

**Frédéric MALLET**

☎ (+33) 659235482

✉ [Frederic.Mallet@univ-cotedazur.fr](mailto:Frederic.Mallet@univ-cotedazur.fr)

🌐 <https://www.i3s.unice.fr/~fmallet/>

**CAC UCA**

*Université Côte d'Azur*

5 novembre 2021

Madame, Monsieur,

Pour faire suite à ma candidature à la direction du laboratoire I3S, je vous prie de bien vouloir trouver ci-joint mon curriculum vitae, le résumé de mon activité et mon projet de recherche. Une liste exhaustive de mes publications est disponible en ligne sur: <https://www-sop.inria.fr/members/Frederic.Mallet/cv.html>.

Très sincèrement,

**Frédéric MALLET**

*Attached: curriculum vitae*

# Frédéric MALLET

## Curriculum Vitae

☎ (+33) 659235482

✉ Frederic.Mallet@univ-cotedazur.fr

🌐 <https://www.i3s.unice.fr/fmallet/>

### Education

- 2010 **HdR**, Habilitation à Diriger des Recherches, *Université Nice Sophia Antipolis*.  
1997–2000 **PhD**, Thèse de doctorat, *Université Nice Sophia Antipolis*.  
bourse DGA/CNRS  
1994–1997 **Ing.**, Ingénieur, *École Supérieure en Sciences Informatique*.

### Carrière

- 2014–Present Professeur des Universités, Université Côte d'Azur (1ère classe)  
2014–2015 Professeur Invité, East China Normal University, Shanghai, Chine  
2007–2009 Délégation, EPI Aoste, Inria Sophia Antipolis Méditerranée  
2003–2013 Maître de Conférences, Université Nice Sophia Antipolis  
2001–2003 *Research Fellow*, Edinburgh University, Royaume Uni  
2000–2001 1/2 ATER, Université Nice Sophia Antipolis

### Responsabilités pédagogiques et collectives

#### Établissement

- 2020–present **Directeur**, DÉPARTEMENT DISCIPLINAIRE D'INFORMATIQUE *regroupant les départements informatiques des composantes DS4H, IUT, Polytech*.  
Président du Comité de Pilotage Département Disciplinaire Informatique  
Bureau, Comité de Pilotage & Comité Scientifique et Pédagogique EUR DS4H  
2017–2021 **Directeur Adjoint**, I3S, UMR 7271 CNRS. *Dir : O. Meste*.  
Conseil Laboratoire et Comité de Direction I3S  
Commission ressources humaines CNU 27  
Commission égalité femmes-hommes & commission développement durable  
2017–present **Administrateur**, CA PÔLE DE COMPÉTITIVITÉ SCS.  
Membre du bureau depuis 2019  
2016–2017 **Adjoint Directeur**, I3S, UMR 7271 CNRS. *Dir : M. Riveill*.  
2015–2018 **Responsable M1 International**, UFR SCIENCES.  
2013–2014 **Directeur Adjoint**, DÉPARTEMENT INFORMATIQUE, UFR SCIENCES.  
2009–2014 **Élu**, CT, CTP, CHSCT, *Université Nice Sophia Antipolis*.  
2009–2013 **Responsable M1 MIAGE**, DÉPARTEMENT INFORMATIQUE, UFR SCIENCES.  
2004–2007 **Responsable L3 MIAGE**, DÉPARTEMENT INFORMATIQUE, UFR SCIENCES.

#### Nationales

- 2019–2022 **Élu PR, Section 27**, CONSEIL NATIONAL DES UNIVERSITÉS.  
2015–2018 **Élu PR, Section 27**, CONSEIL NATIONAL DES UNIVERSITÉS.  
2011–2014 **Élu MCF, Section 27**, CONSEIL NATIONAL DES UNIVERSITÉS.

