecture Product Line Engineering. *Journal of Systems and Software*, 186 :111191.
- [Krishna et al., 2019] Krishna, A., Poizat, P., and Salaün, G. (2019). Checking Business Process Evolution. *Science of Computer Programming*, 170 :1–26.
- [Malenfant, 2018] Malenfant, J. (2018). Towards a well-founded software component model for cyber-physical control systems. In *Second IEEE International Conference on Robotic Computing*, Laguna Hills, California, United States.
- [Martinez et al., 2018a] Martinez, J., Ordoñez, N., Těrnava, X., Ziadi, T., Aponte, J., Figueiredo, E., and Valente, M. T. (2018a). Feature Location Benchmark with ArgoUML SPL. In *Systems and Software Product Line Conference (SPLC)*, Gothenburg, Sweden.
- [Martinez et al., 2018d] Martinez, J. F., Ziadi, T., Papadakis, M., Bissyandé, T. F., Klein, J., and Le Traon, Y. (2018d). Feature location benchmark for extractive software product line adoption research using realistic and synthetic Eclipse variants. *Information and Software Technology*.
- [Pérez et al., 2022] Pérez, F., Ziadi, T., and Cetina, C. (2022). Utilizing Automatic Query Reformulations as Genetic Operations to Improve Feature Location in Software Models. *IEEE Transactions on Software Engineering*, 48(2) :713 – 731.
- [Pinchinat et al., 2019] Pinchinat, S., Fila, B., Wacheux, F. F., and Thierry-Mieg, Y. (2019). Attack Trees : A Notion of Missing Attacks. In *GrAMSec 2019 - 6th International Workshop on Graphical Models for Security*, volume 11720 of *Lecture Notes in Computer Science*, pages 23–49, Hoboken, NJ, United States.
- [Saddem-Yagoubi et al., 2021] Saddem-Yagoubi, R., Poizat, P., and Houhou, S. (2021). Business Processes Meet Spatial Concerns : the sBPMN Verification Framework. In *FM 2021 - 24th International Symposium on Formal Methods*, volume 13047 of *Lecture Notes in Computer Science*, pages 218–234, Beijing, China.
- [Sadovykh et al., 2017] Sadovykh, A., Bagnato, A., Robin, J., Viehl, A., Ziadi, T., and Martinez, J. (2017). RE-VAMP : Challenges and innovation roadmap for variability management in round-trip engineering of software-intensive systems. *Génie logiciel : le magazine de l'ingénierie du logiciel et des systèmes*, (120) :32–36.
- [Sangnier et al., 2017a] Sangnier, A., Sznajder, N., Potop-Butucaru, M., and Tixeuil, S. (2017a). Parameterized verification of algorithms for oblivious robots on a ring. In Stewart, D. and Weissenbacher, G., editors, *Formal Methods in Computer Aided Design*, pages 212–219, Vienna, Austria. IEEE.
- [Sangnier et al., 2020] Sangnier, A., Sznajder, N., Potop-Butucaru, M., and Tixeuil, S. (2020). Parameterized verification of algorithms for oblivious robots on a ring. *Formal Methods in System Design*, 56 :55–89.
- [Shatnawi et al., 2020] Shatnawi, A., Seriai, A.-D., Sahraoui, H., Ziadi, T., and Seriai, A. (2020). ReSlide : Reusable Service Identification from Software Families. *Journal of Systems and Software*, 170 :110748.
- [Shatnawi et al., 2019] Shatnawi, A., Ziadi, T., and Mohamadi, M. Y. (2019). Understanding Source Code Variability in Cloned Android Families : an Empirical Study on 75 Families. In *26th Asia-Pacific Software Engineering Conference (ASPEC 2019)*, Putrajaya, Malaysia.
- [Sznajder et al., 2020] Sznajder, N., Berard, B., Bollig, B., and Lehaut, M. (2020). Parameterized Synthesis for Fragments of First-Order Logic over Data Words. In *23rd International Conference on Foundations of Software Science and Computation Structures (FoSSaCS'20)*, volume 12077 of *Lecture Notes in Computer Science*, pages 97–118, Dublin, Ireland. Springer.

- 
- [Vallade et al., 2020a] Vallade, V., Le Frioux, L., Baair, S., Sopena, J., Ganesh, V., and Kordon, F. (2020a). Community and LBD-Based Clause Sharing Policy for Parallel SAT Solving. In *SAT 2020 - 23rd International Conference on Theory and Applications of Satisfiability Testing*, volume 12178 of *Lecture Notes in Computer Science*, pages 11–27, Alghero / Virtual, Italy. Due to the coronavirus COVID-19 pandemic, the conference was held virtually.
- [Xu et al., 2021] Xu, H., Baair, S., Ziadi, T., Hillah, L. M., Essodaigui, S., and Bossu, Y. (2021). Optimisation for the product configuration system of Renault : towards an integration of symmetries. In *25th ACM International Systems and Software Product Line Conference - Volume B*, volume B of *SPLC '21 : Proceedings of the 25th ACM International Systems and Software Product Line Conference*, pages 86–90, Leicester, United Kingdom. ACM.

A ANNEXE — MEMBRES PERMANENTS AU 31/12/2022

La table ci dessous liste les membres permanents de l'équipe MoVe.

NOM	Prénom	Corps	Employeur
BENDRAOU	Reda	PR	Université paris-Nanterre
BESSE	Cédric	MCF	Sorbonne Université
DUTHEILLET	Claude	MCF	Sorbonne Université
GERVAIS	Marie-Pierre	PR	Université paris-Nanterre
ILIE	Jean-Michel	MCF	Université Paris-Cité
JAUME	Mathieu	MCF (HDR)	Sorbonne Université
KORDON	Fabrice	PR	Sorbonne Université
LEGOND-AUBRY	Fabrice	MCF	Université paris-Nanterre
MALENFANT	Jacques	PR	Sorbonne Université
MOUNIER	Isabelle	MCF	Sorbonne Université
PAVIOT-ADET	Emmanuel	MCF	Université Paris-Cité
POITRENAUD	Denis	MCF	Université Paris-Cité
POIZAT	Pascal	PR	Université paris-Nanterre
PRADAT-PEYRE	Jean-François	PR	Université paris-Nanterre
SAINTE-JAMES	Emmanuel	MCF (HDR)	Sorbonne Université
SZNAJDER	Nathalie	MCF	Sorbonne Université
THIERRY-MIEG	Yann	MCF (HDR)	Sorbonne Université
ZIADI	Tewfik	MCF (HDR)	Sorbonne Université
ZIANE	Mikal	MCF (HDR)	Université Paris-Cité

ÉLÉMENT DE PORTFOLIO 01



Publication

1 DÉFINITION DE CET ÉLÉMENT

Titre de l'élément : Approaches to Co-Evolution of Metamodels and Models : A Survey. Publié dans la revue IEEE Trans. Software Eng. 43(5) : 396-414 (2017)

