



HAL
open science

Complexité du logiciel ambiant : de la composition dynamique à l'exécution distribuée, contextuelle, autonome et large-échelle

Frédéric Le Mouël

► To cite this version:

Frédéric Le Mouël. Complexité du logiciel ambiant : de la composition dynamique à l'exécution distribuée, contextuelle, autonome et large-échelle. Système d'exploitation [cs.OS]. Université de Lyon; Institut national des sciences appliquées de Lyon, 2016. tel-01712992

HAL Id: tel-01712992

<https://inria.hal.science/tel-01712992>

Submitted on 20 Feb 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



INSA

Inria

No d'ordre NNT : HDR 2016 011

Année 2016

**HABILITATION À DIRIGER DES RECHERCHES
DE L'UNIVERSITÉ DE LYON**

préparée au sein de
L'INSA DE LYON

ECOLE DOCTORALE NO 512
INFORMATIQUE ET MATHÉMATIQUES (INFOMATHS)
SPÉCIALITÉ INFORMATIQUE

Soutenue le 28 novembre 2016 par
FRÉDÉRIC LE MOUËL

**COMPLEXITÉ DU LOGICIEL AMBIANT :
DE LA COMPOSITION DYNAMIQUE À L'EXÉCUTION
DISTRIBUÉE, CONTEXTUELLE, AUTONOME ET
LARGE-ÉCHELLE**

devant le jury composé de

Président	PR Michel Riveill	Université Nice Sophia Antipolis
Rapporteurs	PR Thierry Delot	Université de Valenciennes
	PR Daniel Hagimont	ENSEEIH
	PR Michel Riveill	Université Nice Sophia Antipolis
Examineurs	PR Isabelle Guérin Lassous	Université Claude Bernard Lyon 1
	PR Fabrice Valois	INSA de Lyon
	MCF Philippe Roose	Université de Pau et des Pays de l'Adour

TRAVAUX EFFECTUÉS AU SEIN DU CENTRE D'INNOVATION EN TÉLÉCOMMUNICATIONS ET INTÉGRATION DE SERVICES (CITI), LABORATOIRE DE L'INSA DE LYON ET DE INRIA RHÔNE-ALPES, ÉQUIPE DYNAMIC SOFTWARE AND DISTRIBUTED SYSTEMS FOR THE INTERNET OF THINGS (DYNAMID)

Frédéric Le Mouél, *Complexité du logiciel ambiant :
de la composition dynamique à l'exécution distribuée, contextuelle, autonome
et large-échelle*, Habilitation à Diriger des Recherches, ©2016, Lyon

Donc le poète est vraiment voleur de feu.
Il est chargé de l'humanité, des animaux même ;
il devra faire sentir, palper, écouter ses inventions ;

— Arthur Rimbaud - Lettre du Voyant, à Paul Demeny, 15 mai 1871

Le monde est très grand et plein de contrées magnifiques
que l'existence de mille hommes ne suffirait pas à visiter.

— Arthur Rimbaud - Correspondance d'Aden, 15 janvier 1885

À ma constellation
Tess, Ellyn, Dubhe, Jacqueline, Gilbert, Renée, Joachim.

RÉSUMÉ

De pair avec le développement des intergiciels dans les années 1990-2000, l'intelligence ambiante va prendre un grand essor et dessiner les scénarios à horizon de 2020. Avec une croissance forte du nombre de périphériques - téléphones portables, capteurs, montres connectées, etc., avec des contraintes de taille et de batterie, avec des propriétés de mobilité, le déploiement, la gestion, la programmation de ces environnements se sont grandement complexifiés. Les intergiciels présentent de bonnes caractéristiques d'abstraction - permettant la modularité et la réutilisation, et de bonnes capacités d'interconnexions - permettant l'ouverture et la fiabilité des systèmes, et jouent donc un rôle prépondérant dans la mise en place actuelle des environnements ambiants. Ces dernières années, mes contributions ont porté sur l'amélioration de trois propriétés des intergiciels que sont la gestion du dynamisme, le passage à l'échelle et l'autonomie. Le développement de plusieurs intergiciels ont permis de valider les choix technologiques et scientifiques effectués. Jooflux et ConGolo sont des approches basées sur des machines virtuelles permettant d'intégrer le dynamisme au sein des applications, soit de manière explicite avec de la programmation contextuelle avec ConGolo, soit de manière transparente avec des mises à jour dynamiques et des aspects avec Jooflux. AxSeL, ACOMMA et MySIM sont des approches basées sur les architectures orientées-services capturant le dynamisme de l'environnement avec du chargement / déchargement contextuel de services, de l'exécution distribuée collaborative, et de la composition et du remplacement sémantique de services. CANDS permet de son côté de gérer de très grands flux d'information sur de très grands graphes de services tout en conservant de très bons temps de réponses. Pri-REIN y ajoute de la qualité de service. Un des résultats important est d'avoir montré que la propriété d'autonomie est fortement dépendante du domaine d'application des intergiciels et a notamment été testée sur la ville intelligente avec du guidage, de la gestion des parkings et de l'optimisation de trafic. Ces travaux ont été réalisés avec cinq étudiants en thèse ayant soutenus. Pour l'avenir, je propose de considérer ces propriétés des intergiciels pour l'Internet des Objets, avec les problèmes de provisionnement initial à large-échelle et de prise de décision distribuée autonome garantissant une sûreté de fonctionnement.

TABLE DES MATIÈRES

i	CONTEXTE	1
1	INTRODUCTION	3
1.1	Parcours	3
1.2	Contexte de recherche	4
1.3	Organisation du document	7
2	VERS DES INTERGICIELS AUTONOMES POUR ENVIRONNEMENTS AMBIANTS	9
2.1	Le retour du logiciel	10
2.2	Complexité des intergiciels	13
ii	CONTRIBUTIONS	17
3	JVM AS A SERVICE	19
3.1	JooFlux	19
3.2	ConGolo	24
4	CHARGEMENT ET COMPOSITION CONTEXTUELS DE SERVICES	29
4.1	AxSeL	29
4.2	ACOMMA	35
4.3	MySIM	39
5	VERS DES INTERGICIELS DISTRIBUÉS, PERFORMANTS ET LARGE-ÉCHELLE	45
5.1	Pri-REIN	45
5.2	CANDS	48
6	VERS DE L'ALGORITHMIQUE ET DES INTERGICIELS 'SOCIAUX'	55
6.1	Crowdsourcing véhiculaire	55
iii	EN RÉSUMÉ	65
7	CONCLUSIONS	67
8	PERSPECTIVES	69
8.1	Perspectives à court terme	69
8.2	Perspectives à long terme	70
iv	ANNEXES	75
A	CURRICULUM VITAE	77
B	PUBLICATIONS	89
B.1	Livres et chapitres de livre	89
B.2	Editions de livres ou actes	89
B.3	Brevets	89
B.4	Journaux internationaux	89
B.5	Journaux nationaux	90
B.6	Conférences internationales	91
B.7	Workshops internationaux	93
B.8	Conférences nationales	95
B.9	Ateliers nationaux	96
B.10	Rapports techniques et de recherche	96
B.11	Mémoires	98
B.12	Mémoires - encadrement	98
B.13	Divers	98

x Table des matières

C BIBLIOGRAPHIE 99

TABLE DES FIGURES

FIGURE 1	Du logiciel vers le matériel	12
FIGURE 2	Le retour du logiciel	13
FIGURE 3	L'évolution du logiciel vers la complexité	14
FIGURE 4	Agent Jooflux pour la JVM	20
FIGURE 5	Interaction via JMX avec une application modifiée par JooFlux	22
FIGURE 6	Micro-tests pour la comparaison entre Jooflux et plates-formes AOP	22
FIGURE 7	Micro-tests pour la comparaison entre Jooflux et langages dynamiques	23
FIGURE 8	Macro-tests de Jooflux avec SCImark2.0	24
FIGURE 9	Invocation contextuelle de méthode avec ConGolo	25
FIGURE 10	Architecture d'AxSeL - Construction d'un descripteur commun	31
FIGURE 11	Projection du contexte sur le graphe de services / composants AxSeL	32
FIGURE 12	Coloration du graphe de services / composants AxSeL	33
FIGURE 13	Performances d'AxSeL en fonction du nombre de nœuds du graphe	34
FIGURE 14	Architecture générale d'ACOMMA	36
FIGURE 15	Transformation automatique des appels de méthodes en services dans ACOMMA	36
FIGURE 16	Problème du plus court chemin appliqué dans ACOMMA pour la décision d'exécution distante	37
FIGURE 17	Intégration spontanée de services avec MySIM	40
FIGURE 18	Méthodes d'intégration avec MySIM	40
FIGURE 19	Architecture de MySIM	41
FIGURE 20	Correspondance syntaxique et sémantique d'interfaces de services dans MySIM	42
FIGURE 21	Correspondance de qualité de service non-fonctionnelle dans MySIM	42
FIGURE 22	Temps d'exécution et consommation mémoire de l'équivalence syntaxique dans MySIM	43
FIGURE 23	Temps d'exécution et consommation mémoire de l'équivalence sémantique dans MySIM	44
FIGURE 24	Temps d'exécution et consommation mémoire du calcul de degré QoS dans MySIM	44
FIGURE 25	Pri-REIN - exemple	46
FIGURE 26	Problème géométrique du point d'inclusion dans REIN	46
FIGURE 27	Ajout de la priorité dans la liste d'abonnés Pri-REIN	47

FIGURE 28	Performance de REIN en fonction du nombre de souscriptions	48
FIGURE 29	Performance de REIN en fonction du nombre de contraintes	48
FIGURE 30	Ajout de la priorité avec Pri-REIN	49
FIGURE 31	Architecture de CANDS	49
FIGURE 32	Partitionnement de graphe dans CANDS	50
FIGURE 33	Evaluation de performance des requêtes SSSP dans CANDS	52
FIGURE 34	Evaluation de performance des requêtes RSP dans CANDS	53
FIGURE 35	Echange de cartes de phéromones dans un réseau véhiculaire	56
FIGURE 36	Vers du logiciel éthique	71

