
Dossier de candidature à un poste de Professeur en Génie Logiciel au sein de l'Université du Québec à Montréal

présenté par Sébastien Mosser

`mosser@i3s.unice.fr`

`http://www.i3s.unice.fr/~mosser`

Mai 2018

Pièces fournies

1. Lettre de motivation
2. Curriculum vitæ détaillé
3. Projet de recherche
4. Projet d'enseignement
5. Publications récentes
 - (a) “*Assessing the Functional Feasibility of Variability-Intensive Data Flow-Oriented Systems*” (2018, 33th Symposium on Applied Computing, **Best Paper** pour le champ génie logiciel)
 - (b) “*A Delta-oriented Approach to Support the Safe Reuse of Black-box Code Rewriters*” (2018, 17th Int. Conf. on Software Reuse, co-écrit avec Naouel Moha, UQAM)
 - (c) “*Supporting Micro-services Deployment in a Safer Way : a Static Analysis and Automated Rewriting Approach*” (2018, 33th Symposium on Applied Computing)
 - (d) “*ADORE, a Logical Meta-model Supporting Business Process Evolution*” (2013, Revue *Science of Computer Programming*)

Références

- ✉ Jean-Marc Jézéquel, directeur du laboratoire IRISA, Rennes, France
- ✉ Jörg Kienzle, McGill University, Montréal, Canada
- ✉ Igor Litovsky, directeur du dépt. d'Informatique, Polytech Nice – Sophia Antipolis

Louise Laforest

Directrice, Département d'informatique
Université du Québec à Montréal
C.P. 8888, Succursale Centre-Ville
Montréal (Québec) H3C 3P8

Sophia Antipolis, le 16 Mai 2018

Chères et chers membres du comité de sélection,

Je souhaite par ce dossier porter à votre attention ma candidature au poste de Professeur en Génie Logiciel (profil "Systèmes Distribués & Architectures Logicielles") ouvert au sein de département d'Informatique de l'Université du Québec à Montréal. Je suis depuis 2012 Maître de Conférences en Génie Logiciel au sein du département d'Informatique de l'école d'ingénieur Polytech Nice – Sophia Antipolis, faisant partie de l'Université Nice Sophia Antipolis et de l'Université Côte d'Azur. Ma recherche est attachée à l'équipe SPARKS du laboratoire I3S (unité mixte CNRS), dans laquelle je participe à l'axe "Scalable Software Systems". Je suis particulièrement intéressé par l'obtention d'un poste de professeur à l'UQAM, qui est située dans un écosystème industriel et académique propice à une recherche et un enseignement de haut niveau en génie logiciel. De plus, les valeurs défendues par l'UQAM (par ex., "une université ancrée dans son milieu et ouverte sur le monde") correspondent à ma vision de l'enseignement supérieur.

Mes intérêts de recherche portent sur la modélisation et la séparation des préoccupations, appliquées à l'architecture des systèmes distribués à ultra-large échelle tels que les systèmes cyber-physiques ou le *cloud-computing*. Mes travaux précédents ont porté sur la définition d'opérateurs de composition logicielle dans des domaines aussi différents que les applications mobiles, les systèmes cyber-physiques, les interfaces homme-machines, et les applications distribuées déployées sur un *cloud*. Ma proposition de recherche, en adéquation avec les thématiques du département, se propose de capitaliser sur ces résultats (3 articles de revues et 35 articles de conférence internationales depuis 2007 selon l'index DBLP) en définissant un moteur de composition abstrait qui permettra l'application aisée du paradigme de séparation des préoccupations dans différents domaines, du point de vue du passage à l'échelle. Je porte un fort intérêt à la mise en œuvre d'une recherche en génie logiciel pragmatique et portée par l'écriture de logiciel et la publication d'implémentations de références sur des plateformes communautaires telles que GitHub par exemple. Fort de mon expérience directe avec plusieurs partenaires industriels français (représentant un tiers de ma recherche subventionnée), je souhaite aussi tisser des relations avec l'écosystème industriel montréalais pour permettre une validation empirique des travaux de développement menés dans le cadre de ces recherches. En effet, un travail de recherche en génie logiciel doit avant tout rendre service aux ingénieurs développant les logiciels, et il est crucial de pouvoir valider sur le terrain les résultats obtenus en laboratoire.

Mes activités d'enseignement portent sur le génie logiciel au niveau licence, et au niveau maîtrise sur (i) les architectures logicielles (particulièrement les architectures à base de micro-services et la mise en œuvre du principe de "développement & exploitation" en lien avec la modélisation d'une architecture logicielle) et (ii) les langages spécifiques aux domaines. En tant que responsable des projets dans mon école d'ingénieur, j'ai eu l'occasion, en lien avec les industriels locaux, de mettre en place des méthodes d'enseignement originales basées sur des techniques de facilitation de projet et des jeux sérieux, qui ont donné lieu à plusieurs présentations académiques et industrielles en lien avec l'agilité. En licence, j'ai aussi développé pour le cours de génie logiciel une "arène de jeu" permettant la mise en place d'un championnat hebdomadaire entre les différentes équipes étudiantes suivant ce cours, ce qui permet aux étudiants de mesurer leurs progrès au fil du semestre et de se positionner au sein de la promotion. Je trouve important de mettre à disposition des étudiants des enseignements théoriques forts et indépendants de toute technologie, accompagnée de code de mise en œuvre de référence disponible librement sur GitHub en support des travaux pratiques. Mes enseignements comprennent une grande part de mise en œuvre au niveau du code, car il est important que des enseignements même abstraits (par ex., méta-modélisation) soient ancrés dans la réalité du terrain. En termes d'encadrement, j'ai supervisé 6 étudiants de maîtrise et co-supervisé 6 thèses de doctorat (dont 3 soutenues).

J'ai réellement apprécié l'exercice d'écriture de ce dossier, qui m'a permis de prendre du recul sur mon travail de recherche et d'enseignement. Dans l'attente de votre réponse, je me tiens à votre disposition pour tout renseignements complémentaires.

Veuillez agréer, Mesdames, Messieurs, mes salutations distinguées.



Sébastien Mosser

Sébastien Mosser

Université Côte d'Azur – Polytech Nice – Sophia Antipolis – Laboratoire I3S (CNRS)
☎ +33 489 154 280 • ✉ mosser@i3s.unice.fr • 🌐 www.i3s.unice.fr/~mosser

Sébastien Mosser

Maître de Conférences en Génie Logiciel

Université Côte d'Azur
École d'Ingénieur Polytech
Laboratoire I3S (UMR CNRS 7271)
☎ +33 489 154 280
✉ mosser@i3s.unice.fr
🌐 www.i3s.unice.fr/~mosser

Formation

- 2007 – 2010 **Doctorat en Informatique**, Université Nice – Sophia Antipolis (FR).
○ Titre : *Behavioral Compositions in Service-oriented Architectures*
○ Encadrement : Michel Riveill & Mireille Blay-Fornarino
○ Rapporteurs : Don Batory (UT Austin, USA) & Xavier Blanc (Univ. Bordeaux, FR)
○ Jury : Lionel Seinturier (Univ. Lille, FR) & Pierre-Alain Muller (Univ. Mulhouse, FR)
- 2004 – 2007 **Diplôme d'Ingénieur CTI**, Université Nice – Sophia Antipolis (FR).
○ Majeure : Systèmes Distribués, Mineure : Génie Logiciel (rang : 1)
○ École d'ingénieur Polytech Nice – Sophia Antipolis (anciennement ESSI)
○ Titre reconnu par le CEAB et l'OIQ pour admission au sein de l'Ordre des Ingénieurs

Emplois académiques

- Depuis 2012 **Maître de Conférences**, Université Côte d'Azur (FR).
○ Enseignement: Polytech Nice – Sophia, Département Science Informatique
- Membre élu du conseil de département depuis 2014
- Coordination des enseignements de génie logiciel depuis 2014
- Coordination de la majeure "Architecture Logicielle" depuis 2013
- Coordination des projets depuis 2012
○ Recherche : Laboratoire I3S (UMR CNRS 7271), équipe SPARKS.
- Membre nommé du conseil de laboratoire depuis 2018
- Ingénierie des Modèles, Langages spécifiques, Composition, Systèmes Cyber-Physiques
- Membre du groupe de recherche "Scalable Software Systems" (S³).
- 2011 – 2012 **Chargé de Recherche**, SINTEF ICT (NO).
○ Adaptation logicielle, Systèmes infonuagiques, Ingénierie des Modèles
○ Département: *Network & Secured Systems*, équipe MOD.
○ Travail sur projets: européens (ENVISION, REMICS) et nationaux (MODERATES)
- 2010 – 2011 **Chercheur post-doctoral**, Inria Lille–Nord Europe (FR).
○ Adaptation logicielle, Lignes de produits, Ingénierie des Modèles
○ Équipe-projet ADAM (maintenant SPIRALS), dirigée par Laurence Duchien
○ Financement : FUI (projet collaboratif grandes entreprises, PME et laboratoires)
- 2007 – 2010 **Moniteur (Assistant d'enseignement)**, Polytech Nice – Sophia (FR).
○ Direction des TPs de Programmation Web en licence : coordination de l'équipe, création des énoncés, suivi des étudiants, rédaction des quizzs de suivis et corrections
○ Encadement de TPs divers (*p. ex.*, prog fonctionnelle, ingénierie des modèles)

Primes & Distinctions

- 2015 – 2019 **Prime d'Encadrement Doctoral et de Recherche (PEDR)**, CNU 27.
Prime nationale accordée sur critères de recherche et d'encadrement doctoral (14K€)
- Prix du meilleur article
○ "Assessing the Functional Feasibility of Variability-Intensive Data Flow-Oriented Systems", S. Lazreg *et al*, 33th Symposium on Applied Computing (SAC'18 [10])

Recherche & Publications sélectionnées

Composition logicielle dirigée par les modèles

Mes intérêts de recherche portent sur la composition logicielle du point de vue du passage à l'échelle, en utilisant une approche dirigée par les modèles pour définir et valider des opérateurs de composition [22]. Cette approche a été appliquée avec succès à des domaines hétérogènes, comme les micro-services [12], les bases de données orientées graphes [14], les politiques de collecte de données [16] ou encore les tableaux de bords de visualisation de données [19].

Systèmes
Cyber-Physiques

Du point de vue de l'ingénierie dirigée par les modèles, cette classe de systèmes est intéressante pour les contraintes qui lui sont associée. Dans ce domaine, j'ai travaillé à la définition d'opérateurs de composition dédiés aux politiques de collectes de données en provenance de réseaux de capteurs à large échelle [15, 17, 36]. Ces recherches ont été mise en œuvre avec des chercheurs d'autres domaines (*p. ex.*, géo-sciences) pour les appliquer à des cas concrets.

Architecture
orientée services &
Infonuagique

Ma thèse de doctorat porte sur une approche dirigée par les modèles supportant la définition de processus d'affaires isolés et leur composition [8, 9]. J'ai appliqué par la suite dans mes différents emplois ces résultats à la qualité de service [20] et au déploiement de services infonuagiques [21]

Recherche sur projet & Subventions

Synthèse
(depuis 2011)

- Financement industriel direct : 225K€
- Financement doctoral direct : 300K€
- Financement institutionnel collaboratif : 346K€
- Financement Européen collaboratif : 27,4M€

2018–2020

I-WIN, *Financement UCA IDEX*, 36K€.

- Rôle : Responsable de lot (Composition logicielle & Systèmes Cyber-Physiques)
- Ce projet collaboratif (laboratoire d'informatique, d'électronique & Inria Méditerranée) a pour but le déploiement d'un réseau de capteurs reconfigurable, en utilisant des méthodes d'apprentissage automatique pour adapter le comportement des capteurs et atteindre une meilleure efficacité énergétique. Dans ce contexte, je suis responsable du lot portant sur la modélisation des applications consommant les données du réseau, en considérant la composition de ces applications avec le réseau physique.

2018–2020

SmartIoT for Mobility, *Financement UCA IDEX*, 25K€.

- Rôle : Expertise technique (Ingénierie des modèles & Systèmes Cyber-Physiques)
- Ce projet collaboratif inclue des chercheurs de trois laboratoires différents (département d'informatique, d'électronique et de droit) pour construire une équipe pluridisciplinaire liée aux contrats intelligents disponible sur une chaîne de blocs ("*smart contracts*" & "*blockchain*"), en lien avec l'internet des objets. Ce projet à petite échelle vise à mettre en place un démonstrateur permettant de dialoguer concrètement avec des partenaires industriels (*p. ex.* Renault, Cap Gemini).

2018–2020

Formalizing Scalable Composition Operators (FiaSCO), *I3S*, 5K€.

- Rôle : Chercheur Principal
- Ce projet finance un ensemble de missions en Europe et Amérique du Nord pour échanger avec des chercheurs seniors sur des sujets de composition logicielle. L'objectif est d'identifier sur la base d'exemples du terrain des abstractions communes entre les différentes approches rencontrées. Ce projet fait suite au projet M4S.

2017–2020

Composition d'application MERMAID, *PME & Région PACA*, 90K€.

- Rôle : Co-Chercheur Principal (avec Mireille Blay-Fornarino)
- Ce projet, en collaboration avec le laboratoire d'informatique, de géo-sciences et une PME spécialisé en matériel sous-marin, travaille sur le flotteur MERMAID ("*Mobile Earthquake Recorder in Marine Areas by Independent Divers*"). L'objectif est d'identifier comment modéliser à l'aide d'un langage spécifique les applications déployées sur le flotteur (*p. ex.*, détection de tremblements de terre, suivi de population de baleines, analyse de la salinité de l'océan). Ces applications, écrites par des scientifiques divers (*p. ex.*, biologistes, sismologues) sont ensuite composées avant d'être déployées sur les flotteurs pour être utilisée en campagne d'expérimentation de terrain en respectant des propriétés sur les flotteurs (*p. ex.*, une consommation de batterie raisonnable au regard de la durée de la campagne d'expérimentation).

2016–2020

Variabilité dans les Systèmes Cyber-Physiques, *Contrat industriel*, 77K€.

- Rôle : Co-Chercheur Principal (avec Philippe Collet)
- Dans ce projet, nous explorons le lien entre la modélisation de la variabilité et la conception de logiciel embarqué, en lien avec un partenaire industriel du domaine de la construction automobile (VISTEON, équipementier pour *p. ex.* Renault et Range Rover). Le projet exploite les "*Feature Transition Systems*" (FTS) de l'état de l'art pour identifier dans un pipeline embarqué (*p. ex.* un processeur GPU) comment les exigences définies par les équipes de marketing peuvent être mise en œuvre sur une plateforme logicielle embarquée donnée.