URL de l'élément : <https://hal.sorbonne-universite.fr/hal-01525676>

2 MOTIVATIONS DU CHOIX DE CET ÉLÉMENT

Notre choix s'est porté sur cet article du fait du travail conséquent qui a été mené pour cette étude de l'état de l'art sur la co-évolution des méta-modèles et modèles. Au-delà de l'étude de l'état l'art, celle-ci a permis de poser des fondements solides pour la suite de nos contributions sur l'évolution des méta-modèles et de son impact sur les modèles instances. Elle a posé une taxonomie et un Framework pour la comparaison des solutions de co-évolution utilisables par tout chercheur du domaine. Elle est aussi le fruit d'une collaboration industrielle et académique dans le cadre du projet FUI MONOGE¹. Ce dernier regroupait les partenaires suivants : le LIP6 (lead :R.Bendraou, D-E. Khelladi, aujourd'hui chercheur CNRS au sein de l'équipe INRIA Diverse, Post-Doc : R. Hebig, aujourd'hui Professeur à l'université de Rostock), Mines Nantes (lead : J. Cabot), la DCNS (Naval group) Mia Software (Nantes) et Softeam (Paris).

3 PRÉSENTATION DE CET ÉLÉMENT

3.1 Importance de la co-évolution

L'utilisation des DSML a permis d'augmenter la productivité logicielle dans certains domaines applicatifs allant de 200 % à 750 % (domotique, commutateurs téléphonique, applications web et mobiles). Aujourd'hui, presque tous les domaines commerciaux et industriels ont un DSML dédié et de nombreux outils sont fournis pour aider les entreprises à concevoir leur propre langage de modélisation. Cependant, les DSML ne viennent pas seuls. Leur productivité est principalement assurée par l'écosystème qui en découle. Celui-ci comprend des générateurs de code, des éditeurs de texte et d'interfaces graphiques, des moteurs de vérification de règles ainsi que de nombreux autres artefacts qui constituent la chaîne du développement logiciel. Toute modification du métamodèle peut avoir un impact conséquent sur cette dernière. Airbus, Naval Group et Thales, avec lesquelles nous avons collaboré sur les projets de recherche Galaxy, MoNoGe et MeRGE, ont mentionné que certains modèles pouvaient comporter jusqu'à 500 000 éléments de modèle à migrer en cas d'évolution du DSML. Les contraintes OCL constituent un autre exemple d'artefacts susceptibles d'être affectés par l'évolution du métamodèle. Le métamodèle UML en contient plus de 700 ! Par exemple, lors de l'évolution des diagrammes de classe UML de la version 1.5 à 2.0, 238 éléments de langage ont été ajoutés, supprimés ou modifiés impactant ainsi des centaines de règles OCL. Les verrous liés à la problématique de la co-évolution elle-même ne seront pas abordés ici mais sont détaillés dans la section présentant les axes de recherche de l'équipe.

3.2 Contributions

Une multitude d'approches de co-évolution de métamodèle/modèle ont été proposées au cours des dix dernières années. Cependant, leur diversité, l'absence de taxonomie précise, de classification des solutions, mettent les acteurs industriels dans la difficulté pour choisir une approche plutôt qu'une autre. Dans cet article, nous avons mené une étude sur 31 approches de co-évolution de métamodèle/modèle et avons réalisé les contributions détaillées dans les paragraphes suivants.

1. Projet MoNoGe <https://monoge.lip6.fr/index.html> (page vérifiée : 08/03/2023)

3.3 Identification des approches

Le premier objectif était de recenser toutes les approches pertinentes qui traitent de la co-évolution. Nous avons donc effectué une recherche documentaire systématique, comme le décrit Kitchenham [1]. Ce processus a permis d'identifier plus de 130 articles. Dans l'étape suivante, nous avons réexaminé ces articles afin d'exclure les candidats non pertinents. Nous avons aussi exclu les articles qui ne présentaient pas d'approches soutenant la co-évolution MM-M. Nous avons accepté d'inclure dans notre enquête 56 articles appartenant à 31 approches différentes que nous avons par la suite catégorisées en six groupes tels que présentés dans le papier.

3.4 Création de la taxonomie

Dans un premier temps, nous avons systématiquement lu tous les articles et recueilli des informations sur les techniques proposées. Nous avons ainsi obtenu une taxonomie unifiée et révisée pour le domaine, qui est suffisamment complète pour couvrir les tâches et les spécificités des approches existantes, ainsi que les potentielles variations. La taxonomie couvre l'identification des types de changements possibles de méta-modèles sur lesquels nous avons posé des définitions formelles. Elle définit également les étapes du processus de co-évolution qui sont 1) la capture des changements, 2) identification du type de changements selon les définitions formelles que nous avons fournies et 3) impact de l'évolution du méta-modèle sur les modèles instances. Sur chacune de ces trois étapes, nous avons rigoureusement identifié toutes les stratégies et techniques proposées par l'état de l'art que nous avons par la suite catégorisées et formalisées par des Features modèles.

3.5 Classification des approches

Dans un second temps, nous avons repris les articles identifiés afin d'évaluer systématiquement la manière dont les étapes de la co-évolution telles que décrites dans notre taxonomie étaient prises en charge par les différentes approches. Pour chaque approche, nous avons également étudié attentivement les articles sélectionnés et les sites web référencés pour savoir si une mise en œuvre de l'approche a été fournie et si elle était accessible. Cette classification est présentée sous forme de deux grands tableaux.

3.6 Aide à la décision

Notre ultime objectif est d'aider les utilisateurs à choisir entre les approches existantes. Cette aide à la décision permet à l'utilisateur de naviguer entre les différentes caractéristiques prises en compte dans la taxonomie que nous avons fournie. Sur la base des conséquences du choix d'une caractéristique, nous avons identifié des questions que les utilisateurs potentiels devraient prendre en considération. Ces questions concernent l'exactitude des résultats requis, le degré d'automatisation nécessaire, la capacité à consacrer des efforts à l'adoption et à l'application de l'approche, ainsi que les contraintes organisationnelles susceptibles d'influer sur l'applicabilité d'une approche. Pour chaque question, nous proposons des critères qui peuvent être utilisés pour identifier les approches les plus adéquates. Le résultat final se présente sous la forme d'arbres de décisions qui peuvent être utilisés pour guider l'adoption de l'approche de co-évolution répondant au mieux aux besoins de l'utilisateur.

4 RÉFÉRENCES BIBLIOGRAPHIQUES

- [1] Barbara Kitchenham. Procedures for performing systematic reviews. Keele university. technical report tr/se-0401, Department of Computer Science, Keele University, UK, 2004.

ÉLÉMENT DE PORTFOLIO 02



Jeu de données consolidées

1 DÉFINITION DE CET ÉLÉMENT

Titre de l'élément : le Model Checking Contest

URL de l'élément : <https://mcc.lip6.fr>

2 MOTIVATIONS DU CHOIX DE CET ÉLÉMENT

Notre motivation pour faire un focus sur le Model Checking Contest (MCC) est d'illustrer un travail important de l'équipe MoVe au sein de la communauté du model checking via cet événement annuel, créé en 2011 et qui se tient tous les ans depuis (2023 verra la 13^e édition se tenir à Paris en avril dans le cadre de la conférence ETAPS).