LISTE DES TABLEAUX

TABLE 1	Résultats des jeux de tests de ConGolo avec des appels directs	27
TABLE 2	Résultats des jeux de tests de ConGolo avec des envois d'évènements	28
TABLE 3	Performances d'ACOMMA sur des micro-benchmarks	38
TABLE 4	Performances d'ACOMMA sur des macro-benchmarks	38
TABLE 5	Performances d'ACOMMA avec cache collaboratif	38
TABLE 6	Comparaisons de modèles de prise de décision pour le guidage en ville - scénario de trafic classique	60
TABLE 7	Comparaisons de modèles de prise de décision pour le guidage en ville - scénario d'accident dans le trafic	60
TABLE 8	Comparaisons de modèles de prise de décision pour le guidage en ville - scénario de terrorisme	62
TABLE 9	Comparaisons de modèles de prise de décision pour le guidage en ville - scénario de catastrophe naturelle	63

LISTE DES ALGORITHMES

- LISTING 1 Paramètres JMX pour changement de site d'appel
par Jooflux [22](#)
- LISTING 2 Déclaration du contexte en ConGolo [26](#)
- LISTING 3 Déclaration d'un objet contextuel en
ConGolo [26](#)
- LISTING 4 Déclaration du service de décision global en
ConGolo [27](#)
- LISTING 5 Modifications dynamiques de la sensibilité au
contexte et du service de décision local en ConGolo [27](#)

Première partie

CONTEXTE

INTRODUCTION

*We have seen that computer programming is an art,
because it applies accumulated knowledge to the world,
because it requires skill and ingenuity, and especially
because it produces objects of beauty.*

— Donald E. Knuth [Knuth 1974]

Ce manuscrit d'Habilitation à Diriger des Recherches présente un aperçu des recherches que j'ai mené dans le domaine du logiciel - et plus particulièrement des intergiciels appliqués aux environnements ambiants - sur une période de treize ans, de 2003 - date d'obtention de mon doctorat, jusqu'en 2016.

1.1 PARCOURS

Financé par la région Bretagne, j'ai effectué ma thèse dans l'équipe INRIA SOLIDOR (CONSTRUCTION DE SYSTÈMES ET D'APPLICATIONS DISTRIBUTI-
BUÉS)¹ à l'IRISA, UNIVERSITÉ DE RENNES 1 sous la direction de FRAN-
ÇOISE ANDRÉ. Mon sujet de recherche se focalisait au départ sur les
systèmes distribués et je l'ai orienté - par choix scientifique et par tribu-
lations et aléas d'une thèse - également vers les intergiciels et l'informa-
tique mobile. Maître Assistant Associé² à l'ÉCOLE DES MINES DE NANTES
dans l'équipe OCM³, je poursuis sur mes thématiques, soutiens ma thèse
en 2003 et en publie les résultats [Le Mouël et al. 2002, Le Mouël 2003].
Je suis recruté en septembre 2004 à l'INSA DE LYON. Je rejoins le départe-
ment TÉLÉCOMMUNICATIONS, SERVICES ET USAGES (TC) pour l'enseigne-
ment et le laboratoire CENTRE D'INNOVATION EN TÉLÉCOMMUNICATIONS
ET INTÉGRATION DE SERVICES (CITI) pour la recherche. Le laboratoire
CITI est tout nouvellement créé depuis 2000 et est en pleine définition
de ces axes de recherche. J'intègre l'équipe INRIA ARES (ARCHITECTURE
DE RÉSEAUX DE SERVICES), l'axe thématique "Middleware" étant alors
constitué seulement de STÉPHANE FRÉNOT. Nous allons alors constituer
à deux cet axe de 2004 à 2008. Durant cette période, je me focalise
et approfondis les enjeux scientifiques des systèmes adaptatifs, de la
composition de services, et de la modélisation du contexte pour l'in-
formatique ambiante. L'organisation des équipes INRIA s'effectuant sur
des cycles de 4 ans et ces enjeux scientifiques prenant de l'ampleur,
nous démarrons alors sur cette thématique une nouvelle équipe INRIA
AMAZONES (ARCHITECTURES SÉCURISÉES ORIENTÉES-SERVICES POUR RÉ-
SEAUX AMBIANTS). Celle-ci dure jusqu'en 2012 et se poursuit ensuite
sur deux domaines applicatifs différents : la géopolitique des données
avec l'équipe INRIA DICE (DONNÉES DE L'INTERNET AU CŒUR DE L'ÉCO-
NOMIE) – codirigée par STÉPHANE GRUMBACH et STÉPHANE FRÉNOT –
et l'Internet des Objets avec l'équipe DYNAMID (DYNAMIC SOFTWARE

1. La suite thématique de l'équipe INRIA SOLIDOR fût l'équipe INRIA ACES et est
maintenant l'équipe INRIA TACOMA.

2. Maître Assistant Associé est l'équivalent d'ATER au Ministère de l'Industrie.

3. La suite thématique de l'équipe OCM est maintenant l'équipe INRIA ASCOLA.

AND DISTRIBUTED SYSTEMS FOR THE INTERNET OF THINGS) que je dirige depuis 2012. Dans l'équipe DYNAMID, je promeus et développe l'idée que le futur est à l'intégration autonome de systèmes - en particulier pour les environnements fortement dynamiques (réseaux véhiculaires, Internet des Objets, informatique de bordure) et travaille sur les enjeux de passage à l'échelle, de prise de décision distribuée et de système à longue durée de vie.

Dès 2004, je participe à des ateliers Franco-Chinois sur les télécommunications. A partir de 2010, je commence à donner des cours dans le programme SPE-T (SPECIFIC PROGRAM OF ENGINEERING IN TELECOMMUNICATION), un programme conjoint entre l'INSA DE LYON, la prestigieuse UNIVERSITÉ DE SHANGHAI JIAOTONG et l'EM de Lyon fondé par FABRICE VALOIS et JIALIANG LU. Séduit, et de plus en plus impliqué dans le programme, j'en prends la direction en 2013 et effectue une année de CRCT en tant que professeur invité à SJTU. De cette expérience naît des projets de recherche fructueux avec JIAN CAO. J'y découvre une culture et un mode de travail totalement différent dont je m'évertue à savamment mélanger les bonnes influences occidentales et asiatiques au sein de l'équipe DYNAMID.

1.2 CONTEXTE DE RECHERCHE

Obtenir des résultats est, certes, le fruit d'un travail individuel mais ne peut s'effectuer sans efforts collectifs avec le support financier de projets, l'encadrement et les collaborations de doctorants et la reconnaissance par la dissémination scientifique. Dans les sections suivantes, je détaille quelques uns des points marquants qui ont construit mon évolution scientifique.

PROJETS DE RECHERCHE

AMIGO - AMIGO (*Ambient Intelligence for the Networked Home Environment*) est un projet Européen IP 7 auquel j'ai eu la chance de participer directement en arrivant au CITI. De 2004 à 2008, je participe à de nombreux groupes de travail et notamment dirige le Workpackage 3.4 sur "*Programming and Deployment Framework for Amigo Services*" avec une forte montée en compétence sur OSGi. Ce projet me familiarise avec une forte production de livrables et publications [Ramparany et al. 2005, Ben Mokhtar et al. 2005, Ben Mokhtar et al. 2006b, Ben Mokhtar et al. 2006a, Roussaki et al. 2006, Ahler et al. 2006][Jouve et al. 2007] et notamment m'a permis de financer les thèses de NOHA IBRAHIM [Ibrahim 2008] et AMIRA BEN HAMIDA [Ben Hamida 2010].

KAA & PRIAM - Les projets KAA (*Knowledge Authentication Ambient - Multi Disciplinary Project for Trust Management Scheme in Spontaneous Networks*) (ANR ACI, 2004-2007) et PRIAM (*Privacy Issues and Ambient intelligence*) (ARC INRIA, 2007-2009) sont des projets plus en marge de ma discipline principale. Leurs contributions scientifiques ne sont pas couvertes dans ce document. Pluridisciplinaires par nature, ces projets m'ont ouvert vers les disciplines des sciences économiques et du droit et m'ont permis d'appréhender les enjeux de la sécurité et le respect de la vie privée - enjeux devenus majeurs actuellement - et de les appliquer aux environnements ambiants [LB et Le Mouël 2007][Bryce et al. 2007].

ARC7 - Très impliqué dans l'action régionale ARC 7 sur "Territoires et Mobilité", j'y propose un projet sur la mesure de mobilité urbaine avec cartographie des services citoyens (*Mesuring the Urban Mobility for a Citizenship Service Cartography*, 2012-2015) en collaboration avec l'agence d'urbanisme de Lyon. Ce projet est l'occasion d'une recherche conjointe entre systèmes distribués et réseaux avec HERVÉ RIVANO de l'équipe INRIA URBANET. Ce projet a financé notamment la thèse de TRISTA LIN [Lin 2015].

COOPERA - Ce projet est le fruit d'une préparation de deux ans pour collaborer avec la Chine. Il développe le thème de la place des technologies de l'information dans la ville intelligente (*Smart Cities and IT Technologies for a Sustainable Development*, 2013-2015). Projet de la région Rhône Alpes - région jumelée avec la région de Shanghai, il me permet d'enrichir d'une collaboration recherche le partenariat enseignement que nous avons déjà avec l'UNIVERSITÉ DE SHANGHAI JIAOTONG. A l'aide d'une bourse de mobilité Explora'Pro de la région Rhône Alpes, j'y effectue une année de CRCT en tant que professeur invité. Ce projet a produit des publications majeures [Yang et al. 2014, Qian et al. 2015b, Qian et al. 2015a] et la mise en place d'un double diplôme de doctorat entre l'INSA DE LYON et SJTU.

VALEO - Contacté par l'équipement automobile VALEO en 2012, ce projet part de l'idée de mettre en place un boîtier dans un véhicule - boîtier qui permettrait le déploiement de services (*Service Deployment and Invocation under Delay Constraints in Vehicular Networks* - 2012-2016). Contractualisé bilatéralement entre INSAVALOR⁴ et VALEO, ce projet prend plus d'ampleur que prévu initialement, est prolongé, et élargit son spectre à de l'algorithmique distribuée pour de la prise de décision et de l'optimisation dans les réseaux véhiculaires. Ce projet a financé notamment la thèse de MARIE-ANGE LÈBRE [Lèbre 2016].