- 2016–2019 **Modélisation de la Composition Logicielle**, *Financement Doctoral*, 100K€.
- Rôle : Chercheur Principal
 - Ce projet explore comment l'ingénierie dirigée par les modèles peut aider à la définition d'opérateurs de composition logicielle. Nous utilisons des modèles d'actions pour représenter les compositions à effectuer, et définissons des briques logicielles réutilisables à l'échelle de plusieurs domaines (*p. ex.*, modèles de trace, bancs d'essais, ré-écrivains, tests de propriétés) pour supporter la définition d'opérateurs binaires.
- Projets terminés
- 2016 **Modeling for Scaling (M4S)**, *Programme Jeune Chercheur INS2I*, 10K€.
- Rôle : Chercheur Principal
 - Ce projet exploratoire “jeune chercheur” a permis une étude préliminaire des problématiques de composition logicielle à l'échelle de plusieurs domaines hétérogènes. L'objectif était de collecter les exigences nécessaires à la construction d'un moteur de composition abstrait auprès de chercheurs reconnus dans différents domaines.
- 2015 – 2018 **Transfert Technologique DEPOSIT**, *EIT Digital*, 33K€.
- Rôle : Co-Chercheur Principal (avec Philippe Collet)
 - L'*EIT Digital* est un acteur Européen majeur de l'entrepreneuriat et de l'innovation. Ce financement a permis une collaboration avec le centre de recherche SnT de l'Université du Luxembourg autour des systèmes cyber-physiques, et le transfert technologique des résultats de la thèse de Cyril Cecchinell dans la jeune pousse DataThings par la suite.
- 2014 – 2017 **Tailored Comp. for Large-scale Sensing Network**, *Financement Doctoral*, 100K€.
- Rôle : Co-Chercheur Principal (avec Philippe Collet)
 - Ce financement de thèse a permis la définition d'une approche basée sur les modèles pour représenter les applications déployées sur des grands réseaux de capteurs. Des raisonnements au niveau des modèles permettent à plusieurs applications de partager le même réseau de capteurs, par composition.
- 2013 – 2016 **Model-based Sensor Data Visualizations**, *Financement Doctoral*, 100K€.
- Rôle : Chercheur Principal (directeur habilité : Michel Riveill)
 - Ce financement de thèse a exploré comment les applications de type “tableaux de bords” pouvaient être définis au-dessus de réseaux de capteurs. L'idée clé de la thèse a porté sur l'application de mécanismes similaires aux principes d'intégration des Architectures Orientées Services appliquée aux différents langages spécifiques mis en oeuvre lors de la définition de ce type d'application.
- 2012 – 2014 **IDOL**, *EGIDE Aurora*, 20K€.
- Rôle : Co-Chercheur Principal (avec Arnor Solberg)
 - Le programme EGIDE est financé par le ministère des affaires étrangères français pour supporter des collaborations binationales. Plus spécifiquement, l'instrument AURORA permet des collaborations franco-norvégiennes. Ce projet a permis la mise en place d'ateliers de recherche et de présentation croisées entre SINTEF et l'Université Nice Sophia Antipolis en finançant des missions entre Oslo et Nice. Les thématiques abordées portaient sur la modélisation de lignes de produits appliquée aux systèmes infonuagiques.
- 2012 – 2014 **Mod4Cloud**, *Amazon Research Grant*, 25K€.
- Rôle : Chercheur Principal
 - Ce projet a portée expérimentale à été financé par le programme de support à la recherche d'*Amazon Web Services* (AWS). Il a permis la mise en place d'expériences de validation à moyenne puis large échelle sur le système infonuagique d'AWS, en lien avec le développement du canevas logiciel CloudML et du langage CloudScript associé.
- 2012 – 2014 **YourCast**, *ANR Émergence (Transfert Technologique)*, 250K€.
- Rôle : Expertise Technique (Composition logicielle)
 - Ce projet, financé par l'*Agence Nationale de la Recherche* dans sa mission de transfert technologique à destination de l'industrie, a permis la mise en production et exploitation commerciale du logiciel jSeduite. Nous avons utilisé des principes d'adaptation logicielle et de modélisation de lignes de produits pour permettre la conception d'une ligne de produits de systèmes de diffusion d'information (financement d'une thèse et d'un post-doctorat pour la partie recherche). Il a permis la création de la jeune pousse “*The 6th Screen*” pour l'exploitation du produit (2013–2016).
- 2011 – 2012 **MODAClouds**, *Programme EU FP7*, 8.7M€.
- Rôle : Resp. proposition pour SINTEF & Resp. de lot (jusqu'à mon départ de SINTEF)
 - Ce projet Européen dirigé par *Politecnico de Milano* impliquait 10 partenaires pour la définition d'abstractions permettant le développement aisé d'applications sur des systèmes infonuagiques. En tant que chargé de recherche SINTEF, j'ai co-dirigé l'écriture de la proposition de projet au niveau du laboratoire et coordonné les discussions autour du canevas logiciel CloudML dont j'avais la responsabilité. J'ai quitté le projet peu après son démarrage, à la suite de mon recrutement à l'Université Nice – Sophia Antipolis.

- 2011 – 2012 **PaaSage**, *Programme EU FP7*, 9.7M€.
- Rôle : Resp. proposition pour SINTEF & Resp. de lot (jusqu'à mon départ de SINTEF)
 - Ce projet Européen dirigé par l'ERCIM impliquait 15 partenaires et était dédié à la définition d'un environnement de modélisation pour les systèmes infonuagiques, dans le but de supporter l'interopérabilité des systèmes. J'étais responsable jusqu'à mon départ de l'intégration de CloudML dans l'environnement.
- 2011 – 2012 **REMICS**, *Programme EU FP7*, 4.5M€.
- Rôle : Expertise Technique
 - Ce projet était dédié à la rétro-ingénierie de systèmes patrimoniaux (écrits en COBOL) en utilisant une approche dirigée par les modèles. J'étais impliqué dans le lot dédié au déploiement, en travaillant sur des méta-modèles et des générateurs de code permettant le déploiement automatisé d'applications COBOL migrées en application infonuagique.
- 2011 – 2012 **ENVISION**, *Programme EU FP7*, 4.5M€.
- Rôle : Expertise Technique
 - Ce projet, à destination de chercheurs en science de l'environnement, a permis la publication de modèles basés sur des ontologies pour partager des expériences réalisées sur des réseaux de capteurs hétérogènes. J'étais impliqué dans le lot "Models as a Service", où nous avons utilisé le langage BPMN pour coordonner une couche de médiation entre plusieurs réseaux hétérogènes.

Supervision d'étudiants

- Synthèse (depuis 2011)
- Étudiants de maîtrise supervisés : 6 (1 en cours)
 - Étudiants de doctorat encadrés : 6 (2 encadrements, 4 co-encadrements, 3 en cours)

Encadrement doctoral

- Depuis 2017 **Sébastien Bonnieux**, *Université Côte d'Azur*.
- Développement d'une approche dirigée par les modèles pour la composition d'applications embarquées (co-supervisé Mireille Blay-Fornarino).
- Depuis 2016 **Benjamin Benni**, *Université Côte d'Azur*.
- Définition d'artefacts logiciels réutilisables pour la composition logicielle et son passage à l'échelle.
 - Publications : [11, 12, 34].
- Depuis 2016 **Sami Lazreg**, *Université Côte d'Azur*.
- Modélisation de la variabilité pour les systèmes embarqués (co-supervisé avec Philippe Collet).
 - Publications : [10].
- 2014 – 2017 **Cyril Cecchinell**, *Université Côte d'Azur*.
- Modélisation des applications à destination des réseaux de capteurs à large échelle (co-supervisée avec Philippe Collet).
 - Actuellement ingénieur de recherche chez *DataThings* (LU)
 - Publications : [15, 16, 17] (soumission : [1]).
- 2013 – 2017 **Ivan Logre**, *Université Côte d'Azur*.
- Visualisation de données en provenance de capteurs (directeur habilité : Michel Riveill)
 - Actuellement en post-doctorat à Université Grenoble - Alpes (FR)
 - Publications : [19, 40, 51] (soumission : [4]).
- 2010 – 2014 **Alexandre Feugas**, *Université Lille 1*.
- Évolution de Processus d'Affaires (co-supervisé avec Laurence Duchien)
 - Actuellement consultant R&D senior chez *F. Iniciativas* (FR)
 - Publications : [20, 55].

Encadrement de maîtrise

- 2017–2018 **Günther Jungbluth**, *Université Côte d'Azur*, (CNRS apprentice program).
- Modélisation de flux de production de fouille de données & passage à l'échelle.
 - Publications : [35].
- 2016 **Benjamin Benni**, *Université Côte d'Azur*.
- Une approche langage pour la composition de modèles de variabilité logicielle
- 2014 **Cyril Cecchinell**, *Université Nice – Sophia Antipolis*.
- Une approche générative pour les applications orientée utilisateurs sur les réseaux de capteurs à large échelle
 - Publications : [36].
- 2013 **Ivan Logre**, *Université Nice – Sophia Antipolis*.
- Modélisation de visualisation de données en provenance de capteurs

- 2011 **Eirik Brandtzæg**, *Universitetet i Oslo*.
◦ *CloudML, A DSL for model-based realization of applications in the cloud*
◦ Publications: [21, 44].

Enseignements

- Depuis 2014 **Architecture Logicielle, Développement & Exploitation**, Maîtrise.
◦ Site : <https://github.com/mosser/isa-devops/>
Ce cours est conçu en collaboration avec *IBM France Labs* pour créer un continuum entre architecture logicielle et exploitation. Il exploite une approche basée sur un projet de développement, associé à une implémentation de référence disponible en exemple pour les étudiants. L'objectif est de modéliser et implémenter une architecture n-tiers, avec un pipeline de déploiement continu associé et dimensionné pour le projet.
- Depuis 2014 **Modélisation de processus d'affaires**, Maîtrise, Spécialité.
◦ Site : http://www.i3s.unice.fr/~mosser/teaching/17_18/bpm/start
Ce cours est dédié à la modélisation de processus d'affaires, et à leur analyse. En collaboration avec des partenaires industriels (*p. ex.* Air France), les étudiants modélisent des processus d'affaires et identifient des indicateurs de résultats permettant leur analyses. Ce cours implémente deux patrons pédagogiques : *l'échec productif* (les étudiants sont confrontés à des situations qu'ils ne peuvent résoudre, et le cours analyse les raisons de cet échec et donne les méthodologies a posteriori), et *l'enseignement entre pairs* où les étudiants expliquent aux autres groupes de projet les concepts de l'état de l'art qu'ils appliquent.
- Depuis 2013 **Projets Innovation**, Maîtrise.
◦ Démonstrations : <https://www.youtube.com/pnsinnov>
En lien avec le département de gestion de l'école, les projets innovation permettent aux étudiants de simuler la création d'une jeune pousse par équipe de quatre étudiants, durant un mois à temps plein. L'approche pédagogique repose sur des jeux sérieux et des techniques de facilitation pour accompagner les étudiants durant le développement d'un produit minimal viable innovant.
- Depuis 2012 **Architectures Orientées Services**, Maîtrise, Spécialité.
◦ Site : http://www.i3s.unice.fr/~mosser/teaching/17_18/esb/start
Ce cours s'intéresse aux Architectures Orientées Services, du point de vue de la modélisation, de leur implémentation et déploiement. Il travaille particulièrement sur les architectures de micro-services, et les défis d'intégration associés en utilisant des bus de services orientés flots de données ou événements. Ce cours est monté en collaboration avec deux PME qui exploitent cette classe d'architecture en production, donnant des retours industriels de terrain aux étudiants.
- Depuis 2012 **Ingénierie des Modèles & Langage Spécifique**, Maîtrise, Spécialité.
◦ Site : http://www.i3s.unice.fr/~mosser/teaching/17_18/dsl/start
Ce cours s'intéresse à l'ingénierie dirigée par les modèles du point de vue des langages spécifiques. On y utilise UML pour représenter un modèle du domaine, et des patrons de conceptions (*p. ex.* Visiteur, Observateur) sont mis en œuvre pour opérationnaliser ces modèles. Des outils de l'état de la pratique (*p. ex.* XText, MPS, ANTLR) sont utilisés en support à un travail de projet qui permet la création d'un langage spécifique. Ce cours repose sur un zoo d'implémentation de langage montrant 15 implémentations alternatives du même langage pour aider les étudiants dans leurs développements.
- Depuis 2012 **Génie Logiciel & Qualité**, Licence.
◦ Site : http://www.i3s.unice.fr/~mosser/teaching/17_18/se/start
Ce cours met l'accent sur des techniques avancées de génie logiciel, et complète le tronc commun. En lien avec les approches de tests, on y décrit des méthodes comme les tests par mutation, le test de propriétés ou encore les tests d'acceptation, en utilisant des outils de l'état de la pratique en support aux travaux de développement demandés en projet. Les exigences sont modélisées avec des *réécits utilisateurs*, en utilisant la pile logicielle Atlassian en support (Jira, Bitbucket). Ce cours repose sur un jeu de programmation, où les étudiants développent des robots qui explorent des terrains inconnus. Une arène de jeu commune est utilisée pour permettre aux étudiants de mesurer leur avancée par rapport aux autres.

Cours invités

- 2017 – 2019 **Software Engineering & Compilation**, ENS Lyon, Maîtrise.
◦ Site : <https://github.com/mosser/sec-labs>
L'École Normale Supérieure de Lyon est une université intensive en recherche, avec un fort ancrage dans la recherche fondamentale en informatique. Je coordonne pour deux ans avec Laure Gonnord ce cours qui met l'accent sur le génie logiciel appliqué aux langages spécifiques, du point de vue de la compilation. On y utilise l'exemple de la modélisation de systèmes réactifs embarqués (sur microcontrôleurs Arduino) pour discuter des différents niveaux d'abstractions existant et des moyens de mise en œuvre associés.

2015 – 2017 **Ingénierie dirigée par les Modèles**, *École d'été EJCP*, Doctorat.

J'ai été en charge pour trois ans du module de génie logiciel de l'*École des Jeunes Chercheurs en Programmation (EJCP)*. Cette école d'été annuelle regroupe environ 45 étudiants, avec un support financier du CNRS. Durant mon mandat, j'ai proposé un cours de génie logiciel orienté sur les langages spécifiques, discutant de modélisation et de séparation des préoccupations.

Service professionnel

- Comités de pilotage
- Co-Président, *Int. Workshop on Modularity in Modeling*, 2016-...
 - *Social Media Chair* pour la conférence Modularity (2015)
 - *Career development co-chair* pour la conférence SERVICES (2012)
 - *Demo chair* de l'atelier Benevol (*Benelux Software Evolution*, 2010)
- Comités de programme
- *Int. Workshop on Modeling for Micro-services*, depuis 2018
 - *Int. Conference on Big Data (BigData)*, depuis 2015
 - *Int. Workshop on Scalable Data Management (SCDM)*, depuis 2014
 - *Int. Workshop on Model-driven Cloud engineering*, 2014
 - *Int. Conference on Web Services (ICWS)*, depuis 2013
 - *Nordic Workshop on Cloud computing*, 2013 & 2014
- Comités de thèse
- Externes :
 - 2017-... : Thibault Béziers la Fosse (Mines-Télécom Bretagne, comité de suivi)
 - “*A Model-driven Framework for Dynamic Program Analysis in CPS*”
 - 2016: Mai Anh Bui (Université Paris 6, rapporteur & examinateur)
 - “Séparation des préoccupations en épidémiologie”
 - Internes :
 - 2017: Cyril Cecchinel (Université Côte d'Azur, co-encadrant)
 - 2017: Ivan Logre (Université Côte d'Azur, encadrant)
 - 2014: Antoine Feugas (Université de Lille, co-encadrant)
- Relecteur
- *Journal of Software and Systems (JSS)*
 - *Software & System Modelling (SoSym)*
 - *Software Quality Journal (SQJ)*
 - *Empirical Software Engineering (ESE)*
- Expertise
- Conseil de recherches en sciences naturelles et en génie du Canada (Découverte, 2017)
 - Agence Nationale de la Recherche (Émergence, 2011)
- Coordination de groupes de travail
- GL/\CE (2015-2020) : Ce groupe de travail financé par le CNRS (GdR GPL) s'intéresse à l'application du génie logiciel aux systèmes cyber-physiques. Nous y organisons des journées de travail, et participons à l'organisation de conférences thématique sur le sujet. Le groupe implique 17 équipes de recherche.
 - PING (2013) : Ce groupe de travail a réfléchi à la mise en place d'un “bagage” commun nécessaire à la réalisation d'une thèse de doctorat dans une équipe de génie logiciel du CNRS. Il incluait 13 équipes de recherche, et a amené à la création d'un site *wiki* collaboratif partagé.
- Organisation de conférences
- École des Jeunes Chercheurs en Programmation (EJCP 2019)
 - Journées nationales du GdR GPL (2011)
- École d'ingénieur
- Membre élu du conseil de département depuis 2014
 - Coordination des enseignements de génie logiciel depuis 2014
 - Coordination de la majeure “Architecture Logicielle” (~35 étudiants/an) depuis 2013
 - Coordination des projets depuis 2012

- Représentation de l'école
 - Ateliers et formations à l'agilité pour les ingénieurs du CNRS (DevLog, 2015 & 2017)
 - Sommet sur l'Innovation Pédagogique, Université de Rennes, 2017
 - Sommet sur les jeux sérieux dans l'enseignement supérieur, Université du Maine, 2017
 - Conférence technique Scala.io, 2016
 - Agile Tour Sophia Antipolis, 2015
 - Co-fondateur de la *Nuit de l'Informatique* (depuis 2007)

- Communauté
 - Coadministrateur du groupe *Software Craftsmanship meetup* (depuis 2017)
 - Organisateur local d'un *Google Hashcode Hub* (depuis 2017)
 - *Startup Week-end* Nice Sophia Antipolis (organisateur : 2016, pilotage : 2016-...)
 - Organisation de la *Global Day of Code Retreat* (depuis 2016)
 - Co-fondateur de l'*Agile Playground* Sophia Antipolis (2013-2017)

Développement logiciel

Projets en cours

- ACE (depuis 2017)
 - Abstract Composition Engine**, (*Scala, Neo4J*).
 - Rôle : Architecte & Développeur principal
 - <https://github.com/ace-design/ace> (*travail en cours*)

Ce canevas logiciel vise à devenir un moteur de composition abstrait, réutilisable entre plusieurs domaines. Il exploite les capacités fonctionnelles du langage Scala et la base de données Neo4J pour stocker les modèles à composer sous forme de graphes.