## Projets collaboratifs récents

- 2020–2024 Responsable équipe associée PLoT4IoT avec ECNU et Jiaotong Shanghai.
- 2020–2024 ANR Smart IoT for Mobility
- 2020–2022 PSPC Région Adavec
- 2019–2021 PHC Cai Yuan Pei, Shanghai
- 2019–2023 CIFRE Renault Software Labs
- 2018–2020 National Science Foundation in China (NSFC)
- 2018 PHC Xu Guanqi, East China Normal University, Shanghai

## Expertises

- 2021 Membre comité experts HCERES : LIUPPA, LabSTICC
- 2020–2021 Expert externe AAPG ANR
- 2019–2021 Expert projets CIFRE ANRT
- 2019 Expert externe ERC Starting Grant
- 2019–2020 Expert projets FNRS Belgique

## Prix récents

- 2019–2020 Young Talent Program Zhejiang Province in China
- 2019 Lauréat Young Talent Program Campus France France-Chine
- FDL 2018 Best Paper Award
- FACS 2016 Best Paper Award

## Direction de thèses de doctorat

### Soutenues

- 2007–2010 Aamir Mehmood Khan, 50%, Model-Based Design for On-Chip Systems : using and extending Marte and IP-Xact.
- 2010–2013 Carlos Ernesto Gomez-Cardenas, 50%, Modeling Functional and Non-Functional Properties of Systems Based on a Multi-View Approach.
- 2011–2016 Ling YIN, 30%, Modeling, analysis, and verification for Cyber-Physical Systems based on clock and signal constraints.
- 2013–2016 Matias Vara-Larsen, 50%, BCoOL : The Behavioral Coordination Operator Language.
- 2014–2019 Yuanrui ZHANG, 33%, Spatio-Temporal Logic for Trustworthy Transportation Systems.
- 2018–2021 Giovanni LIBONI, 50%, Complex CPS Co-Simulation With The CoSim20 Framework For Efficient And Accurate Distributed Co-Simulations, CIFRE Safran.

### En cours

- 2016–2021 Hui ZHAO, 50%, System Engineering Integration of Cyber Physical Systems, Clarity + Labex.
- 2019– Joelle ABOU-FAYSAL, 100%, Langage de Règles pour Véhicules autonomes sûrs, CIFRE Renault Software Labs.
- 2020– Ming HU, 25%, Learning Safety Rules, NSFC-CSC.
- 2021– Pavlo TOKARIEV, 100%, Abstract Interpretation for Logical Clocks, Bourse CORDI-S.

## Implication dans les enseignements et formation par la recherche

J'ai un service de 192h. Le département est sous-encadrement donc j'essaye de faire des heures pour aider au maximum, mais je n'ai pas le droit de dépasser trop à cause de la prime de charge administrative (PCA) et pour assurer mes fonctions correctement.

### Licence

- 2015-2021 **L3 Miage**, *Conception et Programmation Orientée Objet*, 54h, CM/TD/Exam.  
Introduction à la programmation objet avec le langage Java
- 2015-2021 **L3 Informatique**, *Architecture des Machines*, 27h, TD.  
Logique, algèbre de Boole, architecture des machines et assembleur

### Master

- 2015-2019 **M1 Informatique**, *Safety Critical Systems*, 32h, CM/TP.  
Introduction la conception de systèmes critiques avec Scade. En anglais
- 2015-2021 **M1 Informatique**, *Introduction à la programmation synchrone*, 32h, CM/TP.  
Initiation aux langages synchrones et techniques de vérification associée : lustre, esterel
- 2015-2021 **M1 Informatique**, *Software Engineering*, 32h, CM/TP.  
Techniques avancées de conception du logiciel et introduction à l'ingénierie dirigée par les modèles.  
En anglais

### Tutorat et mineures

- 2015-2021 EUR DS4H. Coordonnateur d'une mineure sur les enjeux de l'Internet des Objets : capteurs et solutions efficaces en énergie, middleware pour l'interopérabilité, la sécurité, exploitation des données et ses enjeux