THÈSES SOUTENUES

J'ai eu la chance de pouvoir encadrer, guider et faire soutenir 5 doctorants [Ibrahim 2008, Ben Hamida 2010, Lin 2015, Lèbre 2016, Golchay 2016]. A mon arrivée au CITI en 2004, STÉPHANE UBÉDA - alors directeur du CITI - me laisse carte blanche pour définir un sujet de thèse. NOHA IBRAHIM fait alors une candidature spontanée et va parfaitement correspondre au profil et démarrer sur le projet Européen AMIGO. Avec un très bon dossier scientifique, suivi de deux postdocs, NOHA est maintenant Maître de Conférences au LIG, à l'INPG à Grenoble. Avec la montée en puissance du projet AMIGO, je décide un an plus tard de démarrer une seconde thèse. Je donne alors des cours dans le master de Jendouba en Tunisie et vais recruter AMIRA BEN HAMIDA - étudiante major du master. AMIRA effectue une bonne thèse, se tourne vers l'industrie chez PETALSLINK et LINAGORA en tant qu'ingénieur R&D, et est maintenant chargé de projet à l'institut SYSTEMX à Paris. Ces deux thèses vont clairement grandement consolider ma recherche jusqu'en 2010. En 2011, grâce aux nouveaux projets et contrats obtenus, je vais accepter en thèse ROYA GOLCHAY - étudiante du master RTS⁵ de Lyon.

4. INSAVALOR est la filiale valorisation de droit privé de l'INSA DE LYON.

5. Le master RTS est le master Réseaux, Télécom et Services de l'Université de Lyon.

ROYA effectue sa thèse de la manière la plus déterminée et courageuse que je connaisse. Rentrée en Iran, elle est maintenant en passe de décrocher un poste de maître assistant. En 2012, je vais démarrer simultanément l'encadrement de MARIE-ANGE LÈBRE - étudiante du master Systèmes Complexes de l'ENS - et le co-encadrement de TRISTA LIN - étudiante du master RTS. MARIE-ANGE et TRISTA effectuent deux thèses de très haut niveau scientifique. TRISTA est actuellement ingénieur architecture réseau et connectivité chez Assystem à Lyon. MARIE-ANGE est actuellement en post-doc à l'IUT Le-Puy-en-Velay. Le mérite de ma recherche passe forcément par le mérite de NOHA - avec qui j'ai posé les bases des intergiciels ambiants, par le mérite d'AMIRA et ROYA - avec qui j'ai approfondi le domaine scientifique, et par le mérite de TRISTA et MARIE-ANGE - avec qui j'ai ouvert le domaine scientifique vers de nouveaux domaines applicatifs.

DISSÉMINATION SCIENTIFIQUE

ATELIERS - Pendant 4 ans, j'organise et anime - avec STÉPHANE FRÉNOT - l'atelier international IEEE *Services Integration in Pervasive Environments* [Le Mouël et Frénot 2006, Le Mouël et Frénot 2007, Le Mouël et Frénot 2008, Le Mouël et Frénot 2009]. Ces ateliers permettent alors de fédérer autour du croisement thématique des Architectures Orientées-Services et de l'Informatique Ambiante. Maintenant, beaucoup de conférences se positionnent sur ces thématiques mais, à ce moment, les conférences "Mobile Computing" se focalisaient sur le réseau et les conférences "Middleware" sur les grilles et le pair-à-pair. Je me souviens de ce que m'avait dit SOFIE VAN HOECKE - Assistant Professor à GHEENT UNIVERSITY - en 2007 : "Cool l'atelier, heureusement, nous ne savions pas où publier!". Ces ateliers permettent également à nos doctorants de participer à l'organisation, de découvrir le système de relecture, et de publier sans leurs encadrants [Parrend et al. 2006].

OPEN* - La production de logiciels en OpenSource et de données en OpenData est un point de dissémination que je considère important et que nous essayons de promouvoir dans l'équipe DYNAMID. Le logiciel Jooflux est disponible sous licence Mozilla - co-déclaration d'invention avec JULIEN PONGE [Ponge et Le Mouël 2012]⁶. Le langage de programmation Golo - dont je ne suis que contributeur - est disponible à la fondation Eclipse sous licence Eclipse Public [Ponge et al. 2013]⁷ et ConGolo [Maingret et al. 2015] - adaptation contextuelle du langage - se présente comme une branche également disponible publiquement⁸. Le logiciel d'analyse multi-thread des traces des taxis de Shanghai est disponible sous licence Apache⁹. Et enfin, la trace de mobilité véhiculaire du rond-point de Créteil est disponible sous licence Creative Commons [Lèbre et al. 2015a]¹⁰.

6. <https://github.com/dynamid/jooflux>

7. <https://github.com/eclipse/golo-lang>

8. <https://github.com/bmaingret/contextual-golo-lang>

9. <https://github.com/flemouel/taxi-trace>

10. <http://vehicular-mobility-trace.github.io>

1.3 ORGANISATION DU DOCUMENT

La suite du manuscrit est organisée comme suit :

- Le chapitre 2 présente ma vision de l'évolution de la complexité du logiciel, en particulier son retour au premier plan d'importance par rapport à l'évolution du matériel et des domaines d'applications. Les chapitres suivants suivent cette ligne rouge et détaillent pour chaque étape importante la contribution qui est effectuée - contribution qui y est présentée de manière collective.
- Le chapitre 3 présente les travaux effectués au niveau de la Machine Virtuelle Java (*Java Virtual Machine* - JVM) pour intégrer des mécanismes rendant les applications dynamiquement adaptables à un environnement changeant.
- Le chapitre 4 présente les travaux réalisés au niveau des Architectures Orientées-Services (*Service-Oriented Architecture* aka SOA) premièrement pour charger et décharger des services dans des environnements d'exécution contraints et deuxièmement pour composer automatiquement et spontanément des services en fonction d'un contexte hétérogène.
- Le chapitre 5 présente les travaux réalisés au niveau des Intergiciels Orientés-Messages (*Message-Oriented Middleware* aka MOM) pour rendre les intergiciels de plus en plus distribués, performants et passant à l'échelle.
- Le chapitre 6 présente enfin les derniers travaux au niveau de l'algorithme sociale (*Social Algorithms*) pour avoir des intergiciels autonomes, particulièrement adaptés et optimisés pour des cas d'utilisation de la société et de la vie courante.
- Les chapitres 7 et 8 concluent ce document et donnent des perspectives à ces travaux et aux évolutions que j'estime possibles pour les intergiciels.
- Les annexes A, B et C contiennent un curriculum vitæ, ma liste de publications et la bibliographie.

*However,
as every parent of a small child knows,
converting a large object into small fragments
is considerably easier than the reverse process.*

— Andrew Tanenbaum [Tanenbaum 2002]

La thématique commune sur laquelle je travaille est le logiciel (*Software*) - et plus particulièrement les intergiciels (*Middleware*) appliqués aux environnements ambiants (*Ambient Intelligence*). Je vais en premier lieu donner trois définitions de ces termes pour clarifier la suite de ce chapitre.

LOGICIEL : “Ensemble de séquences d’instructions - appelé programme - interprétables par une machine - physique ou virtuelle - ou par plusieurs machines, et d’un jeu de données nécessaires à ces opérations. ”

Cette définition - une extension de [Wikipedia 2016b] - capture, par la notion de programme, les domaines des langages de programmation, les systèmes d’exploitation, les intergiciels et les applications. Elle se veut plus généraliste en précisant qu’une machine peut être physique ou virtuelle. Le terme machine se définissant comme un produit fini mécanique, il est souvent utilisé en informatique pour désigner la partie matérielle d’un ordinateur (*Hardware*). En étendant la définition aux machines virtuelles - dans le sens d’une machine de Turing, cette définition permet de capturer les approches actuelles d’informatique dans les nuages (*Cloud Computing*) et virtualisation (*Virtualization*). En étendant la définition à une exécution sur plusieurs machines, cette définition permet également d’englober les systèmes distribués (*Distributed Systems*).

INTERGICIEL : “Logiciel tiers permettant d’abstraire, de publier et d’interconnecter des services pour échanger et traiter des informations.”

Les intergiciels sont depuis le début vus comme une couche d’abstraction entre le système d’exploitation et l’application - i.e. une couche de “glue” dans l’ingénierie d’application ou un “passe-plat” dans les systèmes distribués [Naur et Randell 1968, Gall 2005]. Dans ma définition, je fais évoluer cette vision en retirant les concepts de systèmes d’exploitation et d’application car je considère que les intergiciels ne sont pas propres à une couche particulière. La notion d’abstraction est importante car elle permet de définir des interfaces de programmation formalisant la publication et l’interconnexion des fonctionnalités. Cela permet de capturer les domaines des architectures orientées-services (*Service-Oriented Architecture*), des intergiciels orientés-messages (*Message-Oriented Middleware*) et des implantations

comme les services Web (*WebServices*). J'ajoute la notion de traitement car je considère que la grande valeur ajoutée des intergiciels est cette capacité à prendre des décisions de traitements pertinents en fonction notamment du contenu des informations. Cela permet de capturer les domaines des intergiciels centrés-contenu (*Content-Centric Middleware*) et plus récemment les intergiciels intelligents [Pease et al. 2016][Cisco et Frost & Sullivan 2008, Motorola 2016]. Par analogie avec un réseau informatique - ensemble d'équipements reliés entre eux pour échanger des informations [Wikipedia 2016c], je considère un intergiciel comme un réseau logiciel actif.

INTELLIGENCE AMBIANTE : "Vision de la société de l'information mettant l'accent sur un support efficace, ergonomique, de bien-être répondant aux besoins de l'homme et la société, en toutes circonstances du temps et de l'espace, au travers de périphériques communicants, invisibles, non-intrusifs embarqués dans les objets du quotidien."

Cette vision est promue par Mark Weiser en 1991 [Weiser 1991], sera l'objet de plusieurs définitions une dizaine d'années plus tard [Satyanarayanan 2001][NIS 2001, ISTAG 2001] et commence maintenant à réellement être mise en œuvre. Dans cette définition, je reprends l'élément central classique qui est de mettre l'homme au centre des préoccupations. J'étends ces préoccupations à l'ensemble de la société - car cela correspond beaucoup plus aux enjeux actuels des villes intelligentes (*Smart Cities*) et réseaux sociaux (*Social Networks*). D'autres caractéristiques classiques se retrouvent dans la définition avec des périphériques communicants (*Embedded Systems, Sensor Networks, Internet of Things*), sensibles au contexte (*Context-Awareness*), répondant aux besoins (*Profiling, Adaptive and Proactive Systems*), accessibles de manière transparente partout (*Ubiquitous Computing*). L'aspect ubiquitaire avec un accès à n'importe quel moment devient actuellement un enjeu important avec les systèmes à longue durée de vie (*Long-living Systems*).