3 PRÉSENTATION DE CET ÉLÉMENT

Le Model Checking Contest est un concours annuel de logiciels de *model checking* dédié à l'évaluation des outils de vérification formelle pour les systèmes concurrents. Cet événement a été créé en 2011, dans la lignée d'autres concours similaires dans des disciplines voisines : SAT [1], SMT [2], CASC [8], etc.

Le MCC se focalise sur un problème particulièrement complexe : la vérification de systèmes concurrents asynchrones (au contraire de la *Hardware Model Checking competition* [4] qui concerne les systèmes synchrones). Sans être le premier du domaine, il émerge assez tôt dans le paysage des compétitions logicielles de ce type [3].

Le MCC est constitué de deux parties :

- ▶ un "appel à modèles" permettant à la communauté de soumettre des spécifications qui enrichissent chaque année un jeu de test que nous évoquerons plus loin,
- ▶ un "appel à outils" permettant aux développeurs d'outils de soumettre leur travail et de le comparer dans des conditions similaires aux autres outils compétiteurs.

3.1 Organisation de la compétition

Les outils sont tous soumis à 6 épreuves : calcul de l'espace d'états, calcul de propriétés structurelles, calcul de bornes sur des ensembles de places (16 formules), évaluation de 16 formules d'accessibilités, évaluation de 16 formules CTL¹ et évaluation de 16 formules LTL².

Les outils peuvent bien sûr ne participer qu'à un sous-ensemble des épreuves proposées. Depuis 2014, les outils sont classés pour chacune des 6 épreuves de la compétition.

3.2 Un benchmark de référence consolidé

Le MCC fournit à la communauté un benchmark de référence qui est enrichi chaque année par de nouvelles contributions via l'"appel à modèles".

Ces modèles sont fournis sous la forme d'un réseau de Petri place/transition ou symétrique [5] exprimé avec le format standard PNML [6]. Ils sont vérifiés par un comité des modèles. Chacun d'eux est également fourni avec une "fiche modèle" décrivant brièvement ce qui est spécifié et expliquant, dans le cas des modèles paramétrés, le nombre d'instances du modèle et les valeurs du ou des paramètres caractérisant ces instances et leur complexité (<https://mcc.lip6.fr/models.html>).

1. Computational Tree Logic
2. Linear Temporal Logic

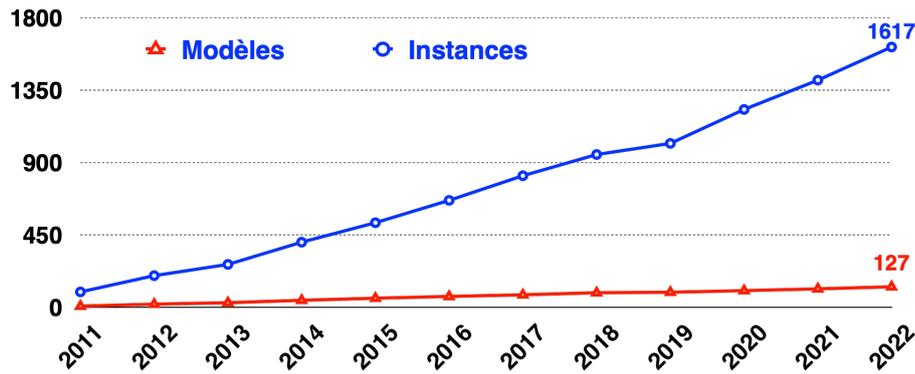


FIGURE 1 – Évolution du nombre de modèles et d'instance de modèles du MCC

La figure 1 illustre l'évolution des modèles offerts à la communauté. Leur intérêt est qu'ils proviennent de toutes origines. Ils sont issus de problèmes académiques, de cas d'études académiques réalisés dans un contexte de collaborations avec des industriels, et enfin de problèmes industriels (parfois anonymisés). Ils modélisent également des problèmes issus de domaines applicatifs très différents, de l'ingénierie des protocoles des caches de processeurs multi-cœurs, des algorithmes répartis, des systèmes biologique, etc.

De plus, pour les 4 épreuves impliquant des formules, un jeu est aléatoirement produit chaque année. Ainsi, si on considère les éditions entre 2015 (date à laquelle le format de la compétition s'est stabilisé) et 2022, ce sont 131 680 formules de chaque catégorie qui ont été produites. Pour environ la moitié d'entre elles, un résultat attendu a été calculé par une majorité qualifiée d'outils.

Les modèles et formules utilisés pour une édition du MCC sont disponibles publiquement dans une archive et les résultats sont également fournis sous la forme d'un fichier CSV consolidé. Cela a permis une analyse pluriannuelle des résultats du MCC et du comportement des outils entre 2015 et 2019 [7].

3.3 Bénéfices pour la communauté

Les bénéfices pour la communauté sont multiples :

- ▶ Le premier est que les outils développés par la communauté cible ont vu leur robustesse augmenter considérablement. Certains développeurs incluent des comparaisons avec les formules dont les résultats sont connus dans leur chaîne d'intégration continue.
- ▶ Le second est que certains développeurs ont pu échanger leurs techniques et partagent des modules logiciels comme, par exemple, le module transformant un réseau de Petri symétrique en un réseau place/transition équivalent.
- ▶ Le second est que le benchmark ainsi produit devient progressivement une référence. Il est utilisé (parfois de manière incomplète) et référencé dans de plus en plus d'articles produisant des résultats expérimentaux. En témoignent les 174 références au benchmark issu du Model Checking contest recensées en février 2023.

4 RÉFÉRENCES BIBLIOGRAPHIQUES

- [1] The international sat competition web page : <http://www.satcompetition.org>. 2022.
- [2] Clark Barrett, Leonardo de Moura, and Aaron Stump. Smt-comp : Satisfiability modulo theories competition. In *17th International Conference on Computer Aided Verification – CAV*, volume 3576 of *Lecture Notes in Computer Science*, pages 20–23. Springer, 2005.
- [3] Ezio Bartocci, Dirk Beyer, Paul E. Black, Grigory Fedyukovich, Hubert Garavel, Arnd Hartmanns, Marieke Huisman, Fabrice Kordon, Julian Nagele, Mihaela Sighireanu, Bernhard Steffen, Martin Suda, Geoff Sutcliffe, Tjark Weber, and Akihisa Yamada. Toolympics 2019 : An overview of competitions in formal methods. In Dirk Beyer, Marieke Huisman, Fabrice Kordon, and Bernhard Steffen, editors, *Tools and Algorithms for the Construction and Analysis of Systems*, pages 3–24, Cham, 2019. Springer International Publishing.
- [4] Armin Biere, Tom van Dijk, and Keijo Heljanko. Hardware model checking competition 2017. In *FMCAD*, page 9. IEEE, 2017.