- Zoo ArduinoML (depuis 2014)
 - Implémentation alternatives d'ArduinoML**, (*langages spécifiques*).
 - Rôle : Architecte, Mainteneur & Contributeur
 - <https://github.com/mosser/ArduinoML-kernel>

ArduinoML est un langage de modélisation simple pour les applications à base de machines à états exécutables sur microcontrôleurs Arduino. Ce dépôt concentre 15 implémentations alternatives du langage, réalisés par 9 contributeurs différents. Il est utilisé en exemple support pour des enseignements dédiés aux langages spécifiques.

- Island (Since 2012)
 - Jeu Sérieux pour l'Enseignement du Génie Logiciel**, (*Scala, Java*).
 - Rôle: Architecte & Développeur principal
 - <http://ace-design.github.io/island> (*site public*)
 - <https://github.com/ace-design/island> (*code source*)
 - <http://ace-design.github.io/IslandExploration> (*contribution externe*)

Island repose sur des techniques de génération procédurale de terrain pour générer des cartes d'îles réaliste (grâce à une collaboration informelle avec le laboratoire de géographie). Un jeu sérieux d'exploration est conçu au-dessus des cartes, et les étudiants sont amenés à développer des robots capables d'explorer ces îles en vue d'y réaliser des contrats. Un moteur de championnat permet de traiter facilement des cohortes d'une centaine d'étudiants avec une livraison par semaine.

- Implémentation support (depuis 2012)
 - Codes de référence en support à des cours**, (*Java*).
 - Rôle : Développeur principal
 - https://github.com/polytechnice-si/4A_ISA_TheCookieFactory (*Architecture Log.*)
 - <https://github.com/polytechnice-si/5A-Microservices-Integration> (*AOS*)
 - <https://github.com/polytechnice-si/5A-BPM-Demo> (*Processus d'Affaires*)
 - *Katas* de code (en support à des démonstrations pendant les cours):
 - https://github.com/polytechnice-si/3A_GL_KataPotter (*POO*)
 - https://github.com/polytechnice-si/3A_GL_FlagCapture (*Exigences*)
 - <https://github.com/polytechnice-si/3A-GL-DiceGame> (*Bouchons & Espions*)
 - https://github.com/polytechnice-si/3A-0GL-TDD_Kata (*Tests*)
 - <http://bit.ly/acceptance-shakespeare> (*Tests d'Acceptation*)

J'ai développé un ensemble d'implémentation de référence donné aux étudiants en support à mes modules d'enseignements. Le plus abouti est *The Cookie Factory*, une architecture J2E de référence utilisant des technologies libres. J'utilise aussi souvent des démonstrations pendant mes cours, et les codes sont disponibles pour les étudiants sous la forme d'un *kata* disponible sur GitHub.

Projets archivés

- SmartCampus (2012–2016) **Réseau de Capteurs Expérimental & Composition**, (*Java, Python, C*).
○ Rôle : Coordination du projet, Architecte & Supervision d'étudiant (Doctorat et Maîtrise)
- <http://smartcampus.github.io/> (*Site public*)
- <https://github.com/SmartCampus/middleware> (*Intergiciel*)
- <https://github.com/SmartCampus/ArduinoSensorServer> (*Déploiement Arduino*)
- <https://github.com/SmartCampus/SimulationFramework> (*Simulation de capteurs*)
- <https://github.com/ace-design/cosmic> (*Opérateurs de composition*)
- <https://github.com/ace-design/DEPOSIT> (*Politiques de collectes de données*)
Développé au dessus de microcontrôleurs Arduino, le canevas logiciel *SmartCampus* supporte la définition rapide d'un réseau de capteurs expérimental. Il définit un ensemble de codes de références pour déployer des politiques de collectes de données sur des cartes Arduino, ainsi que les outils de supervision du réseau associés. La thèse de Cyril Cecchinel a portée sur la définition d'un langage dédié et d'opérateurs de composition dans ce domaine. Plusieurs projets étudiants (maîtrise) ont travaillé sur l'infrastructure matérielle et sa simulation. Mon rôle était un rôle de coordination de projet et d'architecture globale des différentes contributions.
- CloudML (2010–2011) **Déploiement d'application infonuagique**, (*Scala, Java*).
○ Rôle : Architecte, Développeur & Supervision d'étudiant (Maîtrise)
- <https://github.com/SINTEF-9012/cloudscript> (Langage de déploiement)
- <https://github.com/SINTEF-9012/sensapp> (Plateforme d'exemple)
CloudML (initialement connu sous le nom de CloudScript) est un langage de définition d'application infonuagique, en vue d'aider leur déploiement. Initialement développé comme une preuve de concept de la thèse de maîtrise d'Eirik Brandtzæg, le projet a évolué après mon départ de SINTEF grâce aux efforts de Nicolas Ferryet *al.* Nous avons appliqué CloudML à une plateforme de collecte de donnée appelée SensAPP.
- ADORE (2007–2011) **Opérateurs de composition dans les Architectures de Services**, (*Prolog, Java, Lisp*).
○ Rôle : Architecte & Développeur principal
- <https://github.com/ace-design/adore> (*Doctorat*)
- <https://github.com/ace-design/gcoke> (*Post-doctorat*)
Cet outil à initialement supporté la validation empirique de mes travaux de thèse. Il repose sur une implémentation en logique du premier ordre (Prolog) d'opérateurs de compositions pour processus d'affaires. Durant mon post-doctorat, le moteur de composition à évolué pour prendre en charge d'autres langages, en adoptant une approche basée sur des graphes.
- jSeduite (2007–2010) **Système de Diffusion d'Information**, (*Java, BPEL, BPMN*).
○ Rôle : Architecte, Développeur Principal & Supervision d'étudiants (License, Maîtrise)
- <https://github.com/ace-design/jseduite>
La plateforme jSeduite définit un système de diffusion d'information sur-mesure, dont les différents éléments de personnalisation reposent sur la composition de sous-processus d'affaire écrit dans le langage BPEL. Cette plate-forme à été développée comme preuve de concept démontrant les capacités de composition d'ADORE, avant d'être industrialisé par la jeune pousse *The 6th screen*.

Publications (2007 – ...)

- Archives ouvertes nationales (HAL):
- <https://cv.archives-ouvertes.fr/sebastien-mosser>
- Les articles sont aussi disponibles sur mon site professionnel:
- <http://www.i3s.unice.fr/~mosser/research/publications>
- Des versions en prépublication des articles en cours de soumission sont disponible ici :
- <http://www.i3s.unice.fr/~mosser/preprints>

Dans la liste de publications ci-après, les étudiants soulignés étaient sous ma supervision (en Maîtrise ou Doctorat) lors de la soumission de l'article associé.

Articles en cours de soumission (pré-publications)

- [1] C. Cecchinel, F. Fouquet, **S. Mosser**, and P. Collet. Leveraging Live Machine Learning and Deep Sleep to Support a Self-Adaptive Efficient Configuration of Battery Powered Sensors. Submitted to Future Generation Computer Systems (journal, Elsevier) , Mar. 2018. status: *minor revision*.
- [2] L. Gonnord and **S. Mosser**. Du code aux modèles, des modèles au code: enseigner les langages dédiés. Submitted to 7^{ème} Conférence en Ingénierie du Logiciel (national conference), May 2018. status: *under review*.
- [3] B. Combemale, J. Kienzle, G. Mussbacher, O. Barais, W. Cazzola, P. Collet, T. Degueule, R. Heinrich, J.-M. Jézéquel, M. Leduc, T. Mayerhoer, **S. Mosser**, M. Schöttle, M. Strittmatter, and A. Wortmann. Concern-Oriented Language Development (COLD): Fostering Reuse in Language Engineering. Submitted to Computer Languages, Systems and Structures (journal, Elsevier), Apr. 2018. status: *minor revision*.

- [4] [I. Logre](#), **S. Mosser**, P. Collet, and M. Riveill. Leveraging the SOA Paradigm as a support for DSL Composition. Submitted to Software and Systems Modeling (journal, Springer), Aug. 2017. status: *under review*.

Conférences invitées

- [5] **S. Mosser**. Les aspects génie logiciel pour les Systèmes Cyber-Physique. In *Journées IIoT du GDR MACS, CNRS*, France, July 2018.
- [6] V. Aranega, A. Etien, and **S. Mosser**. Using Feature Model to build Model Transformation Chains. In *Journées 2013 du GDR GPL, CNRS*, France, Mar. 2013.
- [7] **S. Mosser**, G. Mussbacher, M. Blay-Fornarino, and D. Amyot. Une approche orientée aspect allant du modèle d'exigences au modèle de conception. In *Journées du GDR GPL*, pages 37–38, Lille, France, June 2011.

Articles de revues internationales

- [8] **S. Mosser** and M. Blay-Fornarino. ADORE, a Logical Meta-model Supporting Business Process Evolution. *Science of Computer Programming*, 78(8):1035 – 1054, 2013.
- [9] **S. Mosser**, M. Blay-Fornarino, and R. France. Workflow Design using Fragment Composition (Crisis Management System Design through ADORE). *Transactions on Aspect-Oriented Software Development (TAOSD)*, Special issue on Aspect Oriented Modeling:1–34, 2010.

Articles publiés en conférences internationales

- [10] [S. Lazreg](#), P. Collet, and **S. Mosser**. Assessing the Functional Feasibility of Variability-Intensive Data Flow-Oriented Systems. In *Symposium on Applied Computing (Best Paper Award)*, Pau, France, Apr. 2018.
- [11] [B. Benni](#), **S. Mosser**, N. Moha, and M. Riveill. A Delta-oriented Approach to Support the Safe Reuse of Black-box Code Rewriters. In *17th International Conference on Software Reuse (ICSR'18)*, Madrid, France, May 2018.
- [12] [B. Benni](#), **S. Mosser**, P. Collet, and M. Riveill. Supporting Micro-services Deployment in a Safer Way: a Static Analysis and Automated Rewriting Approach. In *Symposium on applied Computing*, Pau, France, Apr. 2018.
- [13] **S. Mosser** and J.-M. Bruel. Reconciling Requirements and Continuous Integration in an Agile Context (tutorial). In *26th International Requirements Engineering Conference*, Banff, Canada, Aug. 2018. IEEE.
- [14] F. Fouquet, T. Hartmann, **S. Mosser**, and M. Cordy. Enabling lock-free concurrent workers over temporal graphs composed of multiple time-series. In *Symposium on Applied Computing*, volume 8, Pau, France, Apr. 2018.
- [15] [C. Cecchinell](#), **S. Mosser**, and P. Collet. Towards a (de)composable workflow architecture to define data collection policies. In ACM, editor, *Symposium on Applied Computing (SAC 2016)*, Pisa, Italy, Apr. 2016.
- [16] [C. Cecchinell](#), **S. Mosser**, and P. Collet. Automated Deployment of Data Collection Policies over Heterogeneous Shared Sensing Infrastructures. In *23rd Asia-Pacific Software Engineering Conference*, Hamilton, New Zealand, Dec. 2016.
- [17] [C. Cecchinell](#), **S. Mosser**, and P. Collet. Software Development Support for Shared Sensing Infrastructures: A Generative and Dynamic Approach. In *International Conference on Software Reuse (ICSR'15)*, Miami, United States, Jan. 2015. Springer.
- [18] S. Urli, M. Blay-Fornarino, P. Collet, **S. Mosser**, and M. Riveill. Managing a Software Ecosystem Using a Multiple Software Product Line: a Case Study on Digital Signage Systems. In *Euromicro Conference series on Software Engineering and Advanced Applications(SEAA'14)*, Special issue: Software Product Lines and Software Ecosystems, pages 1–8, Verona, Italy, Aug. 2014. Elsevier.
- [19] [I. Logre](#), **S. Mosser**, P. Collet, and M. Riveill. Sensor Data Visualisation: A Composition-Based Approach to Support Domain Variability. In *European Conference on Modelling Foundations and Applications (ECMFA 2014)*, volume 8569, pages 101–116, York, United Kingdom, July 2014. Springer.
- [20] [A. Feugas](#), **S. Mosser**, and L. Duchien. A Causal Model to predict the Effect of Business Process Evolution on Quality of Service. In *Conference on the Quality of Software Architectures (QoSA)*, pages 143–152, Vancouver, Canada, June 2013. ACM.
- [21] [E. Brandtzæg](#), P. Mohagheghi, and **S. Mosser**. Towards a Domain-Specific Language to Deploy Applications in the Clouds. In *In 3rd International Conference on Cloud Computing, GRIDS, and Virtualization*, pages 213–218, 2012.
- [22] **S. Mosser**, M. Blay-Fornarino, and L. Duchien. A Commutative Model Composition Operator to Support Software Adaptation. In A. Vallecillo, J.-P. Tolvanen, E. Kindler, H. Störrle, and D. Kolovos, editors, *Modelling Foundations and Applications*, pages 4–19, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

- [23] C. A. Parra, D. Romero, **S. Mosser**, R. Rouvoy, L. Duchien, and L. Seinturier. Using Constraint-based Optimization and Variability to Support Continuous Self-Adaptation. In *27th ACM Symposium on Applied Computing (SAC'12), 7th Dependable and Adaptive Distributed Systems (DADS) Track*, pages 486–491, Trento, Italy, Mar. 2012.
- [24] V. Aranega, A. Etien, and **S. Mosser**. Using Feature Model to Build Model Transformation Chains. In R. B. France, J. Kazmeier, R. Breu, and C. Atkinson, editors, *Model Driven Engineering Languages and Systems*, pages 562–578, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.
- [25] F. D. G. Velásquez, M. Blay-Fornarino, and **S. Mosser**. Introducing Security Access Control Policies into Legacy Business Processes. In *Fifteenth International Enterprise Distributed Object Computing Conference (EDOC'11), short paper*, pages 42–49, Helsinki, Finland, Aug. 2011. IEEE.
- [26] **S. Mosser**, G. Mussbacher, M. Blay-Fornarino, and D. Amyot. From Aspect-oriented Requirements Models to Aspect-oriented Business Process Design Models. In *10th international conference on Aspect Oriented Software Development (AOSD'11)*, pages 1–12, Porto de Galinhas, Brazil, Mar. 2011. ACM.
- [27] **S. Mosser**, G. Hermosillo, A.-F. Le Meur, L. Seinturier, and L. Duchien. Undoing Event-Driven Adaptation of Business Processes. In *8th IEEE International Conference on Services Computing (SCC'11)*, pages 234–241, Washington DC, United States, July 2011. IEEE.
- [28] M. Clavreul, **S. Mosser**, M. Blay-Fornarino, and R. B. France. Service-Oriented Architecture Modeling: Bridging the Gap between Structure and Behavior. In J. Whittle, T. Clark, and T. Kühne, editors, *Model Driven Engineering Languages and Systems (MODELS'11)*, volume 6981 of *Lecture Notes in Computer Science*, pages 289–303, Wellington, New Zealand, Oct. 2011. Springer Berlin / Heidelberg.
- [29] M. Alférez, N. Amalio, S. Ciraci, F. Fleurey, J. Kienzle, J. Klein, M. Kramer, **S. Mosser**, G. Mussbacher, E. Roubstova, and G. Zhang. Aspect-Oriented Model Development at Different Levels of Abstraction. In *7th European Conference on Modelling Foundations and Applications (ECMFA'11)*, pages 1–16, Birmingham, United Kingdom, June 2011. Springer LNCS.
- [30] **S. Mosser**, A. Bergel, and M. Blay-Fornarino. Visualizing and Assessing a Compositional Approach of Business Process Design. In *Software Composition 2010*, page Springer's Lecture Notes in Computer Science, Malaga, Spain, June 2010. ACM SIGPLAN and SIGSOFT.
- [31] **S. Mosser**, M. Blay-Fornarino, and J. Montagnat. Orchestration Evolution Following Dataflow Concepts: Introducing Unanticipated Loops Inside a Legacy Workflow. In *International Conference on Internet and Web Applications and Services (ICIW)*, pages 1–6, Venice, Italy, May 2009. IEEE Computer Society.
- [32] **S. Mosser**, F. Chauvel, M. Blay-Fornarino, and M. Riveill. Web Service Composition: Mashups Driven Orchestration Definition. In *International Conference on Intelligent Agents, Web Technologies and Internet Commerce (IAWTIC'08)*, pages 1–6, Vienna, Austria, Dec. 2008. IEEE Computer Society.
- [33] **S. Mosser**, M. Blay-Fornarino, and M. Riveill. Web Services Orchestration Evolution : A Merge Process For Behavioral Evolution. In *2nd European Conference on Software Architecture (ECSA'08)*, pages 1–16, Paphos, Cyprus, Sept. 2008. Springer LNCS.