2.1 LE RETOUR DU LOGICIEL

Pour comprendre l'état actuel des intergiciels, il est important de regarder leur évolution. Je positionne Smalltalk-72 [Goldberg et Kay 1976, Shoch 1979] comme le père des intergiciels. Smalltalk-72 n'est pas à proprement parler un logiciel tiers mais un langage de programmation, le premier présentant simultanément les deux caractéristiques principales d'abstraction et d'interconnexion de la définition pré-citée. En effet, pour ce qui est de l'abstraction, Smalltalk-72 définit les concepts d'objet, de typage, d'encapsulation et polymorphisme qui seront les bases des langages de programmation orientés-objets (*Object-Oriented Programming*). Pour ce qui est des interconnexions, Smalltalk-72 définit la communication par passage de messages, qui sera la base des intergiciels orientés-messages et du modèle à acteurs. Ces mécanismes apparaissent légèrement avant, dans Simula-67 [Dahl 1968] pour les abstractions en classes et avec les appels de fonctions à distance (*Remote Procedure Call - RPC*) [White 1975] pour l'interconnexion, et Smalltalk-72 est le premier à combiner les concepts.

Ces deux aspects d'abstraction et d'interconnexion des intergiciels sont importants car ils vont évoluer différemment :

- La capacité d’abstraction va être influencée par les langages de programmation et l’ingénierie logicielle et va voir émerger (1) les modèles objets, composants [Cox 1986], services [WSAWG 2002] pour la création d’applications et (2) les paquetages, conteneurs [Kamp et Watson 2000, des Ligneris 2005] pour le déploiement d’applications. Ces éléments vont particulièrement caractériser les propriétés de *modularité* et de *réutilisabilité* des intergiciels.
- La capacité d’interconnexion va être influencée par les systèmes distribués et le web et va voir émerger (1) RPC, RMI (*Remote Method Invocation*) [Grosso 2001], XML-RPC [Laurent et al. 2001] avec SOAP [Box et al. 2000], REST [Fielding 2000] avec Web-API [Richardson et al. 2013] pour les appels de fonction à distance, et (2) les Stub/Skeleton, les architectures en bus (*Publish/Subscribe*), les transactions distribuées, la persistance, les annuaires pour le côté manipulation d’objets distribués. Ces éléments vont particulièrement caractériser les propriétés d’*ouverture* et de *fiabilité* des intergiciels.

Les deux premiers intergiciels en tant que logiciels tiers vont être CORBA (*Common Object Request Broker Architecture*) [OMG 1992] et DCOM par Microsoft (*Distributed Component Object Model*) [Horstmann et Kirtland 1997] dans les années 1990. CORBA va pleinement développer la modularité avec une approche à composant (*CORBA Component Model - CCM*) et un langage de description d’interface (*Interactive Data Language - IDL*) et également la fiabilité avec un bus de communication (*Object Request Broker - ORB*), le protocole d’interconnexion IIOP (*Internet Inter-ORB Protocol*), et un adaptateur côté serveur (*Portable Object Adapter - POA*) gérant le cycle de vie des composants avec gestion de la mémoire, redondance, équilibrage de charge. DCOM va également développer la modularité avec son approche à composant (*Component Object Model - COM*) et la fiabilité avec le protocole DCE/RPC (*Distributed Computing Environment/Remote Procedure Calls*) permettant de sérialiser/dé-sérialiser les paramètres des fonctions et de gérer un ramasse-miette distribué.

Ces deux premiers intergiciels vont néanmoins pêcher à développer la propriété d’ouverture avec des problèmes d’espace d’adressage pour localiser facilement des composants distants et des protocoles ne permettant pas de fonctionner au travers des pare-feux [Townsend et al. 2004, Henning 2006]. Avec de nombreux langages de programmation supportés (Ada, C, C++, C++11, COBOL, Java, Lisp, PL/I, Object Pascal, Python, Ruby par exemple par CORBA), plusieurs systèmes d’exploitation (Windows, AIX, HP/UX, Solaris, Linux, AS/400, z/OS, z/VM, BS2000/OSD pour EntireX DCOM), les principales leçons qui vont ressortir sont les capacités de modularité et réutilisabilité des intergiciels pour gérer l’hétérogénéité logicielle et matérielle.

Comme le montre la Figure 1, avec l’explosion des capacités matérielles [Moore 1965, Wikipedia 2016a], chaque élément logiciel va se retrouver intégré avec les systèmes-sur-puces (*System-on-Chip - SoC*), réseaux-sur-puces (*Network-on-Chip - NoC*) et la radio-sur-puces (*Radio-on-Chip - RoC*). De nombreux intergiciels vont être développés pour profiter de cette puissance, standardiser simultanément logiciel et matériel et ainsi pouvoir produire des puces en grande quantité [Grammatikakis et al. 2003, Wolf 2004, Paulin et al. 2006, Popovici et al. 2010].

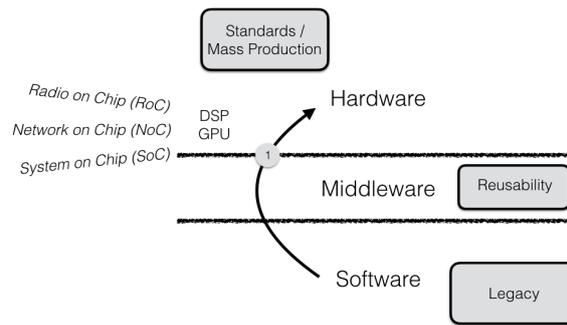


FIGURE 1 – Du logiciel vers le matériel

En 1995, Sun Microsystems met en œuvre le langage Java et la machine virtuelle Java (*Java Virtual Machine* - JVM) [Gosling 1995]. Du point de vue abstraction et réutilisation, la machine virtuelle Java est un intergiciel intéressant qui repose sur un bytecode intermédiaire, permet de ne pas avoir à gérer la mémoire et peut effectuer dynamiquement une compilation optimisée (*Just-In-Time compiler* - JIT). Mais, finalement, la machine virtuelle Java n'est pas révolutionnaire en soi et reste dans la lignée d'abstraction du matériel de l'époque. De mon point de vue, les ruptures technologiques apportées par Sun Microsystems vont être Jini en 1998 [Arnold 1999, Waldo 1999] et JNDI (*Java Naming and Directory Interface*) en 2000 [Lee et Seligman 2000]. En effet, le concept d'annuaire existait déjà avec les annuaires DNS, LDAP, NIS, mais ne permettait d'enregistrer que de l'information et Jini va permettre pour la première fois d'enregistrer du logiciel - un *stub* - associé à des propriétés de l'environnement - un *contrat* avec date d'expiration. JNDI va ensuite fournir une couche d'abstraction permettant de se positionner en interface avec n'importe quel annuaire. Ce sont ces deux approches qui vont donner naissance dans les années 2000 aux architectures orientées-services (*Service-Oriented Architecture* - SOA) [Erl 2005, Townsend 2008] qui pallieront aux défauts d'interconnexions des intergiciels. Popularisée par des intergiciels comme .Net [Puvvala et Pota 2003] ou JEE (*Java Enterprise Edition*) [Asbury et Weiner 2001], les architectures orientées-services vont (1) définir les services avec des descriptions sémantiques comme WSDL (*Web Services Description Language*) [Christensen et al. 2001] ou OWL-S (*Semantic Markup for Web Services*) [Martin et al. 2004], (2) permettre de publier ces services par exemple avec UDDI (*Universal Description Discovery and Integration*) [Walsh 2002], (3) offrir un couplage faible entre l'interface et l'implantation des services permettant ainsi de les remplacer et les composer comme avec BPEL (*Business Process Execution Language*) [Juric 2006] ou WS-Coordination [Newcomer et Robinson 2009].

Comme le montre la Figure 2, cette ouverture des intergiciels - combinée avec deux autres facteurs : le ralentissement de la conjecture de Moore sur le matériel (annoncée [Kumar 2015], réelle [Desai et al. 2016]?) et la demande de traitement et de passage à l'échelle des environnements ambiants [Stolpe 2016] - fait revenir les intergiciels au premier plan des enjeux logiciels. De nombreux élé-

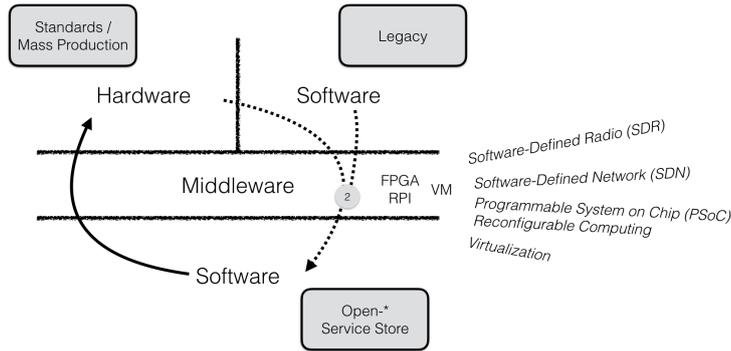


FIGURE 2 – Le retour du logiciel

ments qui se trouvaient dans le matériel trouvent leurs limites et/ou ont besoin de plus d'adaptabilité. Et, bien que ce soit des domaines de recherche démarrés depuis 10 à 20 ans, un nouvel élan et surtout de nouveaux domaines applicatifs apparaissent vers 2007-2008 pour les systèmes-sur-puces programmables (*Programmable System-on-Chip* - PSoC), la radio-logicielle (*Software-Defined Radio* - SDR), les réseaux-logiciels (*Software-Defined Network* - SDN). En 2006, Amazon introduit *Elastic Compute Cloud* (EC2) [Barr 2006] qui va grandement populariser l'informatique dans les nuages (*Cloud Computing*), notamment la virtualisation logicielle (*Software-as-a-Service* - SaaS). Largement popularisé par Google et Apple, cette omniprésence des intergiciels se retrouve même maintenant auprès du grand public avec l'informatique dans les nuages - comme Google Drive, Dropbox ou iCloud - et avec l'informatique mobile distribuée (*Mobile Cloud Computing*) - comme Android et iOS et leurs *Store* de services applicatifs.