- 
- [5] G. Chiola, C. Dutheillet, G. Franceschinis, and S. Haddad. A symbolic reachability graph for coloured Petri nets. *Theoretical Computer Science*, 176(1–2) :39–65, 1997.
 - [6] Lom Messan Hillah, Ekkart Kindler, Fabrice Kordon, Laure Petrucci, and Nicolas Trèves. A primer on the Petri Net Markup Language and ISO/IEC 15909-2. *Petri Net Newsletter*, 76 :9–28, October 2009.
 - [7] Fabrice Kordon, Lom Messan Hillah, Francis Hulin-Hubard, Loïg Jezequel, and Emmanuel Paviot-Adet. Study of the efficiency of model checking techniques using results of the MCC from 2015 To 2019. *International Journal on Software Tools for Technology Transfer*, June 2021.
 - [8] G. Sutcliffe. The cade-26 automated theorem proving system competition - casc-26. *AI Commun.*, 30 :419–432, 2017.

ÉLÉMENT DE PORTFOLIO 03



Logiciel ou bibliothèque logicielle

1 DÉFINITION DE CET ÉLÉMENT

Titre de l'élément : Outil de vérification formelle ITS-Tools

URL de l'élément : <https://ddd.lip6.fr>

2 MOTIVATIONS DU CHOIX DE CET ÉLÉMENT

L'outil ITS-Tools est un outil de vérification formelle et un environnement de modélisation qui permet l'analyse du comportement de systèmes a fortiori concurrents. Il permet en particulier la vérification de propriétés de logique temporelle sur des systèmes exprimés dans des langages divers (réseaux de Petri, langages d'automates communicants, automates temporisés. . .), à l'aide de plusieurs stratégies de décision. Développé au sein de l'équipe MoVe depuis 2008, cet outil de référence a connu des évolutions importantes sur la période 2017-2022, lui permettant de s'imposer notamment au Model-Checking Contest [6] (MCC) avec de nombreuses médailles d'or depuis 2020.

Son développement continu amène des collaborations applicatives dans divers domaines comme la sécurité avec les arbres d'attaque [8], la modélisation et l'analyse d'algorithmes auto-stabilisants [9], l'utilisation pour analyser les systèmes de manufacture flexibles [5], ou l'analyse de configurations d'un noyau linux. Les interactions logicielles fortes entre ITS-Tools et l'outil Spot [4] de l'EPITA renforcent les collaborations long terme (depuis 2005) entre nos deux équipes.

3 PRÉSENTATION DE CET ÉLÉMENT

3.1 Intégration multi-solution

Une des forces de ITS-Tools est la multiplicité des procédures de décision qu'il est capable d'appliquer. Organisées en portfolio, ces stratégies sont complémentaires et coopèrent pour prouver une propriété donnée. En particulier, ITS-tools possède :

- ▶ Des solutions basées sur des diagrammes de décision réduits particuliers [2] (et aussi développés dans l'équipe), qui permettent de faire face à de très grands espaces d'états
- ▶ Des solutions basées sur les stratégies de réduction d'ordre partiel, appuyé par les technologies de LTS-Min [1], un outil développé à Twente (NL).
- ▶ Des solutions qui exploitent la puissance des solvers SMT modernes, en appui sur le solveur Z3 de Microsoft [3] en particulier, et permettent de surapproximer les comportements d'un système par des équations et contraintes
- ▶ Des stratégies d'explorations sans mémoire ou avec occupation mémoire bornée, basés sur des runs pseudo-aléatoires ou guidés heuristiquement [11], qui permettent par "sampling" de sous approximer les comportement d'un système
- ▶ Des stratégies de réduction structurelles [11], qui vont réécrire le modèle initial tout en préservant la propriété de façon à permettre une analyse sur un modèle plus simple
- ▶ Pour l'analyse de propriétés linéaires comme LTL ou PSL, ITS-Tools dispose de stratégies particulièrement avancés, issues de collaborations avec l'EPITA autour de l'outil Spot [4] qui est spécialisé dans la manipulation d'automates de Büchi. Ces stratégies permettent notamment de profiter de réductions (structurelles ou réductions d'ordre partiel en particulier) même si le langage de la propriété est (partiellement) sensible au bégaiements [7].

L'ensemble de l'outil est adapté aux architectures multi-core modernes, est fortement optimisé, et entièrement open source (FOSS, principalement sous licences GPL/EPL).

3.2 Participation au MCC

ITS-Tools participe régulièrement au MCC depuis ses premières éditions ; ces participations ont permis d'améliorer sensiblement l'outil, corrigeant les fautes éventuelles et améliorant les performances continuellement.

L'outil est en général sur les podiums du MCC :

- ▶ en 2017, il remporte 1 médaille d'argent et 4 bronze,
- ▶ en 2018, il remporte 4 médailles de bronze,
- ▶ en 2019, il remporte 2 médailles d'argent et 4 bronze,
- ▶ en 2020, il remporte 3 médailles d'or, 1 médaille d'argent et 1 médaille de bronze,
- ▶ en 2021, il remporte 3 médailles d'or, 2 médailles d'argent et 1 médaille de bronze,
- ▶ en 2022, il remporte 2 médailles d'or, 3 médailles d'argent et 1 médaille de bronze.

Comme on le voit, depuis 2020 ITS-tools a sensiblement progressé, s'arrogeant l'or dans trois catégories en 2021 (propriétés globales, accessibilité, bornes) et l'argent en LTL et CTL. Ceci est principalement lié aux nouvelles technologies introduites dans [11] qui lient l'utilisation de solvers de contraintes [3], d'explorations sans mémoire et de réductions structurelles pour traiter efficacement les problèmes les plus difficiles.

3.3 Utilisations notables de l'outil

L'outil est utilisé par d'autres équipes académiques ainsi que par certains industriels. On peut mentionner en particulier :

- ▶ Intégration comme principal moteur de solution dans AtSyRa [8], un outil développé à l'IRISA de Rennes pour l'analyse d'arbres d'attaque/défense
- ▶ Intégration comme moteur de solution dans l'outil Ecco [12], développé à l'IBISC d'Evry, pour l'analyse du comportement d'écosystèmes et de systèmes écologiques
- ▶ Intégration comme moteur d'analyse de système cyber-physiques dans des travaux [5] des universités de Aachen et Stuttgart.

Le papier de référence sur l'outil [10] est cité 72 fois selon Google scholar, donc 52 citations depuis 2019.