Articles publiés en ateliers internationaux

- [34] B. Benni, P. Collet, G. Molines, **S. Mosser**, and A.-M. Pinna-Dery. Teaching DevOps at the Graduate Level, a report from Polytech Nice Sophia (short paper). In *First International Workshop on Software Engineering Aspects of Continuous Development and New Paradigms of Software Production and Deployment*, Villebrumier, France, Mar. 2018. LASER Foundation, Springer.
- [35] M. Blay-Fornarino, G. Jungbluth, and **S. Mosser**. Applying DevOps to Machine Learning, ROCKFlows, a Story from the Trenches (short paper). In *First International Workshop on Software Engineering Aspects of Continuous Development and New Paradigms of Software Production and Deployment*, Villebrumier, France, Mar. 2018. LASER Foundation, Springer.
- [36] C. Cecchine, M. Jimenez, **S. Mosser**, and M. Riveill. An Architecture to Support the Collection of Big Data in the Internet of Things. In *International Workshop on Ubiquitous Mobile cloud (co-located with SERVICES)*, Anchorage, United States, June 2014.
- [37] **S. Mosser**, P. Collet, and M. Blay-Fornarino. Exploiting the internet of things to teach domain-specific languages and modeling: The arduinoml project. In *EduSymp@MODELS*, 2014.
- [38] P. Collet, **S. Mosser**, S. Urli, M. Blay-Fornarino, and P. Lahire. Experiences in Teaching Variability Modeling and Model-driven Generative Techniques. In *Proceedings of the 18th International Software Product Line Conference: Companion Volume for Workshops, Demonstrations and Tools - Volume 2*, SPLC '14, pages 26–29, New York, NY, USA, 2014. ACM.
- [39] S. Urli, **S. Mosser**, M. Blay-Fornarino, and P. Collet. How to Exploit Domain Knowledge in Multiple Software Product Lines? In *Fourth International Workshop on Product Line Approaches*

in *Software Engineering at ICSE 2013 (PLEASE 2013)*, page 4 p., San Francisco, United States, May 2013. ACM.

- [40] **S. Mosser**, [I. Logre](#), N. Ferry, and P. Collet. From Sensors to Visualization Dashboards: Need for Language Composition. In *Globalization of Modeling Languages workshop (GeMOC'13)*, Miami, United States, Sept. 2013.
- [41] D. Romero, S. Urli, C. Quinton, M. Blay-Fornarino, P. Collet, L. Duchien, and **S. Mosser**. SPLEMMMA: A Generic Framework for Controlled-Evolution of Software Product Lines. In *MAPLE/SCALE 2013*, volume 2, pages 59–66, Tokyo, Japan, Aug. 2013.
- [42] B. Combemale, J. DeAntoni, R. B. France, F. Boulanger, **S. Mosser**, M. Pantel, B. Rumpe, R. Salay, and M. Schindler. Report on the First Workshop On the Globalization of Modeling Languages. *CoRR*, abs/1408.5703, 2013, 1408.5703.
- [43] S. Urli, M. Blay-Fornarino, P. Collet, and **S. Mosser**. Using Composite Feature Models to Support Agile Software Product Line Evolution. In *International Workshop on Models and Evolution in MODELS Conference*, pages 1–6, Innsbruck, Austria, Sept. 2012.
- [44] [E. Brandtzæg](#), **S. Mosser**, and P. Mohagheghi. Towards CloudML, a Model-based Approach to Provision Resources in the Clouds. In *International Workshop on Cloud and MDE (co-located with ECNFA Conference)*, pages 1–6, 2012.
- [45] **S. Mosser**, F. Fleurey, B. Morin, F. Chauvel, A. Solberg, and I. Goutier. SENSAPP As a Reference Platform to Support Cloud Experiments: From the Internet of Things to the Internet of Services. In *Proceedings of the 2012 14th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, SYNASC '12*, pages 400–406, Washington, DC, USA, 2012. IEEE Computer Society.
- [46] **S. Mosser**, L. Duchien, C. A. Parra, and M. Blay-Fornarino. Using Domain Features to Handle Feature Interactions. In A. C. P. Series, editor, *Variability Modelling Software-Intensive Systems (VAMOS)*, pages 101–110, Leipzig, Germany, Jan. 2012. Ulrich Eisenacker, University of Leipzig, DE.
- [47] D. Ardagna, E. Di Nitto, G. Casale, D. Petcu, P. Mohagheghi, **S. Mosser**, P. Matthews, A. Gericke, C. Ballagny, F. D'Andria, C.-S. Nechifor, and C. Sheridan. MODAClouds: A Model-driven Approach for the Design and Execution of Applications on Multiple Clouds. In *Proceedings of the 4th International Workshop on Modeling in Software Engineering, MiSE '12*, pages 50–56, Piscataway, NJ, USA, 2012. IEEE Press.
- [48] C. Quinton, **S. Mosser**, C. Parra, and L. Duchien. Using Multiple Feature Models to Design Applications for Mobile Phones. In *MAPLE / SCALE workshop, colocated with SPLC'11*, pages 1–8, Munich, Germany, Aug. 2011.
- [49] **S. Mosser**, M. Blay-Fornarino, and M. Riveill. Service Oriented Architecture Definition Using Composition of Business-Driven Fragments. In *Models and Evolution (MODSE'09), MODELS'09 workshop*, pages 1–10, Denver, Colorado, United States, Oct. 2009.
- [50] **S. Mosser**. Are Functional Languages a good way to represent productive meta-models ? In *4th European Lisp Workshop (ELW'07)*, pages 1–6, Berlin, Germany, France, July 2007.

Posters & Misc

- [51] [I. Logre](#), **S. Mosser**, and M. Riveill. Composition Challenges for Sensor Data Visualization (poster). In *International Conference on Modularity (MODULARITY 2015)*, Fort Collins, United States, Mar. 2015.
- [52] **S. Mosser**. La Thèse ... (Seminar to new PhD Students). Feb. 2012.

Article de revues nationales

- [53] M. Blay-Fornarino, V. Hourdin, C. Joffroy, S. Laviotte, **S. Mosser**, A.-M. Pinna Déry, P. Renevier, M. Riveill, and J.-Y. Tigli. Architecture pour l'adaptation de Systèmes d'Information Interactifs Orientés Services. *Revue des Sciences et Technologies de l'Information - Série L'Objet : logiciel, bases de données, réseaux*, pages 93–118, 2007.

Articles de conférences et d'ateliers nationaux

- [54] F. Chauvel, **S. Mosser**, and A. Solberg. Reconsidering QoS Analysis in Dynamic and Open Systems. In *1ère conférence en ingénierie du logiciel(CIEL'12), short paper*, , Rennes, June 2012.
- [55] [A. Feugas](#), **S. Mosser**, A.-F. Le Meur, and L. Duchien. Déterminer l'impact d'une évolution dans les processus métiers. In *Journées sur l'Ingénierie Dirigée par les Modèles (IDM'11)*, pages 71–76, Lille, France, June 2011.
- [56] C. Brel and **S. Mosser**. Vers une approche flot de données pour supporter la composition d'interfaces homme-machine. In *Journées sur l'Ingénierie Dirigée par les Modèles(IDM'11)*, pages 1–6, Lille, France, June 2011. CNRS.
- [57] **S. Mosser** and M. Blay-Fornarino. Taming Orchestration Design Complexity through the ADORE Framework. In *Journées 2010 du GDR GPL, CNRS, Pau, France, Mar. 2010*.

- [58] **S. Mosser** and M. Blay-Fornarino. Réflexions autour de la construction dirigée par les modèles d'un atelier de composition d'orchestrations. In *15ème conférence francophone sur les Langages et Modèles à Objets (LMO'09)*, pages 1–16, Nancy, France, Mar. 2009. Cépadués.
- [59] **S. Mosser**, M. Blay-Fornarino, and M. Riveill. Un modèle d'évolution multi-vues des Architectures Orientées Services. In *Actes de l'Atelier Doctorant LMO'08(DOC LMO'08), workshop*, , page 6, Montréal, Mar. 2008. Université de Montréal -.
- [60] **S. Mosser**, M. Blay-Fornarino, P. Collet, and P. Lahire. Vers l'intégration dynamique de contrats dans des architectures orientées services : une experience applicative du modèle au code. In *2ème Conférence sur les Architectures Logicielles (CAL'08)*, pages 1–15, Montréal, Canada, Mar. 2008.
- [61] **S. Mosser**, M. Blay-Fornarino, and M. Riveill. Orchestrations de Services Web : Vers une évolution par composition. In *Atelier RIMEL (Rétro-Ingénierie, Maintenance et Evolution des Logiciels)*, , page 6, Toulouse, France, Mar. 2007. Dalila Tamzalit, Salah Sadou.
- [62] C. Joffroy, **S. Mosser**, M. Blay-Fornarino, and C. Nemo. Des Orchestrations de Services Web aux Aspects. In U. d. T. EMN, INRIA, editor, *3ème Journée Francophone sur le Développement de Logiciels Par Aspects (JFLDPA'2007)*, pages 1–13, Toulouse, France, Mar. 2007.

Projet de Recherche

Sébastien Mosser
mosser@i3s.unice.fr

Mai 2018

Résumé

Dans le contexte des systèmes à très large échelle et de la modélisation prise au sens large, il est établi qu'il n'existe pas un opérateur de composition unique qui permette d'appliquer aisément le paradigme de séparation des préoccupations. L'idée de ce paradigme est de décomposer la complexité d'un système en éléments (*p. ex.*, des modèles) plus petits et plus maîtrisables, et de déléguer à un outillage automatique (*p. ex.*, un opérateur de composition) la complexité de reconstruire le système attendu. Décomposer sans savoir recomposer est une activité condamnée à l'avance, et cette capacité à recomposer est encore plus fondamentale dans les systèmes large échelle (*p. ex.*, infonuagique, systèmes cyber-physiques) : dans ce contexte, il est nécessaire de composer des artefacts provenant de différents domaines, où les choix faits dans un domaine influencent les autres artefacts. Jusqu'à maintenant, un effort non négligeable est nécessaire pour définir, analyser et valider des opérateurs de composition logicielle automatisés. Cet effort est à l'origine d'un problème de passage à l'échelle pour le paradigme de séparation des préoccupations en tant que tel. Sur la base de mes expériences précédentes en méta-modélisation et composition logicielle (appliquée entre autres aux processus d'affaires, aux systèmes cyber-physiques et aux architectures de services), ce projet de recherche propose d'étudier la définition d'un modèle formel dédié à la composition logicielle, de ses racines algébriques à la mise en place d'un canevas logiciel dédié à sa réalisation. L'objectif est de livrer un *Moteur de Composition Abstrait* (MCA), qui permette un passage à l'échelle horizontal et vertical du paradigme de séparation des préoccupations. Au niveau horizontal, définir un moteur de composition pour un nouveau domaine sera ainsi facilité par une réutilisation intensive de résultats factorisés au niveau du MCA. Travailler au niveau "méta" permettra aussi un passage à l'échelle vertical, en rendant mesurables les capacités de passage à l'échelle des opérateurs définis. Cette proposition s'inscrit dans les thématiques de recherche du département d'informatique et permet d'envisager de nouvelles perspectives en termes de cas d'étude (*p. ex.*, internet des objets, services) et de collaborations (*p. ex.*, intelligence artificielle).

Table des matières

1	Contexte	2
1.1	Séparation des préoccupations	2
1.2	Exemple de composition logicielle & Passage à l'échelle	2
2	Travaux précédents	3
2.1	Opérateurs de composition logicielle définis	3
2.2	Collaborations industrielles	4
2.3	Collaborations académiques	5
3	Intégration au sein du Dépt. d'Informatique de l'UQAM	5
4	Plan de recherche détaillé à moyen terme : MAScot	6
4.1	Objectif 1 : Modélisation & Passage à l'échelle	6
4.2	Objectif 2 : Évaluation quantitative	7
4.3	Objectif 3 : Évaluation qualitative	8
4.4	Dissémination académique & Industrielle	9

1 Contexte

1.1 Séparation des préoccupations

Dijkstra a identifié très tôt le paradigme de séparation des préoccupations [14]. Ce paradigme propose de supporter la construction de systèmes complexes par la composition de sous-systèmes de complexité maîtrisée (“diviser pour mieux régner”). La séparation des préoccupations est utilisée dans de nombreux domaines variés [20], et, intuitivement, définir des préoccupations indépendantes nécessite à un moment ou un autre de les recomposer pour intégrer tous ces sous-systèmes en un seul [7]. Garder des préoccupations indépendantes dans savoir les recomposer n’est pas compatible avec le cycle de vie d’un logiciel, qui doit finalement être livrable et exécutable en tant que tel. Il est maintenant établi qu’il n’existe pas d’algorithme de composition idéal qui permette de s’adresser toutes les problématiques inhérentes au domaine de la composition logicielle. Ces opérateurs sont donc définis de manière disjointe, domaine par domaine, *p. ex.*, composition de modèles à classes, tissage d’aspects, combinaison d’automate, développement pas-à-pas [4]. L’apparition des systèmes à très large échelle [17] comme les systèmes distribués infonuagique ou les systèmes cyber-physiques mettent en exergue ce problème de composition : dans ce contexte, il est nécessaire de composer entre eux des modèles interdépendants, où les choix faits dans les uns influencent les autres. De plus, même à l’intérieur d’un seul domaine, plusieurs opérateurs (*p. ex.*, fusion, tissage, décomposition) peuvent être nécessaires [3], amenant à la création d’algèbres mêlant plusieurs lois de compositions sur les artefacts à composer, créant un problème de “composition des compositions”.