2.2 HÉTÉROGÉNÉITÉ + ENVIRONNEMENTS DYNAMIQUES + PASSAGE À L'ÉCHELLE + AUTONOMIE = COMPLEXITÉ DES INTERGICIELS

De pair avec l'ouverture des intergiciels dans les années 2000, l'intelligence ambiante va prendre un grand essor et envisager les scénarios à horizon de 2020 [ISTAG 2003][Basten et al. 2003, Cook et al. 2009]. Je présente, dans la Figure 3, ma vision de cette évolution. Un des points intéressants de ces scénarios est qu'ils partent des besoins exprimés par les domaines d'applications (en haut à gauche de la Figure 3) pour envisager leurs impacts (en bas à droite de la Figure 3). Et ce faisant, permettent la mise en place des technologies matérielles et logicielles pour arriver à ces objectifs. Par exemple, les fournisseurs d'accès (*Internet Providers*) - ayant de forts besoins en gestion de l'hétérogénéité de machines - mirent en place des multi-standards au sein des passerelles domestiques (*Gateways*) - comme UPnP [Jeronimo et Weast 2003] ou DLNA [Heredia 2011]. L'industrie automobile et les industries des technologies (*IT Companies*) voyaient l'avenir en M2M (*Machine-to-Machine*) avec le déploiement de périphériques interconnectés à la maison (*Home Automation*) et à la voiture (*Connected Car*). Ces scénarios s'appuient dans un premier lieu principalement par le développement du maté-

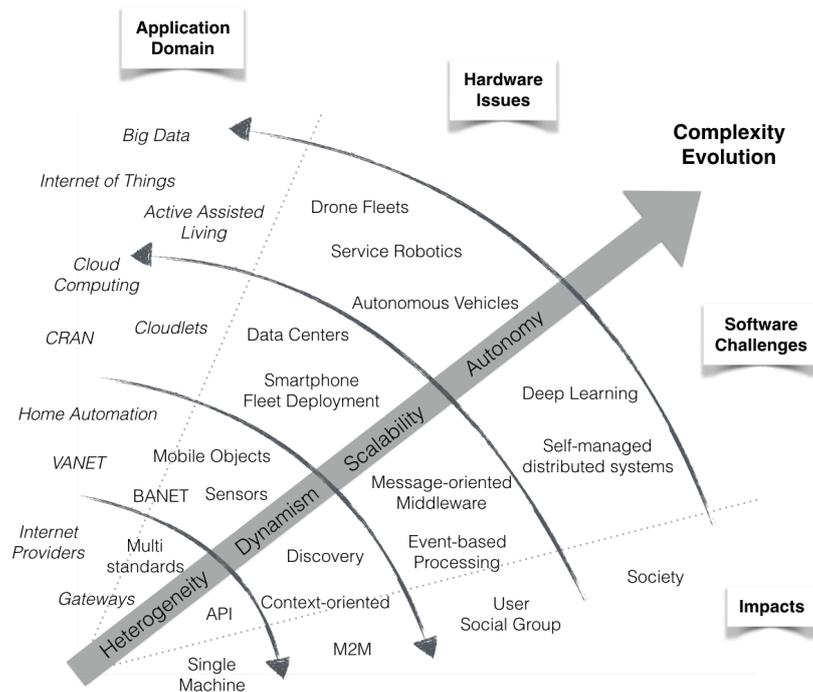


FIGURE 3 – L'évolution du logiciel vers la complexité

riel, le logiciel et les intergiciels ne venant que dans un second temps. Bien qu'étant censé être centré sur l'utilisateur, ces produits n'arrivent finalement que très peu au consommateur final.

En 2006-2007, deux domaines se démocratisent : (1) les smartphones - avec le premier iPhone en 2007 avec écran tactile, et surtout ses applications téléchargeables, et (2) les réseaux sociaux - avec Facebook en 2006. Ces deux révolutions vont réellement mettre l'utilisateur au centre des préoccupations. Depuis lors, les scénarios de l'intelligence ambiante partent des impacts sur l'utilisateur ou sur la société pour construire les domaines applicatifs - souvent d'ailleurs sans bien mesurer les applications qui vont bien pouvoir en découler - comme avec l'Internet des Objets (*Internet of Things*) ou la masse de données accumulées (*Big Data*). Ces nouveaux scénarios présentent de nouveaux défis. En plus de la gestion de l'hétérogénéité, ces scénarios incluent des environnements fortement mobiles et contextuels (*Dynamism*), devant passer à l'échelle (*Scalability*) et pouvant prendre des décisions pour assister l'utilisateur (*Autonomy*). Le matériel ne présentant pas ces capacités d'adaptabilité, et les applications ne pouvant intégrer autant de caractéristiques sans en devenir ingérables, les intergiciels prennent actuellement une place prépondérante dans le développement de l'intelligence ambiante aussi bien scientifiquement [Katasonov et al. 2008, Reijers et al. 2013, O'Grady et al. 2016] qu'économiquement [Cisco et Frost & Sullivan 2008, Motorola 2016].

Les défis des intergiciels intelligents passent notamment par la programmation orienté-contexte (*Context-Oriented Programming*) [Hirschfeld et al. 2008] pour développer des applications sensibles à

leur environnement, les intergiciels à messages (*Message-Oriented & Content-Centric Middleware*) [Chen et al. 2008] pour effectuer du traitement sémantique des information pour disséminer de manière performante et ciblée à large-échelle, et les intergiciels autonomes (*Autonomic Computing*) [Kephart et Chess 2003] pour une auto-gestion distribuée avec apprentissage et prise de décision selon les situations. Je présente ce cheminement de mes recherches en traitant du dynamisme dans les chapitres 3 et 4, du passage à l'échelle dans le chapitre 5 et de l'autonomie dans le chapitre 6.

Deuxième partie

CONTRIBUTIONS

*To understand a program,
you must become both the machine and the program.*

— Alan J. Perlis [Perlis 1982]

Pour s'adapter à un environnement de plus en plus hétérogène et changeant, une des premières étapes est de proposer des logiciels ayant également la capacité de se modifier eux-mêmes. Les travaux présentés dans cette section abordent cela sous deux angles différents :

- Jooflux [Ponge et Le Mouël 2012, Ponge et Le Mouël 2012, Ponge et Le Mouël 2012b, Ponge et Le Mouël 2012a, Ponge et Le Mouël 2013], en section 3.1, est une approche orientée sur la couche intergicielle qui modifie les applications lors de leur chargement pour les rendre automatiquement dynamiquement adaptables,
- ConGolo [Maingret et al. 2015], en section 3.2, est une approche orientée sur les langages de programmation qui permet d'exprimer simplement - à la conception de l'application - un contexte et comment y réagir, y compris si une situation non prévue se présente.

3.1 JOOFLUX

OBJECTIFS La plupart des applications sont - jusqu'à un certain point - statiques. Cela est particulièrement vrai pour les applications développées en utilisant des langages statiquement typés. Les langages typés dynamiquement offrent souvent la possibilité de remplacement de méthodes pendant l'exécution, mais ces changements n'interviennent que depuis le programme lui-même et rarement d'outils externes. Très peu de langages offrent cet accès extérieur, même les langages purement fonctionnels avec un shell interactif. Les modifications à chaud de code ont pourtant de nombreuses et intéressantes vertus [Frieder et Segal 1991, Hicks et Nettles 2005]. Pendant les phases de développement, un temps important est passé et perdu à redémarrer les applications pour observer les effets des modifications du code source. En production, l'application de correctifs d'erreurs ou de sécurité requiert des redémarrages. Un redémarrage complet peut prendre plusieurs minutes pour être opérationnel quand, dans de nombreux cas, une simple correction consiste juste à remplacer l'implémentation d'une méthode sans introduire d'effets de bord comme un changement de signature. L'ajout et le retrait dynamiques de fonctionnalités transversales, comme des aspects de sécurité ou de trace, peuvent également être très utiles pour capturer ou surveiller ponctuellement certains comportements de l'application.

Nous proposons une nouvelle approche pour modifier "à chaud" des méthodes et injecter des aspects pendant l'exécution d'applications dans le cadre spécifique de la machine virtuelle Java (JVM). Nos contributions s'articulent comme suit : (1) Nous avons conçu et développé un agent

Java qui intercepte et modifie le bytecode pendant l'exécution pour remplacer toutes les invocations de méthodes par le nouvel opcode `invokedynamic` introduit dans Java 7 [Gosling et al. 2013], (2) Nous proposons une API de contrôle accessible par un agent JMX qui permet de gérer ces modifications pendant l'exécution et ce pour pouvoir se dispenser d'un langage dédié ou d'insertion d'annotations ou de points de coupe dans le code source, et (3) Nous présentons une comparaison entre notre prototype JooFlux et plusieurs plates-formes à aspect et langages dynamiques existants. Les résultats mettent en exergue que les performances de JooFlux sont proches de celles de Java - avec un coût marginal et même quelques fois un gain - tandis que les plates-formes à aspects et langages dynamiques présentent des surcoûts importants.

CONTRIBUTION ET RÉSULTATS JooFlux est implanté par un agent Java. Il est possible d'attacher des agents à une JVM. Ils ont la possibilité d'effectuer diverses opérations dont l'interception et transformation de bytecode au moment où celui-ci va être chargé en mémoire dans la JVM. Si JooFlux cible principalement le bytecode émis par un compilateur pour le langage Java, il peut théoriquement s'appliquer sur n'importe quel bytecode valide produit par un langage autre que Java sur la JVM tel que Scala [Odersky et Zenger 2005].

JooFlux introduit une indirection sur les appels de méthode, de sorte que le remplacement de méthode et l'application d'aspects puissent se faire à l'exécution. Le fait de pouvoir modifier les références de méthodes avec `invokedynamic` avec de nouvelles chaînes de combinateurs permet de changer les implémentations de méthodes. Le jeu de combinateurs disponibles nous permet également de greffer des traitements supplémentaires pré et post invocation sur les arguments et valeurs de retour. Cela nous permet de faire de la programmation par aspects.