4 RÉFÉRENCES BIBLIOGRAPHIQUES

- [1] Stefan Blom, Jaco van de Pol, and Michael Weber. Ltsmin : Distributed and symbolic reachability. In *CAV*, volume 6174 of *Lecture Notes in Computer Science*, pages 354–359. Springer, 2010.
- [2] Maximilien Colange, Souheib Baarir, Fabrice Kordon, and Yann Thierry-Mieg. Towards distributed software model-checking using decision diagrams. In *CAV*, volume 8044 of *Lecture Notes in Computer Science*, pages 830–845. Springer, 2013.
- [3] Leonardo Mendonça de Moura and Nikolaj Bjørner. Z3 : an efficient SMT solver. In *TACAS*, volume 4963 of *Lecture Notes in Computer Science*, pages 337–340. Springer, 2008.
- [4] Alexandre Duret-Lutz, Etienne Renault, Maximilien Colange, Florian Renkin, Alexandre Gbaguidi Aisse, Philipp Schlehuber-Caissier, Thomas Medioni, Antoine Martin, Jérôme Dubois, Clément Gillard, and Henrich Lauko. From spot 2.0 to spot 2.10 : What's new ? In *CAV (2)*, volume 13372 of *Lecture Notes in Computer Science*, pages 174–187. Springer, 2022.
- [5] Marco Grochowski, Hendrik Simon, Dimitri Bohlender, Stefan Kowalewski, Andreas Löcklin, Timo Müller, Nasser Jazdi, Andreas Zeller, and Michael Weyrich. Formale methoden für rekonfigurierbare cyber-physische systeme in der produktion. *Autom.*, 68(1) :3–14, 2020.
- [6] Fabrice Kordon, Lom Messan Hillah, Francis Hulin-Hubard, Loïg Jezequel, and Emmanuel Paviot-Adet. Study of the efficiency of model checking techniques using results of the mcc from 2015 to 2019. *International Journal on Software Tools for Technology Transfer*, 2021.
- [7] Emmanuel Paviot-Adet, Denis Poitrenaud, Etienne Renault, and Yann Thierry-Mieg. LTL under reductions with weaker conditions than stutter invariance. In *FORTE*, volume 13273 of *Lecture Notes in Computer Science*, pages 170–187. Springer, 2022.
- [8] Sophie Pinchinat, Mathieu Acher, and Didier Vojtisek. Atsyra : An integrated environment for synthesizing attack trees - (tool paper). In *GraMSec@CSF*, volume 9390 of *Lecture Notes in Computer Science*, pages 97–101. Springer, 2015.

- 
- [9] Arnaud Sangnier, Nathalie Sznajder, Maria Potop-Butucaru, and Sébastien Tixeuil. Parameterized verification of algorithms for oblivious robots on a ring. *Formal Methods Syst. Des.*, 56(1) :55–89, 2020.
- [10] Yann Thierry-Mieg. Symbolic model-checking using its-tools. In *TACAS*, volume 9035 of *Lecture Notes in Computer Science*, pages 231–237. Springer, 2015.
- [11] Yann Thierry-Mieg. Symbolic and structural model-checking. *Fundam. Informaticae*, 183(3-4) :319–342, 2021.
- [12] Colin Thomas, Maximilien Cosme, Cédric Gaucherel, and Franck Pommereau. Model-checking ecological state-transition graphs. *PLoS Comput. Biol.*, 18(6), 2022.

ÉLÉMENT DE PORTFOLIO 04



Publication

1 DÉFINITION DE CET ÉLÉMENT

Titre de l'élément : A novel approach for Software Architecture Product Line Engineering. Publié dans la revue Journal of Systems and Software, vol. 186, pp. 111191, (Elsevier) (2022)

URL de l'élément : <https://hal.science/hal-03885616>

2 MOTIVATIONS DU CHOIX DE CET ÉLÉMENT

Il s'agit d'un article publié dans la revue Journal of Systems and Software, une revue de rang A qui est une extension d'un premier article publié dans la conférence ICECCS 2019 [2]. Nous avons choisi de présenter cet article car il montre l'impact très positif de nos développements logiciels sur notre visibilité. L'article présente une extension de l'outil BUT4Reuse pour les architectures logicielles.

3 PRÉSENTATION DE CET ÉLÉMENT

3.1 Contexte

BUT4Reuse (Bottom-Up Technology for Reuse) est le logiciel développé au sein de l'équipe MoVe pour l'extraction (ou l'inférence) de lignes de produits logiciels à partir d'artefacts logiciels existants. BUT4Reuse a été conçu comme un cadre générique et extensible pour supporter différents types d'artefacts (modèles, code source, etc.) [3]. Dans cet article, nous avons considéré l'extraction de lignes de produits à partir d'artefacts logiciels représentant des architectures logicielles. Nous avons considéré des architectures logicielles obtenues à partir de code source de systèmes développés dans le cadre de l'OSGi, tels que l'IDE Eclipse. La spécification OSGi définit un modèle de composant et un cadre de travail pour créer des systèmes Java hautement modulaires [4]. Les IDE basés sur Eclipse sont une collection de produits logiciels similaires qui partagent un ensemble d'artefacts logiciels. Extraire une ligne de produits à partir de cette collection représente l'objectif principal traité dans cet article.

Dans ce contexte, nous avons identifié cinq défis principaux : i) Comment extraire une architecture logicielle à partir de code source de chaque variante ; ii) Comment comparer les variantes d'architecture logicielle pour identifier les parties communes et trouver les caractéristiques (features) variables ; iii) Comment construire la Ligne de Produits Architecturale (LdPA) avec une spécification explicite de la variabilité au niveau architectural ; iv) Comment simplifier et réduire la complexité des architectures récupérées. Cet article propose une approche permettant de résoudre tous ces défis. Une validation expérimentale intensive a été présentée en utilisant le benchmark EFLBench [4].

3.2 Résultats

L'approche proposée est composée de deux processus principaux : i) un processus ascendant (P1) d'inférences de LdPA ; ce sous-processus se commence d'abord par la rétro-ingénierie des architectures logicielles à partir du code source de chaque variante logicielle. Ensuite, il reconstruit un LdPA pour ces variantes d'architecture logicielle ; ii) un processus de dérivation (P2) qui permet de dériver de nouvelles variantes (variantes d'architecture logicielle).

Inférence de LdPA. La construction de LdPA à partir de variantes d'architecture logicielle est réalisée en implémentant un nouvel adaptateur pour BUT4Reuse. Cet adaptateur suit les principes généraux de BUT4Reuse en définissant : 1) les éléments atomiques ; 2) la fonction de similarité ; 3) la définition des contraintes structurelles. Les algorithmes d'identification de blocs de BUT4Reuse sont utilisés par la suite pour la comparaison et l'inférence de la LdPA. La LdPA obtenue est composée d'une part, d'un feature model spécifiant la variabilité et d'autre part, de l'ensemble de fragments d'architecture associés à chacune des features selon une approche compositionnelle [1].

Dérivation de variantes d'architecture logicielles. Le deuxième processus a comme objectif de créer de nouvelles variantes d'architectures à partir de la LdPA obtenue. Il suit les grandes lignes de dérivation dans les approches compositionnelle en intégrant ce qui est appelé un "composer" qui permet d'assembler les fragments d'architecture en fonction des features sélectionnées.