Les propriétés intrinsèques des opérateurs de composition (*p. ex.*, idempotence, régularité, associativité, commutativité) ont un impact fort sur la manière dont les artefacts peuvent être composés, et ne pas les prendre en compte amène à la synthèse de systèmes incohérents [35]. Par exemple, pour l’adaptation dynamique d’un logiciel à son contexte d’exécution, un moteur d’adaptation va activer ou désactiver certaines caractéristiques du logiciel en observant les modifications de ce contexte. Si les opérateurs de composition utilisés ne sont ni commutatifs ni idempotents, il est très difficile d’écrire les règles d’adaptation, qui doivent prendre en compte la redondance de l’observation du contexte et l’ordre dans lesquelles les adaptations sont décidées à la suite de ces observations. Un autre exemple est le système de conteneurs légers Docker (technologie de référence pour le déploiement d’architectures micro-services), où les développeurs créent leurs images de déploiement en composant leurs besoins au-dessus d’images existantes réutilisées en boîte noire. Cette activité est complexe, et la vérification du bien-fondé des images ainsi composées fait intrinsèquement parti du modèle d’affaire de la compagnie. Jusqu’à présent, les approches de compositions sont définies domaine par domaine, et l’effort de développement d’un nouvel opérateur est très important, tout comme la mise en œuvre du paradigme de séparation des préoccupations dans un nouveau domaine.

1.2 Exemple de composition logicielle & Passage à l’échelle

Dans cette section, nous considérons un exemple “jouet” simplifié dans le but d’illustrer les problèmes inhérents à la composition logicielle. On s’intéresse à un modèle d’expressions rationnelles reposant sur des automates, avec une Machine composée d’États reliés entre eux par des Transitions étiquetées par les symboles de l’alphabet à reconnaître. Selon le paradigme de séparation des préoccupations, on cherche à définir des expressions complexes par composition d’expressions plus simple. Ces compositions sont définies au travers d’opérateurs de sémantique connue, comme l’union des mots reconnus par chacune des opérands (\cup), l’intersection (\cap), ou encore la concaténation ($;$).

L’idée même de séparer des modèles pour les recomposer par la suite pose des problèmes de passage à l’échelle, à deux niveaux : *(i)* l’introduction d’opérateurs dans un nouveau domaine et *(ii)* la conception de ces opérateurs. Par exemple, en tant qu’utilisatrice de l’opérateur de composition \cup , Alice souhaite savoir si cet opérateur est compatible avec la taille des modèles qu’elle doit composer, du point de vue du temps d’exécution. La complexité temporelle de l’opérateur est-elle liée au nombre d’états, ou au nombre de transitions ? Quel est le temps d’exécution moyen de l’opérateur quand il est appliqué à des modèles similaires aux siens ? Quelle consommation mémoire ? Est-ce que l’ordre de passage des arguments change la complexité de l’union ? Est-il ainsi possible de l’utiliser dynamiquement dans un contexte d’adaptation par exemple, où doit-il être exécuté statiquement avant le démarrage de

l'application au vu de ses performances ? Du point de vue de la conception de ces trois opérateurs, l'ingénieur en charge de les concevoir (Bob) est confronté à des défis totalement différents. Par nature, l'opérateur d'union doit être commutatif, associatif, et distributif vis à vis de l'intersection. Comment s'assurer du respect de ses propriétés dans son implémentation ? Comment mettre en œuvre de la traçabilité dans la définition des opérateurs pour permettre d'identifier la source de problèmes si besoin ? Comment mesurer les points "chauds" de son algorithme et qui peuvent coûter cher au passage à l'échelle ? Son problème initial était de mettre en œuvre une implémentation d'un opérateur d'union d'expressions rationnelles, et est confronté à des problèmes complexes qui sortent de son champ de compétences.

Ce problème peut sembler simple quand il est restreint à cet exemple illustratif. La suite de ce document donne des exemples d'opérateurs de composition non triviaux définis lors de mes précédents travaux, ainsi qu'un plan de recherche à moyen terme pour étudier le problème sous-jacent.

2 Travaux précédents

Depuis mes travaux de thèse soutenus en 2010, j'ai été amené à appliquer des mécanismes de séparation des préoccupations et de composition logicielle dans différents domaines. Une liste complète de mes publications est disponible dans mon CV, je ne liste ici que les travaux les plus représentatifs de la proposition de recherche faite dans ce document. **Les publications [5, 6, 28, 32] sont jointes à cette proposition, représentant les travaux les plus récents ou ayant le plus d'impact par rapport à cette proposition de recherche.** Les autres publications sont disponibles au téléchargement via le système d'archives ouvertes HAL¹, ainsi que sur mon site professionnel².

2.1 Opérateurs de composition logicielle définis

- **Composition de descripteurs de déploiement pour conteneurs légers.** Docker est le standard de facto pour le déploiement d'architectures micro-services. Sur la base de descripteurs écrits dans un langage spécifique au domaine, les ingénieurs composent les directives de déploiement qui leur sont propres avec des images de déploiement existantes, pour créer leurs propres images [31]. L'opérateur de composition associé est primordial dans l'écosystème Docker, et une mauvaise utilisation de celui-ci génère des "bugs" de déploiement ou des incompatibilités difficiles à éradiquer. Dans ce contexte, un méta-modèle permettant d'exprimer les descripteurs de déploiement et leur composition a été défini, et un mécanisme de détection de conflit basé sur de la logique du premier ordre a été mis en place [5]. Ces travaux (doctorat de Benjamin Benni) ont été validés en exploitant l'infrastructure GitHub, par l'analyse automatisée de 20,000 descripteurs de déploiements et l'identification systématique de conflits parmi ceux-ci qui ne pouvaient être détectés sans prendre en compte la composition logicielle.
- **Composition de réécritures de code.** La réécriture automatique de code est un processus de génie logiciel largement utilisé en support à des tâches comme le refactoring, ou encore la réparation automatisée de code, par exemple dans le noyau Linux [27] ou dans les applications orientées objet [18]. En collaboration avec Naouel Moha du LATECE (UQAM), une approche basée sur un opérateur de composition a été définie pour permettre la réparation simultanée d'anti-patterns co-localisés dans des applications Android [6] à l'aide de processeurs de réécriture Spoon. Le bénéfice de travailler au niveau "méta" a été illustré par l'application de ces résultats au patches automatiques du noyau Linux effectués à l'aide du processeur de réécriture Coccinelle. Dans les deux cas, nous avons identifiés (doctorat de Benjamin Benni) des conflits qui n'étaient pas connus des développeurs des règles de ré-écriture.
- **Composition de modèles d'exigences variables et d'infrastructure matérielle** ("*best paper award*" [28]). Dans ces travaux (doctorat de Sami Lazreg), nous considérons des modèles d'exigences de l'industrie automobile mis en œuvre sous la forme de flots de données variables. Ces modèles doivent être composés avec des modèles d'infrastructure réels (*p. ex.*, pipeline de

1. <https://cv.archives-ouvertes.fr/sebastien-mosser>

2. <http://www.i3s.unice.fr/~mosser>

traitement vidéo sur un processeur GPU) pour permettre la mise en œuvre des applications finales, dans notre cas des systèmes de tableaux de bords matériels. Ces travaux reposent sur l'utilisation de mécanismes de variabilité logicielle et de méthodes formelles pour valider les compositions effectuées.

- **Composition d'applications dans les systèmes cyber-physiques.** Dans ces travaux (doctorat de Cyril Cecchinel), nous considérons les applications déployées sur des systèmes cyber-physiques (restreints aux réseaux de capteurs) du point de vue des ingénieurs concevant ces applications. Un méta-modèle a été défini pour exprimer les applications sous la forme de politique de collectes de données, et une algèbre de composition mise en place via plusieurs opérateurs permettant de (dé)composer ces politiques, ainsi que de gérer leur déploiement [9, 16]. L'objectif de ces travaux est de permettre la réutilisation d'une même infrastructure de capteurs par plusieurs applications hétérogènes. Nous avons validé ces travaux sur des infrastructures réelles à petite et moyenne échelle (grâce à l'infrastructure nationale EQUIPEX FIT Iot LAB), et sur des environnements simulés pour la validation à large échelle (70k places de parkings pour atteindre un modèle de l'ordre de celui de la ville de Moscou, plus grande Smart City répertoriée à date). Nous travaillons actuellement avec le laboratoire de géosciences sur l'extension de ces travaux aux flotteurs MERMAID (doctorat de Sébastien Bonnieux).
- **Composition de tableaux de bords multi-domaines.** Les tableaux de bords prennent une place prépondérante dans les interfaces hommes machines actuellement, typiquement pour le suivi de santé d'un sportif par exemple : une montre connectée collecte des applications sur le sportif, qui les analyse ensuite à travers un tableau de bord dédié. Dans ces travaux (doctorat d'Ivan Logre), nous nous sommes intéressés à la définition de ces applications, ainsi qu'à la composition des différents méta-modèles impliqués dans leur réalisation, *p. ex.*, méta-modèle de visualisation de données, de capteurs et d'exigences médicales [29].
- **Composition d'applications infonuagique orientée services.** Dans le cadre de mon premier poste académique permanent (SINTEF, Oslo), j'ai travaillé à la réalisation du premier prototype de CloudML, un langage spécifique au domaine permettant le déploiement d'application infonuagique [8]. L'originalité de l'approche était de supporter la composition de plusieurs applications sur des machines virtuelles existantes au niveau de la planification du déploiement, par analyse puis composition des déploiements au niveau des modèles CloudML.
- **(Dé-)composition de processus d'affaires.** Dans le cadre de ma thèse de doctorat, j'ai travaillé à la définition du moteur ADORE, un logiciel permettant la composition de processus d'affaires développés dans le langage industriel BPEL [22] au sein des architectures orientées services. Ces travaux sont ma première incursion dans le monde de la composition logicielle, et avait pour originalité la mise en œuvre d'un processus de composition qui, contrairement au tissage d'aspect classique, soit indépendant de l'ordre d'application des compositions [32]. Dans le cadre de mon post-doctorat effectué au sein du centre Inria Lille-Nord Europe, j'ai travaillé à étendre ce formalisme pour supporter des mécanismes de décomposition, et permettre une composition négative (là où mes travaux de thèse étaient uniquement additifs), appliquée à un autre langage de processus d'affaires (BPMN [34], plus orienté modélisation que BPEL).

2.2 Collaborations industrielles

Il est très important pour moi que les recherches menées en génie logiciel soient ancrées dans une réalité industrielle. C'est pourquoi les travaux menés autour de Docker résultent d'une collaboration avec IBM France Labs et la rencontre d'intérêt (*meetup*) *Docker User Group* présente à Sophia Antipolis. Ils seront aussi présentés au "*Global Tech Forum*" organisé par la société Amadeus en juin 2018 (conférence technique invitée, d'une audience d'environ 300 ingénieurs en développement logiciel). Les travaux autour des infrastructures matérielles sont réalisés dans le cadre d'une thèse CIFRE (avec contrat d'accompagnement industriel), en collaboration avec la société VISTEON pour permettre la validation métier et empirique du travail effectué. Les travaux de Cyril Cecchinel ont bénéficié d'un accompagnement du programme européen EIT Digital pour permettre leur transfert technologique au sein de la jeune pousse luxembourgeoise DataThings, et ceux de Sébastien Bonnieux dans la continuité sont menés en collaboration avec la société OSEAN (constructeur du flotteur MERMAID utilisé

comme cible de déploiement des travaux). Mes travaux sur les systèmes infonuagiques faisaient partie d'une collaboration Européenne FP7 (projets MODACloud & PaaSage), qui impliquait de nombreux partenaires industriels, et ont bénéficié pendant deux ans d'un partenariat avec Amazon Web Services pour utiliser leurs infrastructures. Mes travaux de post-doctorat ont été effectués dans le cadre d'un projet industriel régional (FUI), en collaboration avec un grand acteur de la distribution alimentaire en France (Auchan).

2.3 Collaborations académiques

Si la recherche inclut par nature une grande part de réflexion et de travail solitaire, il est pour moi impératif d'échanger avec des collègues et des étudiants pour faire naître et mûrir des idées. Mon approche de la recherche repose ainsi sur de nombreuses collaborations externes à mon laboratoire, permettant d'identifier de nouvelles synergies à l'international et d'enrichir les travaux de chacun. Il est vital que les idées s'échangent, se transmettent, et s'adaptent en fonction des contextes. Si les conférences sont le terrain idéal pour échanger autour de sujets matures et publiés, des ateliers ou des rencontres plus informelles, à échelle locale, nationale ou thématique sont vraiment importantes pour discuter de sujets naissants et initier de nouvelles collaborations. J'attache beaucoup d'intérêt aux collaborations pluridisciplinaires, comme par exemple notre récente collaboration (avec Mireille Blay-Fornarino) avec le laboratoire de géo-science de l'Université Côte d'Azur qui a donné une perspective nouvelle aux travaux de l'équipe sur les systèmes cyber-physiques. Dans la même lignée interdisciplinaire, Philippe Collet et moi avons récemment amorcé une collaboration avec le laboratoire d'Électronique et le département de Droit de l'Université pour travailler sur les capacités de composition et de contractualisation d'une chaîne de blocs ("*blockchain*") appliquée aux systèmes cyber-physiques.

3 Intégration au sein du Dépt. d'Informatique de l'UQAM

Mes travaux portent sur le génie logiciel, plus précisément la (méta-)modélisation et la séparation des préoccupations. D'un point de vue général, le projet de recherche proposé dans ce document est compatible avec les thématiques suivantes abordées dans le département d'informatique par le laboratoire LATECE : (i) Modélisation, validation, et mise en œuvre de processus d'affaires, (ii) Gestion de patrimoine et qualité logicielle et (iii) Infonuagique. D'un point de vue plus ciblé, je dispose déjà d'un début de collaboration avec Naouel Moha sur l'analyse de code qui a donné lieu à une première publication commune (ICSR'18). Les travaux menés par Jean Privat et Etienne Gagnon dans le domaine de la compilation pourraient compléter mes travaux sur les langages spécifiques aux domaines. Les travaux d'Hafedh Mili sur les aspects et les composants peuvent aussi être considérés comme vecteurs de collaborations, ainsi que tout ce qui touche aux architectures de services (*p. ex.*, Naouel Moha, Guy Tremblay). Du point de vue des processus de développement logiciels, les travaux menés sur l'agilité, l'ingénierie des exigences et les processus d'affaires (*p. ex.*, Sylvie Trudel, Hafedh Mili) sont très intéressants du point de vue de la composition logicielle. La défense d'une approche de publication de logiciels libres est alignée avec ma volonté de diffuser le résultat de mes recherches par du code source disponible librement. Pour finir, l'écosystème industriel Montréalais semble propice (en tout cas vu de France) à la mise en place de collaborations industrielles pour valider des résultats et amorcer des relations plus pérennes.

En fonction des opportunités, ma proposition de recherche à moyen terme pourrait aussi évoluer pour prendre en compte par exemple les travaux sur l'internet des objets fait dans le département au niveau réseau, et ainsi compléter mon expertise dans ce domaine qui est plus orientée sur du développement logiciel et de la modélisation architecturale. D'un point de vue formel, les compétences de Roger Villemaire sur la modélisation logique et la vérification de modèles seraient très complémentaires pour la mise en place de certains artefacts du projet de recherche proposé. Le département dispose de plus d'une forte expertise en intelligence artificielle (*p. ex.*, Marie-Jean Meurs, Mohamed Bougessa, Éric Beaudry), qui est un domaine où les mécanismes de composition logicielle sont balbutiants. Intégrer le département me permettrait ainsi de bénéficier de l'expertise de ses membres sur les différents sous-domaines identifiés, mais aussi de mettre mon expertise en composition logicielle à leur service.