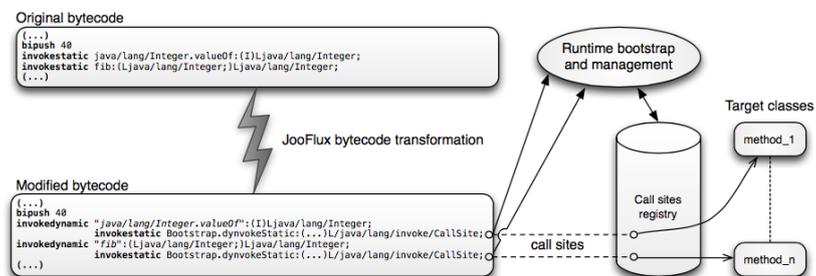


FIGURE 4 – Agent JooFlux pour la JVM

La Figure 4 donne un aperçu du fonctionnement de JooFlux. Il cherche les occurrences de `invokestatic`, `invokevirtual`, `invokespecial` ou `invokeinterface` puis les remplace par une construction sémantiquement équivalente basée sur `invokedynamic`. Les sites d'appels sont liés à une référence de méthode au moment de l'exécution. Les transformations de bytecode s'opèrent à l'aide de la bibliothèque ASM [Bruneton et al. 2002]. Initialement, les sites d'appels sont liés à des références de méthodes pointant sur les méthodes des classes originelles. Ainsi pour un `invokestatic` dans le bytecode, un remplacement est fait avec `invokedynamic` et un nom symbolique basé sur la signature de l'instruction `invokestatic` remplacée. Ceci permet d'avoir un schéma de nommage uniforme. Le type de l'invo-

cation originelle est également préservé. Coté JooFlux, l'initialisation de sites d'appels se fait via un jeu de méthodes statiques spécialisées en fonction du type d'invocation initiale. Ceci introduit une couche d'indirection fine.

Aspects via des combineurs de références de méthodes Une fois les classes transformées par JooFlux et chargées dans la machine virtuelle, JooFlux n'effectue plus de modifications de bytecode. L'injection se fait au niveau des chaînes de combineurs référencées par les sites d'appels. JooFlux utilise 2 combineurs que l'on trouve dans la classe `java.lang.invoke.MethodHandles` :

1. `filterArguments` prend pour argument une référence de méthode cible, un index d'argument et un tableau de références de méthodes à utiliser en filtres, et
2. `filterReturnValue` prend pour argument une référence de méthode cible et une référence de méthode filtre.

Les types des références de méthodes filtres doivent correspondre à ceux des méthodes cibles. Dans notre cas nous avons choisi une approche générique pour qu'un aspect soit agnostique de la signature de la méthode interceptée. Dans le cas des appels de méthode les arguments sont capturés dans un tableau d'objets, et dans le cas des retours un objet est employé.

Plus généralement, les aspects peuvent faire d'autres choses comme lever une exception si un argument ne respecte pas une pré-condition. Étant donné qu'ils agissent comme des filtres, ils peuvent aussi modifier les arguments avant de les passer à la méthode cible ainsi que modifier la valeur de retour. Les aspects peuvent être empilés sur un même site d'appel. Par exemple, il est possible d'empiler des aspects de validation et de trace.

Piloter JooFlux pour remplacer à chaud des méthodes et injecter des aspects La couche de gestion de JooFlux utilise le registre d'appels de diverses façons via un agent JMX [Perry 2002] pour effectuer des requêtes. Cet agent offre les opérations distantes suivantes :

1. remplacer les implémentations de méthode, et
2. injecter un aspect avant ou après certains sites d'appel, et
3. obtenir diverses métriques telles le nombre de sites d'appels gérés par JooFlux ou leur liste.

À titre d'exemple, voyons l'interaction pour un remplacement de méthode via l'interface de l'agent JMX.

La Figure 5 expose une application avec un bouton à gauche et un texte à droite. En se connectant à l'agent JMX, JooFlux - via un outil comme `jconsole`, peut remplacer le gestionnaire d'évènement de bouton de sorte qu'au prochain clic, une image remplace le texte. Dans ce cas précis, nous avons utilisé les paramètres suivants pour l'opération `changeCallSiteTarget` de l'agent JMX :

Comparaison expérimentale avec des plates-formes AOP et des langages dynamiques Pour valider notre approche, nous comparons les performances de la machine virtuelle Java avec notre agent JooFlux à d'autres approches effectuant également de la redirection dynamique

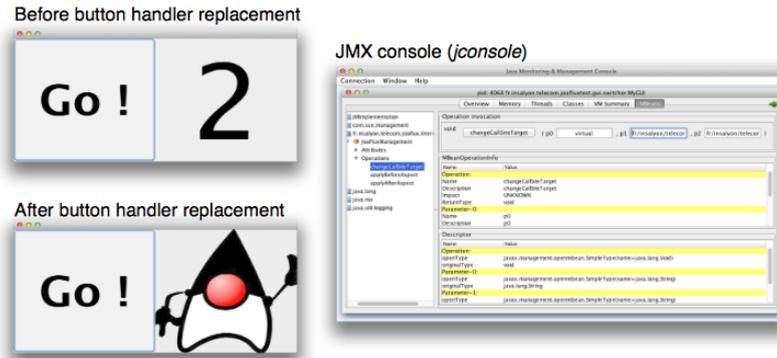


FIGURE 5 – Interaction via JMX avec une application modifiée par JooFlux

```

1 virtual, // virtual method
2 fr/insalyon/telecom/joofluxtest/gui/switcher/MyActionListener.
  counterIncrement:(MyActionListener)void, // old handler id
3 fr/insalyon/telecom/joofluxtest/gui/switcher/MyActionListener.
  pictureSwitch:()V // new handler id

```

LISTING 1 – Paramètres JMX pour changement de site d'appel par JooFlux

de méthodes sur la machine virtuelle : les plates-formes à aspects et les langages de programmation dynamiques.

Nous comparons la fonctionnalité d'injection d'aspects de JooFlux avec deux plates-formes AOP : AspectJ [Kiczales et al. 2001] et ByteMan [Dinn 2011]. Nous utilisons le calcul classique d'une fonction Fibonacci récursive à laquelle nous injectons avant et/ou après un aspect vide – c.à.d. une redirection d'appel vers une méthode vide. La Figure 6 présente les résultats.

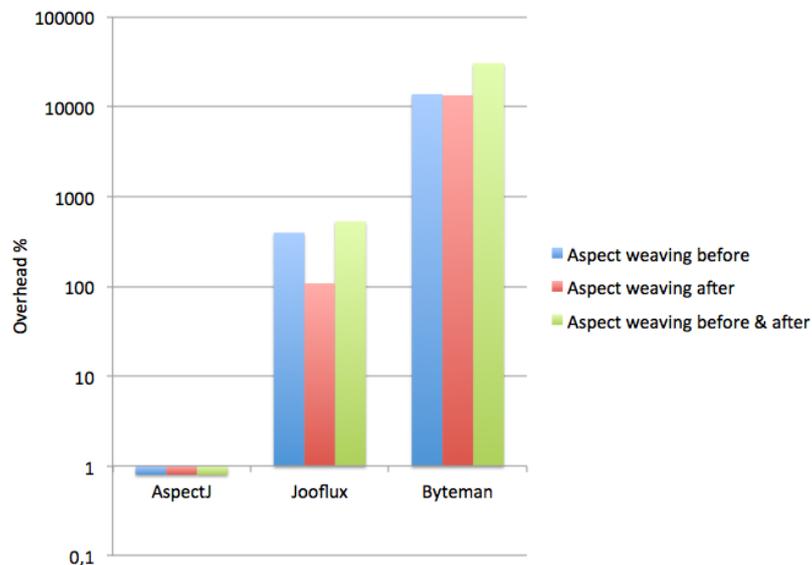


FIGURE 6 – Micro-tests pour la comparaison entre JooFlux et plates-formes AOP

Le temps d'exécution du bytecode généré par AspectJ est proche de celui de Java, seulement il tisse les aspects statiquement pendant la compilation et plus aucune modification ne peut être effectuée à

l'exécution. ByteMan permet l'injection d'aspects pendant l'exécution en déchargeant la classe et modifiant son bytecode pendant son rechargement. Cette technique dégrade significativement les performances et implique que toutes les optimisations JIT sont perdues. Le bytecode généré par JooFlux contient déjà les appels dynamiques, donc le coût de tissage d'un aspect est seulement le coût d'ajout d'un nouveau combineur dans la chaîne d'un site d'appel, de recopier les arguments et de transférer la valeur de retour. En conservant la chaîne d'appel à peu près intacte, JooFlux sauvegarde les optimisations JIT.

De nombreux langages de programmation proposent la redirection dynamique de méthodes directement dans le langage, par exemple avec les APIs de réflexion. Pour tester la fonctionnalité de redirection dynamique de méthodes de JooFlux, nous la comparons aux principaux langages dynamiques basés sur la machine virtuelle : Java [Oracle], Clojure [Clojure], JRuby [JRuby], Groovy [Groovy], Rhino JavaScript [Rhino Javascript] and Jython [Jython]. Nous testons la performance avec le même micro-jeu de test Fibonacci. Les résultats sont présentés dans la Figure 7. Même si l'implémentation fortement typée en Clojure présente seulement un facteur de ralentissement de 1.2-1.4, la plupart des autres langages sont de 3 à 18 fois plus lents que notre prototype JooFlux.