### Autres

- 2015-2020 Intervention dans le DIU de préparation des professeurs de lycée pour la spécialité Numérique et Science Informatique (NSI).
- 2019-2021 Coordonne un module pour le Master DataScience sur l'introduction à l'algorithmique avec Python

## Résumé des activités de recherche des 5 dernières années

Mes activités de recherche s'appuient largement sur l'utilisation du temps logique multiforme et des horloges logiques initialement introduites par Leslie Lamport pour la synchronisation de systèmes distribués, puis raffinées et étendues par les travaux sur les langages synchrones tels que Esterel, Lustre ou Signal. Nous avons défini le langage CCSL (Clock Constraint Specification Language) comme une partie intégrante du profil OMG Marte pour la conception des systèmes temps-réels et embarqués. Le profil Marte a été adopté par l'OMG (Object Management Group) en 2011 comme standard international et est depuis implanté dans de nombreux outils industriels. CCSL a été construit pour avoir une expressivité importante qui du coup amène des problèmes, encore ouverts, pour la résolution de certains systèmes de contraintes [10]. Dans certains cas [8], les spécifications CCSL peuvent être repliées avec une abstraction finie qui permet des analyses exhaustives par model-checking. Dans les autres cas, il faut souvent recourir à des heuristiques ou à des semi-algorithmes.

### Solveurs efficaces pour le langage CCSL

Une partie de mon activité de ces 5 dernières a consisté à construire des méthodes et des outils efficaces pour résoudre des systèmes de contraintes CCSL en lien avec des domaines d'application dédiés. La version courante s'appuie sur une combinaison de solveurs SMT (Satisfiability Modulo Theory), comme Z3 de Microsoft, et d'automates symboliques [10]. Une logique dynamique définit récemment permet de construire des preuves semi-automatiques [2].

- Une étape importante a été de définir les théories efficaces pour une résolution partielle par un solveur SMT. La pénalité la plus importante provenait de l'utilisation de quantificateurs existentiels que nous sommes arrivés à éliminer au prix d'une perte de précision et donc avec un problème sur-constraint [6].
- Dans la phase d'élicitation des exigences, il est particulièrement difficile de trouver les bonnes contraintes qui vont donner des résultats efficaces. Nous avons développé une méthode originale pour l'élicitation d'exigences à base de patrons [4]. Nous avons aussi travaillé sur l'expressivité du langage pour s'approcher d'exigences exprimées en langage naturel [5].
- Dans la troisième phase, nous complétons la partie automatique avec une logique dynamique qui permet de faire des preuves assistées sur les systèmes de contraintes pour lesquels la preuve automatique n'est pas possible. Dans cette logique, les formules sont dérivées de relations CCSL alors que les programmes sont des programmes synchrones impératifs largement inspirés d'Esterel [2]. Nous avons alors un système de preuves interactives qui nous permet de vérifier si un programme réactif donné satisfait un ensemble de propriétés temporelles. Comme nous utilisons là qu'un sous-ensemble de CCSL, nous nous restraignons à un sous-ensemble décidable de relations qui nécessitent seulement l'utilisation de théories SMT décidables et efficaces. Le solveur SMT permet d'établir l'ensemble des règles opérationnelles tirables à un instant donné.

Pour mesurer l'efficacité de ces solveurs nous avons suivi deux approches complémentaires :

- Le premier est l'ordonnancement temps-réel paramétrique, avec des systèmes de tâches concurrentes dont les périodes et échéances présentes une incertitude importante. L'élasticité du temps logique nous permet de remplacer des dates absolues par des intervalles de valeurs. Cela aggrandit rapidement l'espace des solutions mais nous arrivons malgré tout à traiter des cas d'études non triviaux [3]. Cette extension de CCSL peut être vue comme une forme de temps logique paramétrique.
- Dans un second temps, nous avons considéré des grosses spécifications mais qui sont générées automatiquement en appliquant une série de patron prédéfinis. Cette caractérisation est importante pour comprendre ce qui fait qu'une spécification CCSL est difficile ou pas à résoudre. Cette technique s'accompagne d'une méthode automatique qui s'appuie sur de l'apprentissage renforcé pour augmenter une spécification CCSL à partir de traces d'exécutions issues d'un système réel [1].

