Curriculum Vitae

Name : Erwan Mahe   **Date of birth** : 1993.08.15
Nationality : French   **Thesis defense** : 2021.07.15
Current status : postdoctoral researcher at CEA Saclay

✉ erwan.mahe.contact@gmail.com
in linkedin.com/in/maheerwan
⌂ github.com/erwanM974
iD orcid.org/0000-0002-5322-4337

Research

In the context of my thesis I proposed a **formal language** to model the behavior of **distributed systems** that communicate via the discrete asynchronous exchange of messages. Models are elements of an **algebra of interaction terms** and associated to a **trace semantics** (set of sequences of discrete communication events that may occur at execution) expressed in **denotational-style** via inductive composition using some algebraic operators on sets of traces. Properties of those operators can help us identify equivalent interaction terms i.e. terms that are syntactically different but have the same semantics. Equivalent interaction terms are members of the same **equivalence class** and for any such class one can compute a unique **normal form** representative via **term rewriting**. With the aim of developing **formal verification** techniques using such interaction models I also proposed an **operational-style** semantics inspired from **process calculus** which I proved equivalent to the former using the **Coq theorem prover**. I devised several algorithms for analysing observations of the executions of distributed systems against specifications written as interaction terms. Variants of those **multi-trace analysis** algorithms allow the identification of **partially observed** executions. Those algorithms may either rely on simulation steps so as to replace missing (unobserved) events or hiding steps which consist in erasing parts of the interaction model corresponding to sub-systems that are no longer observed.

Education

2021 2018	**PhD in computer science** Université Paris-Saclay - Interfaces doctorate school - financed as part of the DisTA FUI supervised by *Pascale Le Gall* (CentraleSupélec) & *Christophe Gaston* (CEA)
2017 2014	**Diplôme d'ingénieur de l'École Centrale de Paris** - *Master of Science in Engineering* selective postgraduate degree in the French "Grandes Écoles" higher education system.
2014 2011	**Classe Préparatoire aux Grandes Écoles (CPGE) filière MP** - *Prep school in mathematics & physics* Lycée Leconte de Lisle (*Reunion Island, France*) & Champollion (*Grenoble, France*)

Work experience

2018 2017	**Research engineer** Université Paris-Saclay, MICS laboratory
2017	**Internship** - Master's degree internship (*software engineering* R&D - 6 months) Ericsson Hungary (*Budapest, Hungary*)
2017 2014	**Apprenticeship** at CEA (French Alternative Energies and Atomic Energy Commission) - *research institute* • LISE laboratory (laboratory for model-based engineering of real-time and embedded systems) *Palaiseau, France* • LLPR laboratory (laboratory of numerical methods for reactor studies) *Saclay, France*

Academic experience

Publications

• **A structural operational semantics for interactions with a look at loops**
E. Mahe, C. Gaston & P. Le Gall
Preprint `arXiv:2105.00208`
• **A small-step approach to multi-trace checking against interactions**
E. Mahe, B. Bannour, C. Gaston, A. Lapitre & P. Le Gall
ACM/SIGAPP Symposium on Applied Computing - Software Verification & Testing Track
SAC-SVT 2021 `DOI:10.1145/3412841.3442054`
• **Revisiting Semantics of Interactions for Trace Validity Analysis**
E. Mahe, C. Gaston & P. Le Gall
European Joint Conferences on Theory & Practice of Software - Fundamental Approaches to Software Engineering
ETAPS-FASE 2020 `DOI:10.1007/978-3-030-45234-6_24`

Teaching

- **Algorithmics and Complexity**
L3-M1 course in computer science - *labs & exercise classes*
CentraleSupélec 2020-2021 - 30 hours per year
- **Game theory**
L3-M1 course in applications of game theory in computer science - *labs & supervising student projects*
CentraleSupélec 2019-2021 - 40 hours per year
- **Programming in Python**
L3 course for learning programming with the Python language - *exercise classes*
CentraleSupélec 2018-2020 - 30 hours per year

Communications

- FASE-2021 on the 2021.03.30
- SVT-2021 on the 2021.03.22
- Journée MTV2 (GDR GPL CNRS) on the 2020.11.20

Organization

- co-chaired session at SVT-2021
- part of organising committee for IFIP-ICTSS-2019

Involvement in software projects

HIBOU ~ 2019-2021 ~ *main developer*
A tool (in Rust) for modelling the behavior of distributed systems as interactions (sequence-diagram-like models) with various utilities to manipulate those models & test the outputs of distributed systems (logs) w.r.t. specifying interactions.
`github.com/erwanM974/hibou_label` (main project)
An extension HIBOUX which adds data to interactions in the form of variables and which handles constraints via symbolic execution has been successfully used in industrial test cases as part of the DisTA project (Thales Group).
`github.com/erwanM974/hibou_efm` (extension)
Various Coq proofs for verifying the theoretical foundations of the tool as well as experiments are available on my github.

Eclipse-FMP / Diversity ~ 2016-2017 ~ *contributor*
A tool (in C++ for the core & Java for the GUI) for formal analysis based on symbolic execution. Diversity allows to model concurrent systems as sets of communicating Input Output Symbolic Transition Systems. Various techniques can then be applied e.g. the automatic generation of tests, the verification of some predicates, etc.
`projects.eclipse.org/projects/modeling.efm`
With Ericsson Hungary I also implemented a POC MBT toolchain for the IoT (devices using the CoAP protocol). TTCN-3 tests are generated by Diversity and launched by TITAN.
`projects.eclipse.org/projects/tools.titan`

OLYMPE2 ~ 2014-2016 ~ *developer*
A workflow engine (in Python) for orchestrating and distributing tasks via SSH to remote machines. Tasks correspond to Bash scripts that are to be executed via SSH using specific credentials and passwords on specific remote machines. Workflows are graphs which schedule the execution of tasks and sub-taks, enabling sequential and parallel compositions as well as alternatives. We used the Twisted library for the networking back-end and PyQT for the graphical user interface.

Skills

Natural languages

French	native speaker	**English**	proficient TOEFL-ITP 660/677	**German**	intermediate	**Chinese**	beginner

Programming languages & tools

OSs	Languages			Utilities
Windows	Python	Rust	JavaScript	office suites (Microsoft), LaTeX (Beamer, TikZ, etc.),
Linux	Java	Coq	TTCN	IDEs (IntelliJ IDEA, PyCharm, Eclipse, etc.), versioning (Git/SVN),
	R	Bash	C++	image processing (Inkscape, GIMP, Graphviz)