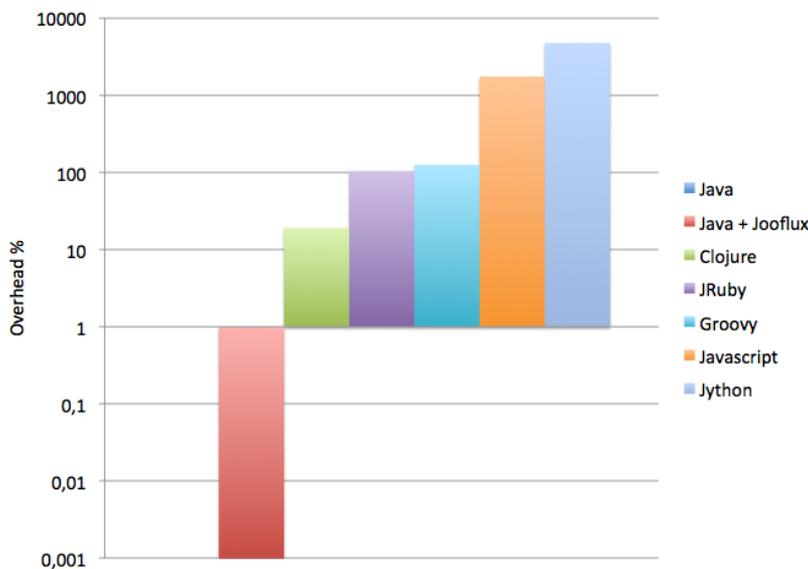


FIGURE 7 – Micro-tests pour la comparaison entre JooFlux et langages dynamiques

Pour tester plus intensément la redirection dynamique de méthodes avec JooFlux, nous avons utilisé 3 macro-jeux de tests : (1) SCImark 2.0 [Poza et Miller 2000] - jeu de tests en Java pour le calcul scientifique et numérique, assez consommateur en CPU et mémoire, et (2) un Fork / Join [Ponge 2011] - jeu de tests multi-threads synchronisés stressant particulièrement la mémoire et les entrées/sorties, et (3) l'implantation du langage Clojure au-dessus de JooFlux - implantation entièrement écrite en Java pour avoir un jeu de tests effectuant un nombre massif de réécritures de bytecode. La Figure 8 présente les résultats pour SCImark 2.0, les autres résultats sont présentés dans [Ponge et Le Mouél 2012]. SCImark 2.0 effectue des transformations de Fourier (Fast Fourier Transformations, FFT), résout des équations de Laplace (Jacobi Successive

Over-relaxation, SOR), approxime la valeur de π (Monte Carlo integration, MC), multiplie des matrices (Sparse matrix multiply, SM), effectue des factorisations de matrices denses (dense LU matrix factorization, LU) et calcule finalement un score composé (Composite Score, CS). Ces différents calculs sont réalisés sur un petit ou un grand jeu de données. La plupart des calculs présentent des sur-coûts ou gains marginaux, excepté pour l'intégration Monte Carlo qui perd 20% en performance. L'algorithme Monte Carlo utilise des fonctions synchronisées et de l'inclusion de fonctions (cf documentation de SCImark) et - les fonctions synchronisées n'impactant pas les performances comme le montre le test du Fork/Join, la perte provient de difficultés d'optimisations JIT du fait de l'inclusion de fonctions.

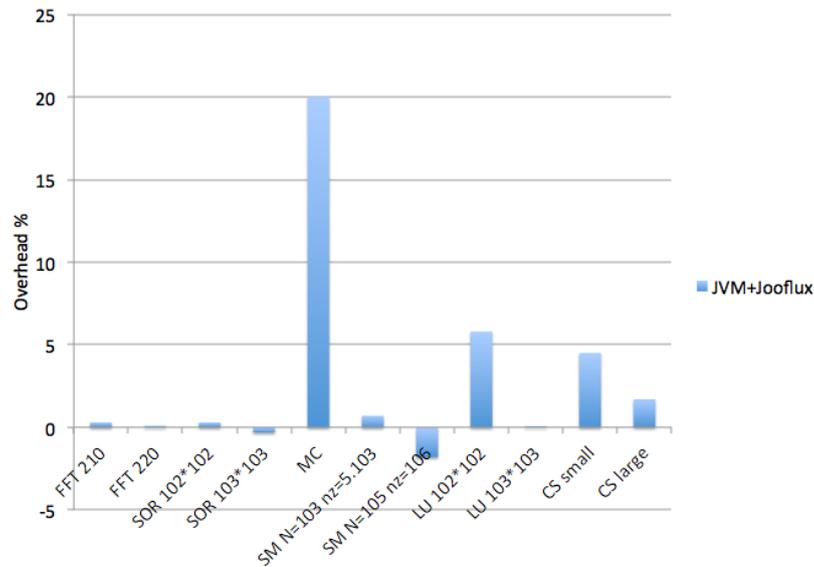


FIGURE 8 – Macro-tests de Jooflux avec SCImark2.0

Appliqué aux autres macro-jeux de tests, Jooflux présente de manière similaire un surcoût faible, voire même de manière surprenante, le bytecode modifié par JooFlux améliore quelquefois les performances. Ce résultat montre bien que les appels synchronisés de méthodes ne sont pas affectés par l'indirection `invokedynamic` et que le JIT peut même en bénéficier et parfaitement trouver de meilleures inclusions de code.

3.2 CONGOLO

OBJECTIFS Facilement programmer le comportement d'une application est déjà un défi en soi, mais lorsque cette application doit s'adapter à un environnement changeant ou hétérogène, cela revêt une encore plus grande importance. Par exemple dans les environnements ambiants, les applications font souvent appel à de la capture de données externes, externalisent l'agrégation de données et/ou l'exécution de code. La plupart des langages de programmation orientés-contexte (*Context-Oriented Programming - COP*) intègrent ces comportements dans la même classe ou des couches proches (*layers*). Et ces comportements sont également le plus souvent définis par des règles ECA (*Event-Condition-Action*) embarquées dans le code métier ou dans la définition du contexte. Cette définition de comportement est plutôt res-

trictive et pourrait bénéficier de mécanismes de prise de décision plus évolués et complexes comme les réseaux de neurones (*Neural Networks*) ou l'apprentissage automatique (*Machine Learning*).

Nous proposons un nouveau langage de programmation orienté-contexte avec une nouvelle approche permettant (1) d'exprimer dans le langage - de façon simple - des méthodes contextuelles, sans avoir besoin de déclarer des couches, et (2) d'abstraire et séparer le contexte et la prise de décision, avec deux APIs distinctes, et (3) d'externaliser ces fonctionnalités sous forme de service - très souvent fait pour le contexte, mais jamais à notre connaissance pour la prise de décision - en les interconnectant de manière asynchrone par envoi de messages. Nous démontrons la faisabilité de l'approche avec ConGolo - *Contextual Golo*, une extension de Golo, langage dynamique basé sur la JVM [Ponge et al. 2013].

CONTRIBUTION ET RÉSULTATS Une invocation de méthode dans ConGolo se déroule comme présenté dans la Figure 9.

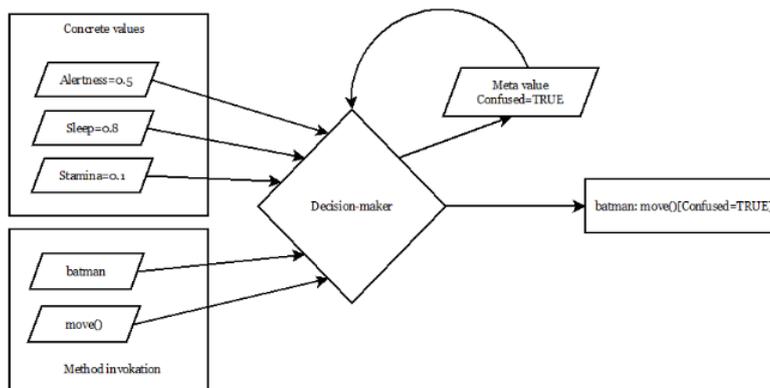


FIGURE 9 – Invocation contextuelle de méthode avec ConGolo

Lors de l'invocation de la méthode `batman: move()`, cette invocation est traitée par le service de prise de décision qui va décider - si l'objet `batman` présente plusieurs méthodes `move()` - laquelle invoquer selon le contexte actuel. L'objet `batman` a enregistré une sensibilité au contexte `Confused`, alors le service de prise de décision va vérifier si celui-ci est positionné à `Confused=TRUE`. L'expression du contexte en ConGolo se fait avec deux niveaux : un niveau *Meta* avec des variables activées ou non, et un niveau *Concrete* avec des variables positionnées à différentes valeurs dans le service de gestionnaire de contexte. Ce découpage en deux niveaux permet au développeur d'application de raisonner sur un contexte général - indépendant du contexte réel. La correspondance avec les vraies situations - i.e. le contexte mesuré - est effectuée par le service de prise de décision - qui peut très bien avoir été mise en œuvre par une autre personne, voire même être totalement décollé de l'application effectuant l'invocation de méthode. Dans notre exemple, le service de prise de décision va calculer que `batman` est moyennement vif ($Alertness=0.5$), plutôt fortement endormi ($Sleep=0.8$) et pas très physiquement en forme ($Stamina=0.1$), et donc vérifier que `Confused` est `TRUE` et invoquer la méthode de déplacement de `batman` en mode confus `batman: move()[Confused=TRUE]`.

Syntaxe du langage La simplicité du langage ConGolo est le facteur le plus important.

```
1 contexts = [ConfusedHero(), BadWeather()]
```

LISTING 2 – Déclaration du contexte en ConGolo

La déclaration du contexte s'effectue avec le mot réservé du langage `contexts`, comme montré dans le Listing 2. Cette déclaration doit inclure un n-uplet d'objets respectant l'interface `Context` - permettant de savoir si cet objet du contexte est `TRUE` ou `FALSE`. La portée de ce contexte est le module dans lequel il est déclaré.

```
1 function Hero = || {
2   return DynamicObject():
3   contexts([ConfusedHero]):
4   # base methods
5   define("move", |this, dir| -> ...):
6   define("getPos"; |this| -> ...):
7   # layered method, invoked before base method
8   define("getPos", |this|@(ConfusedHero=TRUE)+ -> ...):
9   # layered method, invoked after base method
10  define("move", |this, dir|+@(ConfusedHero=TRUE) -> ...):
11  # layered method, with a call to the base method from within
    the layer
12  define("move", |this, dir|@(ConfusedHero=TRUE)+) = {
13    # ...
14    proceed(dir)
15    # ...
16  }
17 }
```

LISTING 3 – Déclaration d'un objet contextuel en ConGolo

Lors de la déclaration d'un objet - comme le `Hero` dans le Listing 3 - il est possible de spécifier si cet objet est sensible au contexte et à quel élément du contexte, comme présenté en ligne 3 avec le même mot réservé `contexts([ConfusedHero])`. Pour spécifier une méthode contextuelle, aucun mot réservé spécifiant une couche n'est nécessaire. Après la déclaration d'une méthode classique `define("getPos", |this|)`, il suffit de déclarer une nouvelle méthode en ajoutant le contexte dans la signature de la méthode `define("getPos", |this|@(ConfusedHero=TRUE)+ -> ...)`. Avec `@(Context)+`, la méthode contextuelle est exécutée à la place de la méthode de base - si le contexte est vérifié. Avec `+@(Context)`, la méthode de base est exécutée d'abord puis la méthode contextuelle. Et pour exécuter une méthode contextuelle puis la méthode de base, la méthode contextuelle - comme présenté en ligne 14 - peut invoquer la méthode de base avec le mot réservé `proceed`.