## Un langage formel pour la spécification de contrats intelligents sûrs

Les "smart contracts" sont des programmes mémorisés sur des registres distribués (blockchain). Ils doivent garantir que les règles d'un contrat sont respectées. Comme ils sont inscrits dans des chaînes de blocks, ils ne sont plus modifiables ni annulables. En conséquence, ils nécessitent une attention particulière de conception proche des méthodes que nous utilisons pour la conception de systèmes critiques. Nous nous sommes donc lancé dans la définition d'un projet ANR, démarré en janvier 2020, pour utiliser ces techniques sur les smart contracts.

Il est apparu rapidement que l'ordonnancement des différentes transactions d'un smart contrat est particulièrement important pour garantir la correction fonctionnelle. Nous utilisons le temps logique et une extension de CCSL pour spécifier les critères d'ordonnancement. Ce travail multi-disciplinaire demande une certaine pédagogie car nous travaillons avec des électroniciens, des économistes et des juristes qui ne sont pas familiers avec nos modèles et nos méthodes. Nous devons donc les dissimuler derrière des langages de spécification dédiés adaptés à nos usagers d'une façon similaire à celle que nous avons utilisée dans un projet antérieur [5].

Les partenaires industriels impliqués sont Symag, une société qui commercialise des solutions logicielles pour la conception de smart contracts et Renault Software Factory qui veut intégrer des chaînes de blocks dans ses véhicules pour avoir plus de transparence dans la chaîne d'entretien du véhicule.

Il y a un doctorant (Enlin ZHU) et une post-doctorante (Ankica Barisic) qui ont été recrutés dans notre équipe pour ce projet.

## Une extension stochastique et probabiliste du temps logique : pCCSL

Par nature une spécification CCSL s'appuie sur des horloges logiques qui sont par nature élastiques. Cela donne une grande flexibilité dans la spécification et une capacité de raffinement potentiellement infinie. Le processus naturel est de commencer avec peu de contraintes et donc beaucoup de flexibilité puis de raffiner petit à petit avec parfois des ensembles de solutions infinis difficiles à caractériser. Une façon de réduire cet espace de solutions est d'utiliser des taux probabilistes d'occurrences des événements non contrôlables du système. Nous avons travaillé notamment sur la définition d'une notion de précédence stochastique dont l'avance est décrite par une loi de distribution et d'une notion de sous-horloge probabiliste dont la coïncidence dépend d'une probabilité et d'un taux d'activation. Sur le papier, la définition sémantique est simple, les méthodes de résolution elles ne le sont pas, mais nous avons obtenu un résultat intéressant en utilisant des outils de model-checking statistiques [7]. Notre objectif maintenant est de travailler à une extension minimale qui puisse se traduire par une solution déterministe et ne soit pas le résultat de simulations aléatoires.

Cette tâche est l'action centrale de l'équipe associée que je mentionne par ailleurs et dont je suis le responsable scientifique. Ce travail a largement été ralenti en 2020 et 2021 à cause de la crise sanitaire et du ralentissement des collaborations avec nos partenaires chinois. Mais nous avons obtenu récemment un résultat intéressant en remplaçant les lois de distribution par des intervalles [3]. Une loi de distribution discrète peut également être le moyen de caractériser un intervalle de valeurs.

## Application : Sûreté de fonctionnement des transports intelligents

Une application récente de mon travail consiste à utiliser le temps logique pour définir des contraintes qui allient temps et espace afin de décrire les règles de sûreté de fonctionnement d'un véhicule autonome. En effet, l'algorithme d'assistance à la conduite est largement basé sur des règles d'apprentissage dont il est difficile, voire impossible d'assurer la validité avec certitude. Il est donc nécessaire de définir un enveloppe comportementale sûre qui définit les conditions suffisantes pour assurer l'intégrité du véhicule et des passagers. Ce comportement dépend à la fois de contraintes spatiales et temporelles élastiques qui dépendent largement des conditions extérieures (visibilité, trafic, densité).