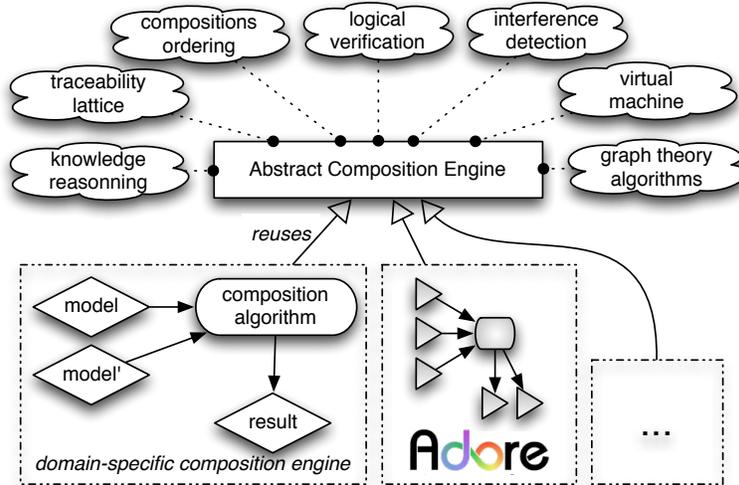


FIGURE 1 – Moteur Abstrait de Composition réutilisé dans plusieurs domaines (*p. ex.*, ADORE)

4 Plan de recherche détaillé à moyen terme : MAScot

Pour mettre en œuvre la vision décrite dans les sections précédentes, mon projet de recherche est ici détaillée à moyen terme (5 ans). L’objectif est la définition de MAScot, “*a Model-driven Approach to Support Scalable Software Composition*”. L’originalité de MAScot est de considérer la composition logicielle comme un citoyen de première classe et de réifier les abstractions communément utilisées dans un modèle formel dédié aux systèmes larges échelles tels que les systèmes infonuagiques ou l’internet des objets (Objectif 1). Ce modèle formel sera associé à une implémentation de référence d’un *Moteur de Composition Abstrait* (MCA) qui permettra d’évaluer les bénéfices et limitations de l’approche proposée de manière quantitative (Objectif 2). La validation qualitative en termes d’expressivité (Objectif 3) passera par l’application de celui-ci à des cas d’étude de taille industrielle, et l’interaction avec des partenaires industriels.

L’idée générale de l’approche est représentée en FIG.1. Le MCA (“Abstract Composition Engine”, ACE) définit les différents composants nécessaires à la conception d’un opérateur de composition, de manière indépendante du domaine d’application visé. Ces abstractions sont ensuite réutilisées par raffinement dans un domaine donné, permettant aux experts domaines de se concentrer sur leurs problématiques et de réutiliser les abstractions formalisées au niveau méta dans le MCA sans être confronté à leur complexité intrinsèque. Par exemple, en suivant cette approche, le moteur ADORE (composition de processus d’affaire décrits en BPEL) n’est plus défini à partir de rien, mais par la réutilisation puis spécialisation d’une machine virtuelle de composition, d’un ordonnanceur et d’un vérificateur de règles logique.

4.1 Objectif 1 : Modélisation & Passage à l’échelle

L’idée ici est de définir la version formelle du MCA, en mettant l’accent sur les propriétés associées aux lois de composition qui ont un impact direct sur le passage à l’échelle (*p. ex.*, commutativité pour réduire le nombre de configurations possible, régularité pour permettre la simplification des compositions, distributivité pour réécrire des compositions peu efficaces en l’état).

- **Propriétés des opérateurs de compositions.** Du point de vue algébrique, une structure équipée d’un opérateur (aussi appelée loi) de composition forme un magma. L’identification de propriétés supplémentaires sur ce magma permet de créer de nouvelles structures (*p. ex.*, anneaux, groupes, algèbre), et de bénéficier des résultats formels associés au niveau théorique. La difficulté est ici d’identifier les propriétés et structures algébriques pertinentes pour un développeur [25], en lui donnant accès uniquement aux éléments servant son métier : son point de vue est par exemple de chercher un opérateur pouvant s’appliquer sans connaître l’ordre des

modèles à composer à priori. Cette préoccupation métier est liée à la propriété de commutativité, qui peut se traduire au niveau du code de l’opérateur par la non utilisation de quantificateur universels dans celui-ci. En effet, un opérateur exprimant une règle de composition du type “*Pour toute classe C, faire ...*” est fragile sur son ordre d’application (il suffit qu’un des modèles à composer change la cardinalité de l’espace des classes pour obtenir un résultat différent). Cet exemple en particulier est identifié par la communauté travaillant sur les approches de composition par aspects sous le nom de point de coupe fragile (“*fragile pointcut*” [36]).

- **Ordonnement des compositions.** Les approches actuelles de composition reposent sur l’identification d’un ordre total sur l’ensemble des modèles à composer [21]. Quand l’ordre d’application des compositions n’est pas connu d’avance, ces approches sont problématiques, et peuvent amener à des situations où des systèmes incohérents sont synthétisés. En exploitant les propriétés identifiées précédemment, il est possible d’analyser les compositions demandées, pour potentiellement en proposer une réécriture plus efficace. Par exemple si un utilisateur demande la composition $(A \cup B) \cap C$, et qu’il est connu dans une base de faits que (i) l’union réagit mal aux modèles de grande taille et que (ii) l’intersection fait diminuer la taille des modèles, une réécriture possible serait de distribuer l’intersection sur l’union pour minimiser le temps de composition en exécutant $(A \cap C) \cup (B \cap C)$.
- **Détection d’interférences.** En gardant les préoccupations séparées, il est naturel de se retrouver face à des situations où des éléments disjoints interfèrent les uns avec les autres [23, 24]. Ces interférences ont besoin d’être identifiées lors de la composition pour ne pas générer de systèmes inconsistants [15]. Certaines interférences sont inhérentes au problème de composition, par exemple, quelle que soit les lois de composition considérées, si les deux tentent de modifier de manière concurrente le même élément, il existe un conflit entre les deux applications des lois, indépendamment du domaine d’application. Il peut aussi exister des conflits dépendant du domaine d’application, par exemple quand une loi crée des éléments qui ne sont pas compatibles avec une autre. Des méthodes formelles reposant sur la satisfaction de prédicats exprimés en logique du premier ordre ou encore l’analyse de paires critiques [19] permettent de traiter efficacement cette classe de problèmes.

4.2 Objectif 2 : Évaluation quantitative

L’idée clé de ce projet de recherche étant le passage à l’échelle des compositions, il est primordial de prendre en compte des moyens de mesure et d’évaluation empirique de celui-ci. On cherchera ainsi ici à produire une implémentation de référence du MCA, qui permette la mise en place d’expériences empiriques à large échelle.

- **Distribution des compositions.** Le modèle formel décrit dans la section précédente doit être compatible avec une distribution de l’exécution des compositions sur différentes machines pour s’exécuter de manière distribuée quand cela est possible, Cette problématique est liée au défi de la distribution des modèles dans les systèmes à ultra large échelle [12], ainsi qu’à la gestion de versions distribuée et à la théorie des patches [1]. Des opérateurs de composition exprimés sous la forme de règles de réécriture ou de règles logique [37] sont de bons candidats pour supporter une telle distribution.
- **Instrumentation des compositions.** L’évaluation quantitative d’un opérateur de composition passe par la collecte de métriques associées à son exécution. D’un point de vue indépendant du domaine d’application, des métriques comme le temps d’exécution, le nombre d’éléments de modèle lus ou modifiés, ou encore la consommation mémoire peuvent être collectées [26]. Des observateurs disponibles “sur étagères” peuvent alors être utilisés pour instrumenter les opérateurs définis. Et le canevas logiciel ayant permis leur écriture peut aussi être mis à disposition des développeurs pour permettre d’écrire facilement des instrumentations spécifiques à un domaine d’application particulier.
- **Banc de tests de référence.** La définition de bancs de tests est une tâche fastidieuse, et qui quand elle est mal maîtrisée peut amener à des conclusions faussées. D’un point de vue abstrait, des éléments comme la caractérisation du corpus de modèles à utiliser, l’automatisation des exécutions, ou leur traitement statistique et la visualisation des résultats associés

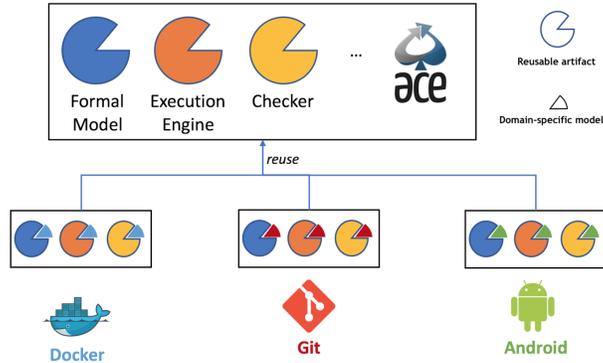


FIGURE 2 – Application du MCA à l'écosystème open-source

sont factorisables [13]. Dans certains domaines, il est possible de reposer sur des techniques de génération de modèles [33] pour automatiser la fabrication du corpus à utiliser dans le banc de tests. De tels bancs permettent de valider une complexité théorique (*p. ex.*, le temps d'exécution est quadratique en fonction du nombre d'éléments à composer) avec une réalité empirique. Par rapport aux propriétés des opérateurs de compositions identifiées précédemment, de tels bancs de tests peuvent aussi être utilisés pour les valider, en suivant des approches de tests basés sur les modèles [2], les tests métamorphiques [10] ou encore basés sur des propriétés [30].

4.3 Objectif 3 : Évaluation qualitative

En travaillant sur la composition logicielle au niveau abstrait, l'idée est de réduire le temps de mise sur le marché de nouveaux logiciels en diminuant l'effort nécessaire pour y introduire des mécanismes de composition. L'objectif est ici d'identifier et de caractériser des études de cas représentatives pour mesurer de manière qualitative les bénéfices de l'approche.

- **Validation Industrielle.** Cette validation est critique, et devra reposer sur un réseau de partenaires industriels rencontrant des problématiques de composition logicielle et souhaitant participer à la validation du MCA sur leurs cas. Par exemple, à travers la collaboration avec l'entreprise OSEAN (et le laboratoire de géo-sciences de l'Université), nous développons actuellement un moteur de composition dédié au flotteur MERMAID fabriqué par OSEAN. Ce flotteur permet de lancer des campagnes de terrains pour les géo-scientifiques (*p. ex.*, suivi de migration de baleines, détection de tremblements de terre), et le moteur de composition permet de ne pas se soucier des autres campagnes d'expérimentation actuellement déployées sur le flotte de MERMAIDS immergées en mer pour y déployer une nouvelle campagne. Nous travaillons aussi avec des entreprises de la technopole de Sophia-Antipolis sur un projet traitant de la contractualisation dans la blockchain [11] (en collaboration avec le laboratoire d'électronique et le département de Droit), du point de vue de la composition pour ma contribution au projet.
- **Logiciels Libres.** Une manière de pallier l'absence de partenaires industriels est de travailler sur les logiciels libres dont le code source est disponible à travers des plateformes comme GitHub ou Bitbucket par exemple. Ainsi, le MCA décrit dans ce document est applicable à des infrastructures logicielles comme Git, Android ou encore Docker, suite aux premières expérimentations faites à l'aide de ces cas d'études (FIG. 2). Travailler sur des logiciels libres permet une validation selon les deux dimensions suivantes : *(i)* un passage à l'échelle horizontale via l'ajout de nouveaux cas d'études, et *(ii)* un passage à l'échelle verticale permettant d'approfondir les résultats obtenus dans un domaine donné via la richesse de l'écosystème libre et la communauté d'utilisateur potentiellement plus large que dans des validations industrielles selon le type de cas d'étude.

4.4 Dissémination académique & Industrielle

Les résultats de ce travail de recherche ont vocation à être disséminés à travers des publications scientifiques, en conférences et en journaux. Il est très important pour moi de mettre à disposition dans un système d’archives ouvertes (*p. ex.*, HAL, Arxiv) les prépublications des articles acceptés, pour permettre la diffusion de ces résultats dans un esprit d’ouverture. En termes de conférences, ce travail de recherche vise particulièrement des forums dédiés à la modélisation et au génie logiciel, comme ASE, ECMFA, GPCE, ICSE, ICSR, MODELS, Programming<>, ou encore SAC. En termes de journaux, des publications comme SoSyM, FACS, JSS et TSE sont des cibles de premier ordre pour la valorisation de ces travaux. D’un point de vue applicatif, il est possible de disséminer les résultats obtenus du point de vue des cas d’étude, en visant des forums spécialisés dans ces domaines (*p. ex.*, les travaux sur Docker ont été publiés dans un forum dédié aux services, ceux sur les anti-patrons Android dans un forum dédié à la réutilisation de code).

Le MCA sera disponible sous la forme d’un logiciel libre mis à disposition de la communauté. Un site internet dédié au projet sera mis en ligne, avec une description des cas d’études utilisées en validation et la mise en ligne d’un ou plusieurs tutoriels montrant comment réutiliser le MCA dans un nouveau contexte. L’objectif de ce site web est la mise en place d’une vitrine démontrant les résultats obtenus en laboratoire, à destination de la communauté académique, mais aussi industrielle.

Du point de vue industriel, la participation de partenaires industriels est cruciale (voir Objectif 3) pour une validation empirique des résultats obtenus, et ainsi mesurer le gain apporté pour les développeurs de logiciel. La mise en place d’un réseau industriel via de la prospection et la participation à des rencontres de type “*meetups*” de développement représentera un effort non négligeable mais nécessaire pour faire connaître cette activité de recherche et identifier de nouvelles études de cas et potentiels sources de financements de ces travaux. L’écosystème montréalais dispose d’une très bonne implantation d’industriels au fait des problématiques de génie logiciel (*p. ex.*, Microsoft, Amazon, Google, IBM, CGI, Alten, Ubisoft, Autodesk).

Références

- [1] Carlo Angiuli, Edward Morehouse, Daniel R. Licata, and Robert Harper. Homotopical patch theory. In *Proceedings of the 19th ACM SIGPLAN International Conference on Functional Programming*, ICFP ’14, pages 243–256, New York, NY, USA, 2014. ACM.
- [2] Larry Apfelbaum and John Doyle. Model based testing. In *Software Quality Week Conference*, pages 296–300, 1997.
- [3] Don Batory. Using modern mathematics as an fosc modeling language. In *Proceedings of the 7th International Conference on Generative Programming and Component Engineering*, GPCE ’08, pages 35–44, New York, NY, USA, 2008. ACM.
- [4] Don Batory, Jacob Neal Sarvela, and Axel Rauschmayer. Scaling step-wise refinement. In *Proceedings of the 25th International Conference on Software Engineering*, ICSE ’03, pages 187–197, Washington, DC, USA, 2003. IEEE Computer Society.
- [5] Benjamin Benni, Sébastien Mosser, Philippe Collet, and Michel Riveill. Supporting Micro-services Deployment in a Safer Way : a Static Analysis and Automated Rewriting Approach. In *Symposium on applied Computing*, Pau, France, April 2018.
- [6] Benjamin Benni, Sébastien Mosser, Naouel Moha, and Michel Riveill. A Delta-oriented Approach to Support the Safe Reuse of Black-box Code Rewriters. In *17th International Conference on Software Reuse (ICSR’18)*, Madrid, France, May 2018.
- [7] Jean Bézivin, Salim Bouzitouna, Marcos Didonet Del Fabro, Marie-Pierre Gervais, Frédéric Jouault, Dimitrios Kolovos, Ivan Kurtev, and Richard F. Paige. A canonical scheme for model composition. In *Proceedings of the Second European Conference on Model Driven Architecture : Foundations and Applications*, ECMDA-FA’06, pages 346–360, Berlin, Heidelberg, 2006. Springer-Verlag.
- [8] Eirik Brandtzæg, Sébastien Mosser, and Parastoo Mohagheghi. Towards CloudML, a Model-based Approach to Provision Resources in the Clouds. In *International Workshop on Cloud and MDE (co-located with ECNFA Conference)*, pages 1–6, 2012.