Expérimentation et validation. Nous avons validé l'approche proposée en utilisant un ensemble de données générées à partir de EFLBench. En effet, EFLBench permet de générer d'une manière intensive des distributions de l'IDE Eclipse. Nous avons identifié deux questions de recherche liées aux deux processus P1 et P2 enrichis avec un ensemble de métriques d'évaluation. Dans nos expérimentations, nous avons généré 100 variantes d'IDE Eclipse et les résultats obtenus montrent des taux élevés de précisions. Nous avons utilisé la LdPA obtenue pour générer des variantes qui sont fonctionnellement correctes.

4 RÉFÉRENCES BIBLIOGRAPHIQUES

- [1] Sven Apel, Don S. Batory, Christian Kästner, and Gunter Saake. *Feature-Oriented Software Product Lines - Concepts and Implementation*. Springer, 2013.
- [2] Mohamed Lamine Kerdoudi, Tewfik Ziadi, Chouki Tibermacine, and Salah Sadou. Recovering Software Architecture Product Lines. In *ICECCS 2019 - 24th International Conference on Engineering of Complex Computer Systems*, pages 226–235, Nansha, Guangzhou, China, November 2019. IEEE.
- [3] Jabier Martinez, Tewfik Ziadi, Tegawendé Bissyandé, Jacques Klein, and Yves Le Traon. Bottom-Up Adoption of Software Product Lines - A Generic and Extensible Approach. In *19th International Software Product Line Conference (SPLC)*, pages 101–110, Nashville, TN, United States, July 2015. ACM.
- [4] Jabier F Martinez, Tewfik Ziadi, Mike Papadakis, Tegawendé F. Bissyandé, Jacques Klein, and Yves Le Traon. Feature location benchmark for extractive software product line adoption research using realistic and synthetic Eclipse variants. *Information and Software Technology*, July 2018.

ÉLÉMENT DE PORTFOLIO 05



Publication

1 DÉFINITION DE CET ÉLÉMENT

Titre de l'élément : Parameterized Synthesis for Fragments of First-Order Logic over Data Words. Publié dans les actes de FOSSACS'20.

URL de l'élément : <https://arxiv.org/pdf/1910.14294.pdf>

2 MOTIVATIONS DU CHOIX DE CET ÉLÉMENT

Cet article a été publié dans les actes de la conférence internationale FOSSACS'20, conférence de rang A qui rassemble des travaux de recherche fondamentale ayant des applications claires pour la science du logiciel. Il rassemble une partie du travail fait pendant la thèse de Mathieu Lehaut, doctorant de l'équipe co-dirigé par Béatrice Bérard et Nathalie Sznajder de l'équipe MoVe d'une part, et Benedikt Bollig du LMF d'autre part. Il illustre les travaux de l'équipe dans les aspects théoriques de la vérification.

3 PRÉSENTATION DE CET ÉLÉMENT

3.1 Contexte

Cet article traite du problème de synthèse, tel que défini par Alonzo Church en 1963 : il consiste à dériver automatiquement un système (sous la forme d'un modèle formel) à partir de sa spécification. Ainsi, si la spécification est irréalisable, on peut le savoir avant tout travail de réalisation et, dans le cas contraire, on obtient automatiquement un modèle correct par construction du système à réaliser. Il s'agit dans ce problème de systèmes réactifs ouverts, c'est-à-dire des systèmes qui doivent maintenir un comportement correct en interaction avec un environnement incontrôlable. Le problème est formulé de la façon suivante : étant donné un alphabet d'actions pour le système et un alphabet d'actions pour l'environnement, chacun des acteurs choisit une action alternativement, ce qui produit un mot infini. Étant donné une spécification sur les mots infinis, est-il possible pour le système d'avoir une stratégie lui permettant de toujours produire un mot autorisé par la spécification, quel que soit le comportement du système ? Dans le cas de systèmes simples, le problème est à présent bien compris et des outils arrivent à maturité. Lorsque le système à synthétiser doit être composé de plusieurs processus, ayant chacun une vision locale, le problème est plus difficile (et en général indécidable). Dans cet article nous nous sommes intéressés à la synthèse de systèmes distribués dans lesquels le nombre de processus est variable : les systèmes paramétrés. Ce type de systèmes est très répandu, que ce soit dans l'algorithmique distribuée, les réseaux ad-hoc, les systèmes biologiques... La question devient donc : existe-t-il un nombre de processus pour lesquels le système peut s'assurer de toujours satisfaire la spécification, quelles que soient les actions choisies par l'environnement ? On peut également considérer une version "universelle" : le système peut-il toujours garantir de satisfaire la spécification, quelles que soient les actions choisies par l'environnement, et quel que soit le nombre de processus ?

3.2 Résultats

Dans le cas des systèmes infinis, il n'est plus possible de décrire les exécutions de tels systèmes par des mots sur un alphabet *fini* : en effet, chaque action doit être localisée sur le processus qui l'a émise, ce qui revient à étiqueter chaque action par l'identifiant du processus sur lequel elle s'exécute. Dans le cas où le nombre de processus n'est pas connu à l'avance, cela revient à considérer un alphabet *infini*.

Décrire une spécification revient donc à définir un langage sur un alphabet infini. Plusieurs formalismes ont été proposés pour cela ces dernières années, que ce soit par des modèles d'automates ou des logiques. S'il n'existe pas (encore) de notion de langage régulier sur les alphabets infinis, la logique du premier ordre (FO) sur les mots avec données est largement acceptée comme standard pour décrire de tels modèles : les mots sont constitués de lettres ayant une composante finie, et une composante prise dans un domaine infini (les données), qui dans

notre cas, représente l'identifiant d'un processus. Son fragment FO², dans lequel au plus deux variables peuvent apparaître dans les formules, est décidable pour la satisfaisabilité et l'universalité.

Nous étudions donc dans cet article le problème de synthèse pour les systèmes paramétrés de la façon suivante : étant donnée une formule logique donnée dans la logique du premier ordre sur les mots avec données, existe-t-il un nombre de processus permettant au système de garantir une exécution correcte, quelle que soit la manière dont l'environnement se comporte ?

Notre modélisation comporte encore deux spécificités : dans le cas d'un système distribué paramétré, donc mettant potentiellement en jeu un grand nombre de processus, il n'est pas raisonnable d'envisager une communication synchrone entre le système et son environnement, avec une alternance d'actions entre les deux. Nous considérons donc le problème dans un cadre asynchrone, dans lequel le système n'a pas de contrôle sur le moment où les actions de l'environnement vont avoir lieu. Enfin, nous considérons que le système est composé de trois types de processus : les processus "système" qui n'ont aucune interaction avec l'environnement, les processus "environnement", qui ne comportent aucune action du système, et les processus "mixtes", sur lesquels ont lieu à la fois des actions du système et des actions de l'environnement.

Nos résultats sont les suivants :

- ▶ le problème de synthèse que nous considérons est indécidable pour FO²,
- ▶ il est également indécidable pour le fragment de FO dans lequel on s'interdit d'imposer un ordre aux actions
- ▶ il devient décidable si l'on fixe le nombre de processus contrôlés par l'environnement, en laissant libre le nombre de processus "système".