En collaboration avec Renault Software Factory nous définissons un langage formel qui permet d'exprimer à la fois les suppositions de fonctionnement du véhicule et les exigences de sûreté attendues. Ce langage permet de décrire des règles qui permettent de valider ou invalider un ensemble de trajectoires envisagées.

Ce travail fait l'objet d'une thèse CIFRE et d'un contrat d'accompagnement.

La sémantique du langage s'appuie sur des travaux antérieurs qui définissent une logique dynamique pour la vérification de spécifications CCSL. Ce langage s'applique à des systèmes de transports intelligents au sens large, en incluant le système européen de contrôle des trains [9].

## 10 références choisies

- [1] Ming HU et al. "Enumeration and Deduction Driven Co-Synthesis of CCSL Specifications Using Reinforcement Learning". In : *42nd IEEE Real Time Systems Symposium (RTSS 2021)*. Déc. 2021.
- [2] Yuanrui ZHANG et al. "A clock-based dynamic logic for the verification of CCSL specifications in synchronous systems". In : *Sci. Comput. Program.* 203 (2021), p. 102591. URL : <https://hal.inria.fr/hal-03135428>.
- [3] Fei GAO et al. "Modeling and Verifying Uncertainty-Aware Timing Behaviors using Parametric Logical Time Constraint". In : *2020 Design, Automation & Test in Europe Conference & Exhibition, DATE 2020*. Grenoble, France : IEEE, mar. 2020, p. 376-381. URL : <https://hal.archives-ouvertes.fr/hal-02429533>.
- [4] Ming HU et al. "Sample-Guided Automated Synthesis for CCSL Specifications". In : *Proceedings of the 56th Annual Design Automation Conference 2019, DAC 2019*. Las Vegas, NV, USA : ACM, juin 2019, p. 98. ISBN : 978-1-4503-6725-7. URL : <https://hal.inria.fr/hal-02402971>.
- [5] Aamir M. KHAN, Frédéric MALLET et Muhammad RASHID. "A framework to specify system requirements using natural interpretation of UML/MARTE diagrams". In : *Software and System Modeling* 18.1 (2019), p. 11-37. URL : <https://hal.inria.fr/hal-01670423>.
- [6] Min ZHANG et al. "SMT-Based Bounded Schedulability Analysis of the Clock Constraint Specification Language". In : *Fundamental Approaches to Software Engineering, FASE 2019*. T. 11424. Lecture Notes in Computer Science. Prague, Czech Republic : Springer, avr. 2019, p. 61-78. URL : <https://hal.inria.fr/hal-02080763>.
- [7] Dehui DU et al. "pCSSL : a Stochastic Extension to MARTE/CCSL for Modeling Uncertainty in Cyber Physical Systems". In : *Science of Computer Programming* 166 (2018), p. 71-88. ISSN : 0167-6423. URL : <https://hal.inria.fr/hal-01898202>.
- [8] Min ZHANG, Feng DAI et Frédéric MALLET. "Periodic scheduling for MARTE/CCSL : Theory and practice". In : *Science of Computer Programming* 154 (mar. 2018), p. 42-60. URL : <https://hal.inria.fr/hal-01670450>.
- [9] Y. BAO et al. "Quantitative Performance Evaluation of Uncertainty-Aware Hybrid AADL Designs Using Statistical Model Checking". In : *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 36.12 (déc. 2017), p. 1989-2002. ISSN : 0278-0070. URL : <https://hal.inria.fr/hal-01644285>.
- [10] Frédéric MALLET et Robert de SIMONE. "Correctness issues on MARTE/CCSL constraints". In : *Sci. Comput. Program.* 106 (2015), p. 78-92. URL : <https://hal.inria.fr/hal-01257978>.

Une liste exhaustive est disponible en ligne : <https://www.i3s.unice.fr/~fmallet/publis/>