- [9] Cyril Cecchinel, Sébastien Mosser, and Philippe Collet. Automated Deployment of Data Collection Policies over Heterogeneous Shared Sensing Infrastructures. In *23rd Asia-Pacific Software Engineering Conference*, Hamilton, New Zealand, December 2016.
- [10] Tsong Yueh Chen, Fei-Ching Kuo, Huai Liu, Pak-Lok Poon, Dave Towey, TH Tse, and Zhi Quan Zhou. Metamorphic testing : A review of challenges and opportunities. *ACM Computing Surveys (CSUR)*, 51(1) :4, 2018.
- [11] K. Christidis and M. Devetsikiotis. Blockchains and smart contracts for the internet of things. *IEEE Access*, 4 :2292–2303, 2016.
- [12] Cauê Clasen, Marcos Didonet Del Fabro, and Massimo Tisi. Transforming Very Large Models in the Cloud : a Research Roadmap. In *First International Workshop on Model-Driven Engineering on and for the Cloud*, Copenhagen, Denmark, July 2012. Springer.
- [13] Marco D’Ambros, Michele Lanza, and Romain Robbes. Evaluating defect prediction approaches : a benchmark and an extensive comparison. *Empirical Software Engineering*, 17(4) :531–577, Aug 2012.
- [14] Edsger W. Dijkstra. *On the Role of Scientific Thought*, pages 60–66. Springer New York, New York, NY, 1982.
- [15] Rémi Douence, Pascal Fradet, and Mario Südholt. A framework for the detection and resolution of aspect interactions. In *Conference on Generative Programming and Component Engineering*, Pittsburgh, PA, USA, October 2002.
- [16] François Fouquet, Thomas Hartmann, Sébastien Mosser, and Maxime Cordy. Enabling lock-free concurrent workers over temporal graphs composed of multiple time-series. In *Symposium on Applied Computing*, volume 8, Pau, France, April 2018.
- [17] Richard P. Gabriel, Linda Northrop, Douglas C. Schmidt, and Kevin Sullivan. Ultra-large-scale systems. In *Companion to the 21st ACM SIGPLAN Symposium on Object-oriented Programming Systems, Languages, and Applications*, OOPSLA ’06, pages 632–634, New York, NY, USA, 2006. ACM.
- [18] G. Hecht, R. Rouvoy, N. Moha, and L. Duchien. Detecting antipatterns in android apps. In *2015 2nd ACM International Conference on Mobile Software Engineering and Systems*, pages 148–149, May 2015.
- [19] Reiko Heckel, Jochen Malte Küster, and Gabriele Taentzer. Confluence of typed attributed graph transformation systems. In *Proceedings of the First International Conference on Graph Transformation*, ICGT ’02, pages 161–176, London, UK, UK, 2002. Springer-Verlag.
- [20] Walter L. Hürsch and Cristina Videira Lopes. Separation of concerns. Technical report, 1995.
- [21] Praveen Jayaraman, Jon Whittle, Ahmed M. Elkhodary, and Hassan Gomaa. Model composition in product lines and feature interaction detection using critical pair analysis. In Gregor Engels, Bill Opdyke, Douglas C. Schmidt, and Frank Weil, editors, *Model Driven Engineering Languages and Systems*, pages 151–165, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.
- [22] Matjaz B. Juric. *Business Process Execution Language for Web Services BPEL and BPEL4WS 2Nd Edition*. Packt Publishing, 2006.
- [23] Emilia Katz and Shmuel Katz. Incremental analysis of interference among aspects. In *Proceedings of the 7th Workshop on Foundations of Aspect-oriented Languages*, FOAL ’08, pages 29–38, New York, NY, USA, 2008. ACM.
- [24] Jörg Kienzle, Ekwa Duala-Ekoko, and Samuel Gélineau. *AspectOptima : A Case Study on Aspect Dependencies and Interactions*, pages 187–234. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.
- [25] Jacques Klein, Franck Fleurey, and Jean-Marc Jézéquel. *Weaving Multiple Aspects in Sequence Diagrams*, pages 167–199. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.
- [26] Michele Lanza and Radu Marinescu. *Object-Oriented Metrics in Practice : Using Software Metrics to Characterize, Evaluate, and Improve the Design of Object-Oriented Systems*. Springer Publishing Company, Incorporated, 1st edition, 2010.
- [27] Julia Lawall. Coccinelle : Reducing the barriers to modularization in a large c code base. In *Proceedings of the Companion Publication of the 13th International Conference on Modularity*, MODULARITY ’14, pages 5–6, New York, NY, USA, 2014. ACM.
- [28] Sami Lazreg, Philippe Collet, and Sébastien Mosser. Assessing the Functional Feasibility of Variability-Intensive Data Flow-Oriented Systems. In *Symposium on Applied Computing, Best Paper Award*, Pau, France, April 2018.

- [29] Ivan Logre, Sébastien Mosser, Philippe Collet, and Michel Riveill. Sensor Data Visualisation : A Composition-Based Approach to Support Domain Variability. In *European Conference on Modelling Foundations and Applications (ECMFA 2014)*, volume 8569, pages 101–116, York, United Kingdom, July 2014. Springer.
- [30] Andreas Löscher and Konstantinos Sagonas. Targeted property-based testing. In *Proceedings of the 26th ACM SIGSOFT International Symposium on Software Testing and Analysis*, pages 46–56. ACM, 2017.
- [31] Dirk Merkel. Docker : Lightweight linux containers for consistent development and deployment. *Linux J.*, 2014(239), March 2014.
- [32] Sébastien Mosser and Mireille Blay-Fornarino. ADORE, a Logical Meta-model Supporting Business Process Evolution. *Science of Computer Programming*, 78(8) :1035 – 1054, 2013.
- [33] Alix Mougnot, Alexis Darrasse, Xavier Blanc, and Michèle Soria. Uniform random generation of huge metamodel instances. In Richard F. Paige, Alan Hartman, and Arend Rensink, editors, *Model Driven Architecture - Foundations and Applications*, pages 130–145, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.
- [34] OMG. Business Process Model and Notation (BPMN), Version 2.0, January 2011.
- [35] A. Rauschmayer, A. Knapp, and M. Wirsing. Consistency checking in an infrastructure for large-scale generative programming. In *Proceedings. 19th International Conference on Automated Software Engineering, 2004.*, pages 238–247, Sept 2004.
- [36] M. Stoerzer and J. Graf. Using pointcut delta analysis to support evolution of aspect-oriented software. In *21st IEEE International Conference on Software Maintenance (ICSM'05)*, pages 653–656, Sept 2005.
- [37] Greg Straw, Geri Georg, Eunjee Song, Sudipto Ghosh, Robert France, and James M. Bieman. Model composition directives. In Thomas Baar, Alfred Strohmeier, Ana Moreira, and Stephen J. Mellor, editors, *“UML” 2004 — The Unified Modeling Language. Modeling Languages and Applications*, pages 84–97, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

Projet d'Enseignement

Sébastien Mosser
mosser@i3s.unice.fr

Mai 2018

J'occupe depuis 2012 un poste permanent de Maître de Conférences au sein de l'école d'ingénieur Polytech Nice – Sophia Antipolis, qui fait partie de l'Université Nice Sophia Antipolis (UNS) et de l'Université Côte d'Azur (UCA). J'y assure des enseignements de génie logiciel et d'architecture logicielle, au niveau licence et maîtrise, dans le département d'informatique de l'école. J'ai créé puis développé dans le département des enseignements originaux portant sur des aspects de génie en licence ou des concepts plus fondamentaux pour mes enseignements de maîtrise et de spécialité.

Dans tous mes enseignements, j'ai pour objectif de faire passer une base théorique solide, indépendante des outils utilisés en travaux dirigés ou travaux pratiques pour les mettre en œuvre. Cela passe par la mise en œuvre de cours indépendants des technologies, et la publication de code source de référence dans des technologies de l'état de la pratique (et si possible libre) à destination des étudiants pour leur permettre de faire le lien entre les concepts théoriques et une mise en œuvre pratique. La réussite des étudiants passe dans mes enseignements par la mise en pratique des concepts des cours pour ancrer ceux-ci dans une réalisation non triviale.

Responsabilité des projets. Je coordonne pour mon département les 5 modules d'enseignement (obligatoire) dédiés aux projets étudiants : deux en licence, et trois en maîtrise. J'ai beaucoup travaillé avec des experts en animation de groupes travaillant dans l'industrie pour permettre la mise en place d'un cadre pédagogique innovant autour de nos enseignements de projet et d'agilité. Ce cadre implique par exemple l'utilisation de jeux sérieux et de techniques d'accompagnements pour permettre aux étudiants d'expérimenter dans un "bac à sable" avant de mettre en pratique les concepts expérimentés (*p. ex.*, développement incrémental, rétrospectives, spécification, gestion de conflits, amélioration continue) sur leurs propres projets. Cette organisation est en cours de dissémination dans le réseau des écoles d'ingénieur Polytech et a fait l'objet de plusieurs présentations, industrielles (*p. ex.*, AgileTour) et académique (*p. ex.*, Journées Innovation Pédagogique, non nécessairement liées à l'informatique). En addition du travail de coordination et de promotion, j'accompagne des équipes d'étudiants lors des projets à temps partiel en licence, et des projets à temps plein (2 fois 4 semaines) en maîtrise, entre 12 et 25 équipes de 4 étudiants selon les promotions. Cet accompagnement passe par l'organisation de réunions quotidiennes d'avancement ("*standup meeting*"), l'organisation d'ateliers à la demande pour les équipes en ayant besoin (pédagogie différenciée), la coordination de l'équipe d'assistants d'enseignements et la logistique d'évaluation des réalisations techniques.

L'originalité de notre proposition d'organisation est que l'équipe d'accompagnement n'est pas impliquée dans l'évaluation des projets, ce qui permet aux étudiants de se confier plus facilement et de mieux mettre en avant leurs difficultés lors des séances. En maîtrise, les projets des étudiants donnent lieu à la réalisation de vidéos de démonstration qui sont mises en ligne sur la plateforme YouTube¹ afin de servir de carte de visite aux étudiants pour obtenir des stages en entreprises, et de vitrine de communication pour le département.

Enseignement au niveau licence. Mon enseignement principal est dédié au génie logiciel du point de vue des développeurs, en mettant en avant des principes fondateurs de l'agilité (livraison de valeur en continue, auto-organisation, spécification continue) et en les appliquant à des projets de taille

1. <https://www.youtube.com/user/pnsinnov>

conséquence et à un outillage industriel comme Jira pour la spécification ou Bitbucket pour la gestion de versions (je suis responsable de notre partenariat avec la société Atlassian pour les outils de suivi de projets depuis 2013, et m'occupe de l'administration du serveur hébergeant la solution pour notre université², avec 450 utilisateurs actifs), JUnit pour les tests et Sonarqube pour la mesure de qualité. Il est très important pour moi que les étudiants prennent du plaisir lors des enseignements, et à cette fin j'ai transformé en jeu le projet qu'ils doivent réaliser durant le semestre. Les étudiants se retrouvent face à un moteur de jeu (logiciel libre Island développé à cet effet, représentant 7k lignes de Scala et disponible sur GitHub³) dans lequel chaque semaine s'exécute la dernière version livrée de leur projet. Ils voient ainsi leurs progrès semaine après semaine, et peuvent positionner l'état d'avancement de leur travail par rapport au reste de la promotion, ainsi que par rapport au "champion" de l'année précédente. Cette compétition n'est pas prise en compte dans la notation finale pour éviter l'installation d'un état d'esprit de non-coopération, et les étudiants sont invités à s'entraider entre groupes pour ne laisser personne à la traîne. Ceci est rendu possible par l'évaluation de la manière dont ils ont construit leur logiciel et non par le résultat obtenu, ce qui limite les possibilités de plagiat. Bien que ce cours soit optionnel dans notre *curriculum*, il attire environ les deux-tiers de la promotion ingénieur annuellement. Ce travail d'enseignement autour des spécifications, de l'agilité et du génie logiciel a donné lieu à l'acceptation d'un tutoriel à la conférence internationale d'ingénierie des exigences ("*Reconciling Requirements and Continuous Integration in an Agile Context*"⁴, 26th *IEEE Int. Requirements Engineering Conf.*, 2018) en collaboration avec l'Université de Toulouse.

Enseignement au niveau maîtrise. Mes enseignements généralistes portent sur une introduction à l'architecture logicielle (classiquement des architectures n-tiers et un développement à base de composants sur la pile technologique J2E, en lien avec nos partenaires industriels locaux), en lien avec une approche "développement et exploitation" (*devops*) qui établit un continuum entre le développement d'une architecture logicielle et son exploitation en production, sous la forme d'un cours double (comptant 6 crédits contre 3 pour un cours habituel, et une implication étudiante de 8 heures hebdomadaires). Ce cours est monté en partenariat avec IBM France Labs dont un de leurs architectes assure 50% des enseignements, et a donné lieu à une publication le décrivant à l'atelier international "*Software Engineering Aspects of Continuous Development and New Paradigms of Software Production and Deployment*"⁵ organisé par la fondation LASER en mars 2018. L'originalité de ce module est de reposer sur un projet fil rouge commun exposé sous la forme d'un appel d'offre, et qui permet de travailler tant au niveau architectural qu'au niveau implémentation et exploitation. A la fin de ce cours, les étudiants livrent une réponse à l'appel d'offre sous la forme d'une architecture J2E, accompagnée de l'écosystème de déploiement continu (plans de compilation et de tests, conteneurs légers, mise en production automatique) associé à leur travail. En support à ce cours, une architecture de référence a été développée⁶, représentant 5k lignes de Java. Ce cours attire en moyenne 65% de la cohorte d'étudiants de maîtrise.

Enseignement de spécialité. En dernière année de maîtrise, mon département offre des cours de "spécialité", qui sont des demi-modules de cours optionnels, dédiés à des sujets précis. Dans ce contexte, j'assure deux cours dans le cadre des architectures orientées services : un premier cours sur les micro-services et leur déploiement dans des infrastructures à large échelle, et un second cours sur la modélisation de processus d'affaires. Dans le cadre du cours sur les micro-services, la partie théorique insiste fortement sur la qualité des interfaces de services, et la métrologie de supervision des services déployés, avec une approche très pratique sur un projet non trivial. Le cours de modélisation de processus d'affaires repose quant à lui sur une approche plus orientée recherche, avec une étude bibliographique de la part des étudiants sur des thèmes de recherche comme l'évaluation de processus d'affaires, les indicateurs de performance ou la fouille de processus dans des traces événementielles. Ce cours, à faible effectif (entre 15 et 20 étudiants), repose sur de nombreux échanges entre enseignant et étudiants, et un système de rendez-vous de suivi de projet.

2. <http://mjollnir.unice.fr>

3. <https://github.com/ace-design/island/>

4. <http://www.re18.org/tutorials.html#T02>

5. <https://www.laser-foundation.org/devops-18-program/>

6. https://github.com/polytechnice-si/4A_ISA_TheCookieFactory

Toujours dans le cadre des cours de spécialité, en lien avec mon activité de recherche sur la méta-modélisation et la composition logicielle, j’assure un cours sur les langages spécifiques aux domaines. Du point de vue technique, ce cours met en œuvre les principes de langages spécifiques sur des micro-contrôleurs Arduino, en permettant aux étudiants de trouver les bonnes abstractions à mettre en œuvre dans leurs langages pour programmer ces cartes électroniques. Il repose sur un zoo d’implémentations libres de ce langage (appelé ArduinoML⁷), qui donne un panorama d’implémentation très diverses du même problème, en utilisant un panel de technologie différentes. Ce cours est décliné de deux manière différentes, dans deux établissements : (i) au sein de Polytech Nice Sophia Antipolis avec une teinte génie et un projet conséquent de développement de langage, et (ii) au sein de l’*École Normale Supérieure de Lyon* (ENSL), avec une teinte recherche beaucoup plus forte, en collaboration avec Laure Gonnord. Préalablement au cours donné à l’ENSL, ce module a été décliné 3 ans dans l’école d’été francophone des jeunes chercheurs en programmation⁸. Un article décrivant ce module et son organisation est en cours d’écriture pour l’*Educator Symposium* de la conférence MODELS’18.