3.3 Impact

La logique sur les mots avec données, en lien avec la théorie des automates, et les problèmes de vérification et de synthèse est intensivement étudiée ces dernières années, pour ses applications à la fois dans les systèmes paramétrés et dans les bases de données. Ce travail apporte donc un éclairage supplémentaire important dans ce domaine.

ÉLÉMENT DE PORTFOLIO 06



Publication

1 DÉFINITION DE CET ÉLÉMENT

Titre de l'élément : A First-Order Logic Semantics for Communication-Parametric BPMN Collaborations. Publication à la 17ème International Conference on Business Process Management (BPM 2019).

URL de l'élément : <https://hal-cnrs.archives-ouvertes.fr/hal-02138366>

2 MOTIVATIONS DU CHOIX DE CET ÉLÉMENT

Cet article [2] revêt une importance particulière pour nous à plusieurs titres. Tout d'abord il s'agit d'un travail collaboratif à la fois en termes de collaborations internes et externes. Ce travail est une illustration de l'une des forces de l'équipe MoVe : intégrer des chercheurs en sciences du logiciel (modélisation en particulier) et en vérification. Mettant en oeuvre, en complément des compétences de MoVe, celles de l'équipe ACADIE à l'IRIT, il s'intègre totalement dans les objectifs du projet ANR PARDI (<https://anr.fr/Projet-ANR-16-CE25-0006>) dont les deux équipes étaient partenaires. Ce travail a plus généralement été le point de démarrage de nouvelles collaborations [3–5] entre le LIP6 (Paris), l'IRIT (Toulouse), l'IFSTTAR (Villeneuve d'Ascq), et à l'avenir le LIS (Aix-Marseille), autour du cadre logiciel fBPMN (<https://github.com/pascalpoizat/fbpmn>). Enfin, cette contribution, qui se positionne dans le domaine de la science des processus métiers, a donné lieu au *Best Paper Award* de l'édition 2019 de la conférence majeure du domaine, BPM.

3 PRÉSENTATION DE CET ÉLÉMENT

3.1 Problématique et synthèse des contributions

BPMM [1] est le standard de fait pour la représentation des processus métiers sous forme diagrammatique. C'est aussi une norme ISO depuis 2013. BPMN permet à la fois la modélisation de ces processus à un haut niveau d'abstraction et leur exécution à l'aide de moteurs d'exécution de processus. La sémantique de BPMN n'est cependant décrite, au sein du standard, que sous une forme semi-formelle ayant en grande partie recours au langage naturel (en pratique, l'anglais).

Un important effort de formalisation de BPMN, ainsi que de développement d'outils d'analyse des processus, a été fait par la communauté académique au cours du temps. Cependant, cet effort se limitait souvent aux parties les plus simples du standard, par exemple celles dont la sémantique était aisément décrite par un plongement dans les réseaux de Petri. Certains éléments du standard, comme l'élément de structuration de flot de contrôle appelé "porte inclusive" et sa sémantique non-locale, faisaient à eux-mêmes l'objet de plusieurs publications. Enfin, des aspects tels que ceux liés à la communication inter-processus ou à la gestion du temps étaient moins étudiés ou laissés de côté, alors qu'ils pouvaient être centraux pour certains processus, ceux opérant au dessus de plateformes IoT par exemple.

Dans ce travail, nos contributions ont été de plusieurs natures :

- ▶ nous avons proposé une sémantique directe en logique du premier ordre (sans plongement dans un langage ou modèle tiers, donc assurant une meilleure indépendance et ouvrant à l'utilisation de différents outils d'analyse des processus) à un sous-ensemble important de BPMN, qui intègre la possibilité de prise en compte de différents modèles de communication inter-processus ;
- ▶ nous avons implanté cette sémantique et les principales propriétés portant sur les processus métiers (soundness et terminaison) dans la logique TLA^+ .
- ▶ nous avons outillé notre approche avec un cadre logiciel open source et extensible, fBPMN.

3.2 Contributions formelles

Notre modélisation de la sémantique de BPMN se base sur deux modèles principaux. Les processus BPMN sont représentés par (comprendre traduits en) un graphe étiqueté, où les types correspondent aux éléments de modélisation BPMN (conditions de démarrage, envoi/réception de message, tâches de travail, gestion du flot de contrôle, etc). La sémantique d'exécution d'un tel modèle est ensuite définie sur un graphe marqué où des jetons sont associés aux nœuds du graphe. Chaque élément syntaxique de BPMN se voit équipé de règles formelles d'exécution qui lui sont propres et transforment le marquage du graphe, induisant ainsi une structure de graphe d'états. C'est ce graphe d'état qui peut ensuite être analysé à l'aide d'outils en fonction de la traduction de la sémantique proposée dans tel ou tel outil de vérification. Dans notre cas nous avons proposé une traduction en TLA⁺.

3.3 L'outil fBPMN

L'outil fBPMN est constitué d'un ensemble de théories TLA⁺ implantant la sémantique formelle d'un important sous-ensemble de BPMN, de scripts et de programmes de transformation de modèle BPMN vers TLA⁺ et d'un moyen d'animation de contre-exemple utilisable lorsque le modèle vérifié (par un outil externe d'analyse TLA⁺ appelé TLC) ne respecte pas certaines propriétés souhaitées. fBPMN permet de vérifier à la fois les modèles mono-processus (workflow diagrams) et les modèles avec communication (collaboration diagrams). Le modèle de cette communication peut être pris parmi 7 différents modèles représentatifs des modèles de communications réels. Depuis la publication [2], fBPMN intègre aussi une Web application pour faciliter son utilisation par le grand public.

3.4 Développements postérieurs

Suite à cette contribution, plusieurs perspectives ont été développées portant sur la prise en compte d'éléments additionnels au noyau BPMN traité (différentes sémantiques du temps dans [3, 4], aspects spatiaux-contextuels dans [5]). Au niveau international, l'un des auteurs de la publication a intégré le comité de programme de la conférence BPM et été rapporteur d'une thèse en Italie sur le sujet.

4 RÉFÉRENCES BIBLIOGRAPHIQUES

- [1] Business Process Model and Notation, version 2.0.2. Technical report, OMG, January 2014. <http://www.omg.org/spec/BPMN/2.0.2/>.
- [2] Sara Houhou, Souheib Baair, Pascal Poizat, and Philippe Quéinnec. A first-order logic semantics for communication-parametric BPMN collaborations. In *17th International Conference on Business Process Management (BPM)*, volume 11675 of *Lecture Notes in Computer Science*, pages 52–68, 2019. Best paper award.
- [3] Sara Houhou, Souheib Baair, Pascal Poizat, and Philippe Quéinnec. A direct formal semantics for BPMN time-related constructs. In *16th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE)*, pages 138–149, 2021.
- [4] Sara Houhou, Souheib Baair, Pascal Poizat, Philippe Quéinnec, and Laid Kahloul. A first-order logic verification framework for communication-parametric and time-aware BPMN collaborations. *Information Systems*, 104 :101765, 2022.
- [5] Rim Saddem-Yagoubi, Pascal Poizat, and Sara Houhou. Business processes meet spatial concerns : the sBPMN verification framework. In *24th International Symposium on Formal Methods (FM)*, volume 13047 of *Lecture Notes in Computer Science*, pages 218–234, 2021.

