

Initiation à la recherche

Équipe LMV et parcours

Jolan PHILIPPE

17 Novembre 2025

Université d'Orléans

L'équipe LMV in a nutshell

Historique

- Création en **2015**, au sein du **LIFO** (Université d'Orléans).
- Initiée par **Frédéric Loulergue**, dans le contexte de la **fiabilité des logiciels embarqués**.
- Objectif : **concilier conception et vérification formelle** dans le développement logiciel.



LMV aujourd’hui : membres permanents

- 2 Professeurs
- 6 Maîtres de conférences
- Domaines couverts : **Langages, Modèles, Vérification**

LMV aujourd’hui : membres non permanents

- 2 Post-doctorants
- 4 Doctorants
- 2 étudiants de Master

Et dans le passé ...

- 10 thèses terminées
- 3 permanents partis à la retraite
- 3 étudiants de Master **dont 3 qui ont continué en thèse**

Objectifs scientifiques de l'équipe

- Garantir la **fiabilité** des logiciels.
- Agir à deux niveaux :
 - **Par construction** (langages et modèles sûrs).
 - **A posteriori** (vérification formelle, preuves, model-checking).



LMV = **Languages**, Models and Verification

Languages

Exprimer des systèmes avec des **langages bien structurés**.

- Sémantique formelle (définir précisément ce que “signifie” un programme).
- Systèmes de types pour éviter des classes d’erreurs.
- DSL avec différents niveaux d’expressivité

LMV = Languages, **Models**, and Verification

Models

Représenter un système par une **abstraction** maîtrisée.

- Différents types : Models de coût; models de language; IDM; etc.
- Permet de raisonner, transformer, générer du code.

LMV = Languages, Models and **Verification**

Verification

- Vérifier \neq tester : on **prouve** que le système est correct.
- *forall ... respecting ... then ...*
- Dijkstra : “ *Testing can show the presence of bugs, not their absence.* ”
- Preuve mécanisée (Rocq), model checking (Maude), analyse statique.

Contributions techniques majeures

- **SyDPaCC** : développement **prouvé** de programmes parallèles (Coq).
- **SailL** : langage réactif **sûr**, conçu pour l'IoT.
- **VRAC** : vérification d'assertions lors de l'exécution.
- **BSML** : bibliothèque OCaml pour le **parallélisme BSP**.
- **PySke** : squelettes algorithmiques en Python (parallélisme distribué).

La recherche dans LMV

Professeur, fondateur de l'équipe.

- Programmation **parallèle BSP**.
- Développement **certifié** par Coq.
- **BSML, SyDPaCC**.

Professeur.

- **Model-checking**, logique temporelle.
- Vérification de systèmes concurrents.
- Algo de vérification *on-the-fly* (référence).

Maître de conférences.

- Calcul parallèle **fonctionnel** vérifié.
- Squelettes algorithmiques (**OSL**).

Maître de conférences.

- Réécriture de termes et **automates d'arbres** pour la sécurité.
- Vérification par approximation.

Maître de conférences HDR.

- Systèmes **réactifs synchrones**.
- Preuves + analyse statique.

Maître de conférences.

- **Logique linéaire**, réseaux de preuves.
- Vérification de langages réactifs (Coq).

Maître de conférences.

- Co-auteur de la **norme ISO Prolog**.
- Spécifications exécutables.

Maître de conférences depuis Sept 2025.

- **Big Data** et traitement en distribué.
- **Ingénierie dirigée par les modèles (MDE)**.
- **Modèles de reconfiguration** (systèmes répartis / cloud).
- **Espace de configuration**.

Les thèses en cours

Flo Groult

- Vérification de modules critiques écrits en RUST de systèmes d'exploitation pour l'internet des objets

Téo Bernier

- Méthodes formelles intégrées pour la vérification de logiciels critiques

Jérémy Damour

- Formalisation d'un modèle mémoire à région pour le langage C

Terence Clastre

- Preuve déductive de programmes réactifs en Sall

Mon parcours (plutôt atypique)

À l'Université d'Orléans (2013-2017).

- Licence Informatique
- Master 1 MIAGE

À la Northern Arizona University (2017–2019)

- Research Assistant: vérification + parallélisme.
- Master of Science: écriture d'une première thèse

À l'IMT Atlantique de Nantes (2019-2022)

- Transformation de modèles distribuée.
- Vérification de sémantique de moteur de transformation
- Analyse de variabilité des outils de “model management”

À l'IMT Atlantique de Nantes puis à l'Université de Rennes
(2023–2025)

- Reconfiguration de systèmes distribués.
- Décentralisation, programmation par contraintes.

- Qualification CNU 27 (2024).
- Recrutement MCF à l'Université d'Orléans (2025).

Travaux de recherche: Correction dans les systèmes distribués, de l'usage à la reconfiguration

Slides d'un séminaire donné en Janvier 2025