Encadrement doctoral. J’ai eu la possibilité de superviser six étudiants (dont trois thèses soutenues), qui ont pu publier leurs recherches dans des conférences de bon niveau. Alexandre Feugas travaille maintenant dans une entreprise de conseil en recherche et développement, Cyril Cecchinel est ingénieur de recherche dans l’entreprise où nous avons effectué un transfert technologique de ses résultats de thèse sur les réseaux de capteurs, et Ivan Logre est actuellement en post-doctorat à l’Université de Grenoble, sur un projet en lien avec la méta-modélisation d’Interfaces Homme Machines. L’accompagnement de jeunes chercheurs à travers un travail doctoral est pour moi une part considérable du métier de professeur, dans laquelle il faut savoir guider avec bienveillance tout en s’assurant que l’étudiant ne parte pas sur une voie de garage dans ses recherches. J’essaie de mettre en place une relation de confiance avec mes étudiants de maîtrise et de doctorat, qui repose fondamentalement sur la notion d’équipe, de cohésion sociale et de “vivre-ensemble”. Cette approche permet aux étudiants d’identifier d’eux-mêmes des liens entre leurs travaux, et de s’entre-aider dans leurs recherches, tant d’un point de vue scientifique que psychologique dans les périodes de doute inhérente au travail doctoral.

Philosophie d’enseignement. Plus globalement, j’essaie dans mes enseignements de mettre en œuvre une relation proche entre l’équipe enseignante et les étudiants. Il est critique que ceux-ci se sentent libre d’évoluer au sein d’un cours, et de le faire évoluer en tenant compte de leurs retours. Il est pour moi crucial que les étudiants s’impliquent dans le département par l’organisation d’évènements, et restent impliqués par la suite. C’est pourquoi je tiens particulièrement à faire intervenir d’anciens étudiants dans les modules de cours, en accompagnement des équipes d’étudiants actuels, ou pour donner des mini-conférences où ils peuvent partager leurs expériences de jeunes industriels. Je défends une communication la plus directe possible, ce qui est passé dans le département par l’abandon des courriels au profit d’un service de messagerie instantanée interne, commune aux enseignants et aux étudiants. Nous avons aussi mis en place un accès physique plus pratique au laboratoire pour permettre aux étudiants de venir discuter plus régulièrement avec leurs professeurs.

Je crois fortement en une approche de l’enseignement de l’informatique et du génie logiciel par la pratique. A cette fin, je participe à de nombreuses rencontres d’intérêt (“*meetups*”) pour tenir à jour les technologies de mise en pratique de nos enseignements, ainsi que nos pratiques d’accompagnement de projet. J’ai par exemple co-fondé l’*Agile Playground Sophia Antipolis*, un espace d’expérimentation bimensuel où des experts en facilitation de groupe peuvent s’entraîner à l’animation de jeux sérieux. Plus en lien avec le développement logiciel, je fais partie des membres fondateurs de la Nuit de l’Informatique (depuis 2007), une compétition nationale annuelle qui regroupe maintenant plus de 3,300 étudiants participants répartis dans 47 universités et une cinquantaine d’industriels partenaires offrant environ 20k€ de lots. Je participe aussi à l’organisation logistique de marathons de programmation (*p. ex.*, *Google Hashcode*, *BattleCode*, *DevBattle*) pour permettre à nos étudiants d’y participer dans des conditions optimales. Avec un collègue enseignant en algorithmique et informatique théorique, nous avons mis en place des séances d’accompagnement pour aider les étudiants le souhaitant à s’entraîner pour ce type de compétition. Enfin, afin de favoriser les initiatives d’entrepreneuriat étudiants, je suis membre de

7. <https://github.com/mosser/ArduinoML-kernel>

8. <http://gdr-gpl.cnrs.fr/Ecoles/EJCP> (éditions 2015, 2016 & 2017)

l'organisation du Startup Week-end Nice-Sophia Antipolis (comité d'organisation en 2016 puis pilotage) depuis que nous l'avons hébergé dans notre université.

J'ai aussi pris depuis 2013 au sein du département des responsabilités collectives (*c.-à-d.* direction de la spécialisation "Architecture Logicielle", coordination des enseignements de génie logiciel, codirection de la réforme des enseignements et membre élu du conseil de département) qui m'ont permis de comprendre la mécanique de développement d'une filière d'enseignement dans le contexte académique actuel, tant du point de vue pédagogique que financier et administratif. Par suite d'un problème structurel dans mon département (arrêt maladie de longue durée, départs en retraite, et mise en disponibilité de collègues auprès d'entreprises privées), j'ai été pour les années 2015-2016 et 2016-2017 le seul enseignant de génie logiciel de mon département. Cette situation a conduit à une très lourde charge d'enseignement (dans mon cas un double service) pour assurer la continuité de service et de formation aux étudiants, qui a conduit au recrutement (externe et par mutation) de 3 professeurs et 2 assistants doctorants dans la discipline afin de revenir à un rythme de croisière depuis la rentrée 2017.

Intégration au sein du Dépt. d'Informatique de l'UQAM

Cette section est basée sur les informations disponibles en ligne dans le *syllabus* de l'UQAM⁹. Les propositions faites dans cette section ne sont pas limitatives, je suis ouvert à toute proposition pour faire évoluer mes enseignements en fonction des besoins du département (particulièrement, la coloration des enseignements en génie logiciel ou en informatique est toujours adaptable). J'apprécie beaucoup les enseignements d'algorithmique et programmation pure (j'ai effectué la majeure partie de mon service d'assistant d'enseignement pendant ma thèse dans des matières de ce type), que je n'ai pas eu la possibilité de remettre en œuvre dans mon département vu le déficit d'enseignants en génie logiciel.

Au niveau baccalauréat (programme 7316, "Baccalauréat en informatique et génie logiciel"), mes compétences portent sur le génie logiciel, et peuvent compléter des cours comme "Développement de logiciel dans un environnement Agile" (INF2015), "Construction et maintenance de logiciels" (INF3135), et les cours de génie logiciel (INF5151 et INF5153). Je serais particulièrement intéressé de voir si le logiciel Island développé comme support au cours niçois de Génie Logiciel pourrait être utilisé dans le département en support à un enseignement. Mes compétences en facilitation de projet couplés à mon expérience de l'agilité dans le tissu industriel de la technopole de Sophia Antipolis pourraient aussi compléter l'offre de cours tournant autour des projets dans ce programme, *c.-à-d.* "Projet Génie Logiciel : conduite de projets informatiques" (INF6150), et "Projet d'analyse et de modélisation" (INF5151), par exemple par l'introduction de jeux sérieux et d'ateliers pour permettre l'expérimentation des concepts de génie logiciel.

Au niveau de la maîtrise en informatique (programme 3281) et en génie logiciel (programme 3822), mes compétences sont proches des cours de "Vérification et assurance qualité de logiciels" (MGL7560) pour la partie exigences et tests, ainsi que du cours "Qualité et productivité des outils logiciels" (MGL7760) qui mixe des concepts que j'enseigne ici en génie logiciel en licence et dans le cadre du cours de langage spécifique en spécialité de maîtrise. Le cours de "Réalisation et maintenance de logiciels" (MGL7460) comporte une grande partie dédiée aux tests, correspondant là aussi à mon domaine de compétence immédiat en enseignement. De manière plus lointaine, vis-à-vis de mes compétences en langage domaine, le cours de "Compilation" (INF7641) est aussi un point d'intérêt. Du point de vue de la programmation, mes compétences pourraient compléter les cours de baccalauréat dédié à cette activité (*p. ex.* INF1120, INF2120). Le cours de "paradigme de programmation" (INF2160) représente un fort point d'intérêt, ayant appris la programmation sur un langage fonctionnel et implémenté l'outil support à ma thèse de doctorat en Prolog.

Sauf erreur de ma part, le département d'informatique ne semble pas offrir de cours de maîtrise dédié (*i*) aux architectures de services, ou (*ii*) aux langages spécifiques aux domaines. S'il existe une volonté dans le département d'aller vers l'enseignement de ce genre de domaine (tant du point de vue génie que du point de vue informatique), je serais ravi de faire partager mon expérience sur ces sujets depuis 2012 et de faire une proposition de syllabus cohérente avec le reste de l'offre d'enseignement du département. Au niveau baccalauréat, je pourrais proposer un cours sur l'évaluation empirique

9. <http://syllabus.uqam.ca/>

d’algorithmes et structures de données en lien avec mon travail sur les bancs de tests et d’essais, en complément des cours de “programmation” I et II (INF 1120, INF2120) et de “structures de données et algorithmes” (INF3105).

Service d’enseignement détaillé assuré depuis 2012 à l’UNS

En tant que maître de conférences, mon service d’enseignement annuel théorique est de 192 heures de présentiel. A l’exception des deux années où j’étais en situation exceptionnelle de double service (2015-2016 et 2016-2017), ma charge d’enseignement tourne autour des 250 heures en comptant en plus des heures de cours les participations aux jurys d’évaluation, remplacement de collègues et autre charges collectives qui permettent à un département d’enseignement de fonctionner au quotidien en bonne intelligence. La table TAB. 1 synthétise mes enseignements depuis mon recrutement, avec la taille des cohortes étudiantes et les volumes associés. Un module de licence ou de maîtrise correspond à un volume d’un tiers de cours magistral et deux tiers de TD/TPs, sur 12 semaines. Un module de spécialité couvre 8 semaines, à raison d’une demi journée par semaine mélangeant cours magistraux et travaux pratiques. Mes cours sont disponibles en ligne : <http://www.i3s.unice.fr/~mosser/teaching/start>.

A l’exception de mes années d’assistant d’enseignement (“moniteur”) où j’ai majoritairement fait de l’encadrement de TP en programmation (et créé les TP de programmation web), j’assure les cours ainsi que les séances de travaux pratiques associés, en collaboration avec d’autres collègues ou assistants d’enseignements en fonction de la taille des cohortes.

Responsabilités Collectives à l’UNS

Depuis mon recrutement en 2012, je suis responsable des projets dans le département ¹⁰, ce qui implique la coordination des différentes équipes pédagogiques de chaque projet pour proposer une offre de formation cohérente sur les trois années du cycle ingénieur. J’ai ainsi participé à la transformation des modules de projets sur les trois années (5 modules de cours), en lien avec des industriels spécialistes de la facilitation et de l’accompagnement de projets. La participation à cette refonte m’a permis de comprendre comment articuler une offre de formation sur plusieurs années, ainsi qu’à coordonner des équipes d’enseignements large (par exemple au niveau licence, le projet de semestre implique une équipe de 8 enseignants).

En 2013, j’ai pris la responsabilité du parcours de spécialité “Architecture Logicielle” ¹¹, qui est depuis devenu le plus gros parcours du département : classiquement depuis 2015 environ 35 étudiants par année (sur 120, pour 7 parcours de spécialité). Cette responsabilité est avant tout pédagogique, et passe par la coordination des équipes enseignantes des différents cours de spécialités et la participation aux soutenances de stages. Une charge administrative est associée, qui passe par la validation des jurys de fin d’étude, l’organisation d’épreuves complémentaires au besoin et la validation des choix de cours des étudiants entrants. J’ai aussi eu la charge de gérer le recrutement d’un professeur associé (poste mi-industriel mi-académique, renouvelable par tranches de trois ans) en Architecture Logicielle pour compléter notre potentiel enseignant, ce qui a initié notre collaboration avec *IBM France Labs*.

Depuis 2014, j’assure la coordination des enseignements de Génie Logiciel du département. Cette responsabilité pédagogique consiste à s’assurer de l’articulation des cours de génie logiciel, et à garantir le potentiel enseignant associé. A cet effet, j’ai pu en 2016 défendre la mutation interne d’un collègue de la Faculté des Sciences pour renforcer nos cours de modélisation, et mis en avant la nécessité de recruter un Professeur permanent sur des thèmes de génie logiciel. Je suis membre de la “*Commission Permanente aux Ressources Humaines*” en Informatique, et participe ainsi au recrutement des assistants d’enseignement (doctorat et post-doctorat) lié au génie logiciel. Je suis aussi membre élu du conseil de département.

En 2016, mon université de tutelle a demandé une réduction drastique des coûts d’enseignement. A cet effet, j’ai codirigé avec trois collègues une réforme de fond de notre offre de formation. Nous avons mis en place un programme de cycle ingénieur sur trois ans (dernière année de licence et maîtrise) qui

10. <http://informatique.polytechnice.fr/SI/projets.html>

11. <http://informatique.polytechnice.fr/jahia/Jahia/informatique/parcoursSI5/AL>

TABLE 1 – Synthèse des enseignements depuis 2007

Année	Cours	Niveau	Cohorte	Vol. (h)
2017–2018	Qualité & Génie Logiciel	Licence	60	50
	Intro. à l’Architecture Logicielle & DevOps	Maitrise	50	40
	Projets Innovation	Maitrise	70	40
	Langages spécifiques aux domaines	Spécialité	30	32
	Architectures de Micro-services	Spécialité	40	32
	Modélisation de processus d’affaires	Spécialité	20	32
2016–2017*	Qualité & Génie Logiciel	Licence	80	70
	Projet de développement logiciel	Licence	80	50
	Intro. à l’Architecture Logicielle & DevOps	Maitrise	65	40
	Projet de développement agile	Maitrise	55	60
	Projets Innovation	Maitrise	60	40
	Langages spécifiques aux domaines	Spécialité	30	32
	Architectures SOA & Bus d’entreprise	Spécialité	50	32
	Modélisation de processus d’affaires	Spécialité	15	32
2015–2016*	Qualité & Génie Logiciel	Licence	90	70
	Projet de développement logiciel	Licence	90	50
	Intro. à l’Architecture Logicielle & DevOps	Maitrise	50	40
	Projet de développement agile	Maitrise	65	60
	Projets Innovation	Maitrise	65	40
	Langages spécifiques aux domaines	Spécialité	35	32
	Architectures SOA & Bus d’entreprise	Spécialité	50	32
	Modélisation de processus d’affaires	Spécialité	15	32
2014–2015	Outils pour la Qualité & Génie Logiciel	Licence	90	70
	Serveurs d’application	Maitrise	55	40
	Projets Innovation	Maitrise	70	40
	Ingénierie des Modèles	Spécialité	40	32
	Architectures de services	Spécialité	50	32
	Modélisation de processus d’affaires	Spécialité	20	32
2013–2014	Outils pour la Qualité & Génie Logiciel	Licence	70	70
	Modélisation d’architecture N-tiers	Maitrise	50	40
	Projets Innovation	Maitrise	65	40
	Ingénierie des Modèles	Spécialité	40	32
	Architectures de services	Spécialité	40	32
2012–2013	Outils pour la Qualité & Génie Logiciel	Licence	80	90
	Ingénierie des Modèles	Spécialité	30	32
	Architectures de services	Spécialité	45	32
Assistant d’enseignement 2007–2010	Programmation Web (création des TPs)	Licence	50	40/an
	Programmation Fonctionnelle (TPs)	Maîtrise	30	24/an
	Bases de données bioinfo (TPs)	Maîtrise	20	20
	Ingénierie dirigée par les modèles (TPs)	Spécialité	45	10

(*) Situation de double service en raison d’un sous-encadrement en génie logiciel dans le département.

met en avant une plus grande autonomie des étudiants, et un système de choix d’options qui nous a permis de dépasser les objectifs déjà hauts fixés par l’université sans avoir à trop réduire notre offre de cours. Nous manquons de recul sur la mise en place de la réforme (qui date de la rentrée 2017), mais les retours étudiants sont très positifs pour le moment, et nous avons prévu l’organisation d’ateliers impliquant professeurs et étudiants pour mettre en place un processus d’amélioration continue de cette nouvelle offre de formation dès le mois de Juin 2018.