ÉLÉMENT DE PORTFOLIO 07



Logiciel ou bibliothèque logicielle

1 DÉFINITION DE CET ÉLÉMENT

Titre de l'élément : Outil de développement automatique de SAT solveurs parallèles *Painless*

URL de l'élément : <https://www.lrde.epita.fr/wiki/Painless>

2 MOTIVATIONS DU CHOIX DE CET ÉLÉMENT

Painless (PARallel INSTantiABLE Sat Solver) est un framework qui simplifie l'implémentation et l'évaluation de nouveaux solveurs SAT parallèles pour les environnements à plusieurs cœurs [1]. Les composants de *Painless* peuvent être instanciés indépendamment pour produire un nouveau solveur complet. Le principe directeur est de séparer les composants techniques dédiés à un aspect spécifique de la programmation concurrente, des composants implémentant des heuristiques et des optimisations intégrées dans un solveur SAT parallèle.

Painless a été conçu et développé à partir de 2016 grâce à une collaboration entre les équipes DELYS et MoVe. Ce travail s'est poursuivi ensuite en collaboration avec des collègues du LRDE à l'EPITA et des collègues de l'Université de Waterloo (Canada).

3 PRÉSENTATION DE CET ÉLÉMENT

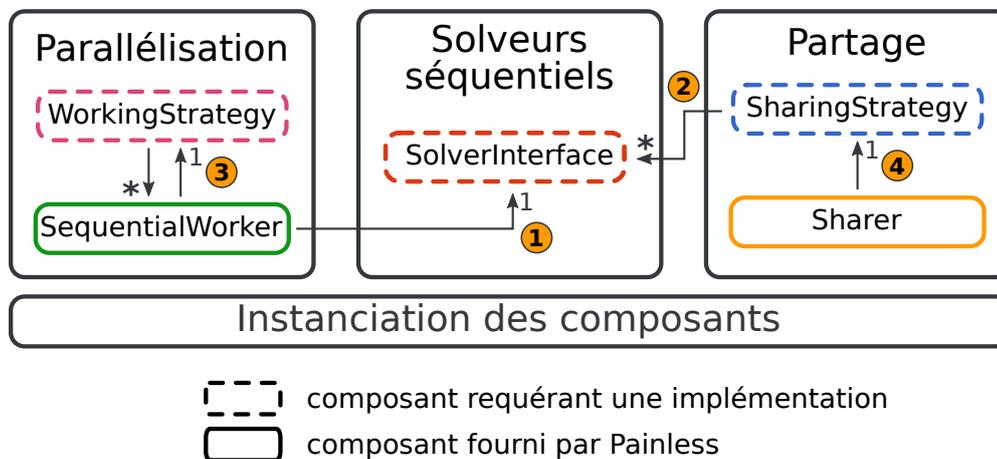


FIGURE 1 – Architecture général de *Painless*

3.1 Architecture de *Painless*

Un solveur SAT parallèle typique repose principalement sur trois concepts de base : moteur(s) séquentiel(s), parallélisation et partage. Ces concepts forment le cœur de l'architecture de *Painless* (voir Figure 1) : le moteur séquentiel est géré par le composant *SolverInterface*. La parallélisation est mise en œuvre par les composants *WorkingStrategy* et *SequentialWorker*. Les composants *SharingStrategy* et *Sharer* sont chargés du partage.

3.2 Participation au à la compétition SAT parallèle

Des SAT-solveurs construits à partir de `Painless` participent régulièrement à la compétition SAT¹ (track parallèle). Ces participations ont permis d'améliorer sensiblement l'outil, corrigeant les fautes éventuelles et améliorant les performances continuellement.

L'outil est en général sur le podium de la compétition : en 2017 (sa première participation), il remporte le bronze. en 2018, 2020 et 2021, il reporte l'or.

3.3 Utilisations notables de l'outil

Au fil des années, `Painless` devient un outil incontournable de la communauté de SAT-solving parallèle. Il a notamment été utilisé pour approfondir les études autour des stratégies dites de *Divide-and-Conquer* [2,3], l'exploitation des structures des communautés dans le SAT-solving parallèle [4], ou encore le renforcement des contraintes apprises [5].

Par ailleurs, `Painless` a été utilisé par d'autres équipes dans la compétition SAT parallèle avec des résultats remarquables. En particulier, 4 des 10 équipes participantes à la compétition de 2022 utilisaient le framework, parmi lesquelles se trouvait le gagnant.

4 RÉFÉRENCES BIBLIOGRAPHIQUES

- [1] Ludovic Le Frioux, Souheib Baarir, Julien Sopena, and Fabrice Kordon. `PaInleSS` : a framework for parallel SAT solving. In *Proceedings of the 20th International Conference on Theory and Applications of Satisfiability Testing (SAT'17)*, volume 10491 of *Lecture Notes in Computer Science*, pages 233–250. Springer, Cham, August 2017.
- [2] Ludovic Le Frioux, Souheib Baarir, Julien Sopena, and Fabrice Kordon. Modular and efficient divide-and-conquer SAT solver on top of the `Painless` framework. In *Proceedings of the 25th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'19)*, volume 11427 of *Lecture Notes in Computer Science*, pages 135–151. Springer, Cham, April 2019.
- [3] Saeed Nejati, Ludovic Le Frioux, and Vijay Ganesh. A machine learning based splitting heuristic for divide-and-conquer solvers. In *Proceedings of the 26th International Conference on Principles and Practice of Constraint Programming (CP'20)*, volume 12333 of *Lecture Notes in Computer Science*, pages 899–916. Springer, Cham, September 2020.
- [4] Vincent Vallade, Ludovic Le Frioux, Souheib Baarir, Julien Sopena, Vijay Ganesh, and Fabrice Kordon. Community and LBD-based clause sharing policy for parallel SAT solving. In *Proceedings of the 23rd International Conference on Theory and Applications of Satisfiability Testing (SAT'20)*, volume 12178 of *Lecture Notes in Computer Science*, pages 11–27. Springer, Cham, June 2020.
- [5] Vincent Vallade, Ludovic Le Frioux, Souheib Baarir, Julien Sopena, and Fabrice Kordon. On the usefulness of clause strengthening in parallel SAT solving. In *Proceedings of the 12th NASA Formal Methods Symposium (NFM'20)*, volume 12229 of *Lecture Notes in Computer Science*, pages 222–229. Springer, Cham, August 2020.

1. <http://www.satcompetition.org/>