Ensuite, la prise de décision peut s'effectuer par un service de décision global - comme dans le listing 4, ou par un service de décision local propre à l'objet - comme dans le listing 5. Dans ce second exemple, la sensibilité au contexte et le service de décision sont modifiés dynamiquement. Le service de prise de décision reçoit des événements d'invocations de méthodes, évalue le contexte auquel celles-ci sont sensibles et ré-envoie un ou plusieurs événements résultat permettant

```

1 let misterPresident = my.app.myDecisionMaker()
2 misterPresident: global(true)
3
4 let batman = Hero() # Context-dependent object
5 batman: move() # Call to a layered method that will be handled by
   the global decision-maker

```

LISTING 4 – Déclaration du service de décision global en ConGolo

```

1 function Hero = |newContext, decisionMaker| {
2   return DynamicObject():
3   contexts(newContext): # Context change
4   decisionmaker(decisionMaker) # Local decision add
5 }
6
7 let misterVicePresident = my.app.myOtherDecisionMaker()
8 let batman = Hero([ConfusedHero, BadWeather], misterVicePresident)

```

LISTING 5 – Modifications dynamiques de la sensibilité au contexte et du service de décision local en ConGolo

l’invocation de la bonne méthode. Ce service peut facilement s’implanter avec des règles ECA (*Event - invocation / Condition - context / Action - method dispatch*), mais peut avoir n’importe quel système de décision - comme par exemple un réseau de neurones avec les nœuds d’entrée prenant les invocations, les arcs du treillis pondérés par le contexte, et les nœuds de sortie invoquant la bonne méthode contextuelle.

Comparaison ConGolo / Golo Pour évaluer un langage orienté-contexte, un cas classique de jeu de tests est de comparer les appels de méthodes en couche avec leurs homologues standards et de mesurer le surcoût [Appeltauer et al. 2009]. Nous avons testé l’imbrication de 10 méthodes. Les résultats sont présentés dans le Tableau 1. Nous avons constaté une perte de performance significative : les invocations de méthodes de ConGolo sont deux ordres de grandeur plus lentes que celles de Golo, et les invocations imbriquées sont encore un ordre de grandeur plus lente que les invocations simples.

Test	Langage	Score (nops/ms)
Invocation simple	ConGolo	55
	Golo	5939
Invocation en couche (10)	ConGolo	2.7
	Golo	3840

TABLE 1 – Résultats des jeux de tests de ConGolo avec des appels directs

Cette différence de performance peut facilement s’expliquer. Premièrement, l’instruction `invokedynamic` - bytecode utilisé par Golo - peut être liée à différents types de méthodes cibles : `static`, `mutable` et `volatile`. Dans le cas `static`, la méthode de démarrage (`bootstrap`) ne peut pas changer pour une nouvelle cible, peut être appelée directement et ses enchaînements d’appels peuvent donc être optimisés. Dans le cas

mutable¹, chaque appel de méthode est effectué avec un test de garde préalable et peut être dynamiquement changé. Dans notre cas, ce test nous permet de vérifier si le contexte a changé ou non. Pour chaque appel de méthode, notre système doit donc prendre une décision, en invoquant le service de prise de décision.

Le deuxième point impactant les performances est la méthode de communication utilisée entre la méthode de démarrage et le service de prise de décision. Nous avons opté pour une communication par messages - implanté dans un premier temps avec un répartiteur d'évènements mono-thread. Nous avons comparé les performances entre l'invocation directe de méthodes et l'invocation par événements. Les résultats sont dans le Tableau 2 et montrent que le temps d'invocation d'une méthode est divisé par deux. Ce résultat est encourageant car il montre qu'en optimisant - avec du multi-thread par exemple - l'impact de la communication par message pourrait rester raisonnable comme cela l'est pour Golo.

Test	Langage	Score (nops/ms)
Invocation simple	ConGolo	29
	Golo	6135
Invocation en couche (10)	ConGolo	1.9
	Golo	3998

TABLE 2 – Résultats des jeux de tests de ConGolo avec des envois d'évènements

Même si ConGolo est moins performant de deux ordres de grandeur par rapport à Golo, pouvoir effectuer *par milliseconde* 30 appels de méthodes contextuelles faiblement imbriquées - ou 2 à 3 appels de méthodes contextuelles fortement imbriquées, semble suffisant pour des environnements ambiants/IoT où le changement de contexte est habituellement lié à des changements physiques assez stables [González et al. 2011].

1. Mutable est le cas d'appel dynamique de méthode, car volatile n'est jamais utilisé avec Golo ni avec ConGolo.

CHARGEMENT / DÉCHARGEMENT CONTEXTUELS, COMPOSITION ET INTÉGRATION SPONTANÉES DE SERVICES

*The cheapest, fastest, and most reliable components of a computer system
are those that aren't there.*

— Gordon Bell

Le second volet des travaux sur lesquels j'ai travaillé s'inscrit dans la poursuite de mes travaux de doctorat [Le Mouël 2003]. Il s'attache à la combinaison de la modularité des applications et du dynamisme de l'environnement - en particulier l'application des Architectures Orientées-Services aux périphériques contraints et mobiles :

- AxSeL - travaux de thèse d'AMIRA BEN HAMIDA [Ben Hamida 2010] - présente, en Section 4.1, une approche à 'gros grain' faisant le lien entre les services découverts dans l'environnement et les composants de déploiement implantants ces services. Le choix des composants à charger / décharger s'effectue avec une heuristique gloutonne de coloriage de graphe [Ben Hamida et Le Mouël 2006, Ben Hamida et al. 2007, Ben Hamida et al. 2008b, Ben Hamida et al. 2008a, Ben Hamida et al. 2011].
- ACOMMA - travaux de thèse de ROYA GOLCHAY [Golchay 2016] - présente, en Section 4.2, une approche à grain fin permettant d'effectuer le déchargement d'exécution de méthodes dans un nuage de proximité spontanée et collaboratif. Le choix des méthodes à décharger s'effectue selon un algorithme multi-critères bio-inspiré par le comportement des fourmis [Golchay et al. 2011, Golchay et al. 2011, Golchay et al. 2013, Golchay et al. 2016a, Golchay et al. 2016b].
- MySIM - travaux de thèse de Noha Ibrahim [Ibrahim 2008] - présente, en Section 4.3, une approche permettant de composer spontanément des services découverts dans l'environnement. La sélection ou le remplacement de service s'effectue sémantiquement en fonction de leur qualité de service [Ibrahim et al. 2005, Ibrahim et al. 2006, Ibrahim et Le Mouël 2006, Ibrahim et al. 2006, Ibrahim et al. 2007a, Ibrahim et al. 2007b, Ibrahim et Le Mouël 2007, Ibrahim et Le Mouël 2009, Ibrahim et al. 2009, Ibrahim et al. 2010, Ibrahim et al. 2014].

4.1 AXSEL

OBJECTIFS L'expansion de l'utilisation des terminaux mobiles offre une multitude de fonctionnalités aux utilisateurs. En effet, ces plateformes rendent accessibles les services, applications et fonctionnalités n'importe où et n'importe quand. Par ailleurs, la conscience du contexte est un requis majeur dans les environnements intelligents principalement caractérisés par un constant changement dû à la mobilité

des utilisateurs et à l'apparition de nouveaux périphériques électroniques proposant de nouveaux services. Par conséquent, l'adaptation dynamique à ces changements est essentielle mais doit s'effectuer en respectant les contraintes matérielles mais aussi les préférences de l'utilisateur.

Bien que la conscience du contexte soit fondamentale, les applications usuelles ne sont pas conçues pour s'adapter à un contexte donné. Elles nécessitent souvent l'installation de la totalité de leurs composants sans appliquer une quelconque sélection. En partant de cette problématique, nous proposons d'intégrer la notion de contexte dans le déploiement des applications afin de les adapter lors du déploiement initial et de l'exécution, aux diverses contraintes de l'environnement omniprésent. Pour cela, nous proposons AxSeL (A conteXtual Service Loader) - une architecture de déploiement d'applications orientées services. AxSeL considère les applications orientées services composées par des éléments nécessaires (prioritaires) et d'autres optionnels. Nous proposons une adaptation structurelle de l'application au moment du déploiement et lors de l'exécution afin de rompre l'aspect figé des applications usuelles, mais aussi afin de donner la possibilité de toujours fournir la fonctionnalité dans des conditions contextuelles différentes. Notre approche propose trois contributions principales : (1) la représentation d'une application sous la forme d'un graphe de dépendances bidimensionnel de services et de composants, (2) un déploiement contextuel basé sur une coloration du graphe de dépendances entre services et composants avec prise en compte de plusieurs critères aboutissant à des configurations différentes d'une application donnée, et (3) une adaptation contextuelle au cours de l'exécution de l'application à travers une remise en cause de la coloration initiale suite à des notifications d'événements contextuels.

CONTRIBUTION ET RÉSULTATS Dans un environnement omniprésent, les services sont hébergés sur des dépôts distants ou locaux décrits par des descripteurs. Les descripteurs de services incluent des données sur les services (dépendances, emplacements, taille mémoire). Les terminaux mobiles accèdent à ces dépôts à travers leur descripteur de services, et y puisent les fonctionnalités requises. Une fois trouvé, le composant implantant le service est chargé localement. AxSeL effectue la prise en compte du contexte par une vue unifiée des différents dépôts de services au travers d'un unique descripteur. Le descripteur offre aux dispositifs une vision contextuelle des services disponibles dans l'environnement. La Figure 10 présente une vue générale de l'architecture.

Pour effectuer le chargement de services, AxSeL suit quatre étapes :

1. l'extraction des dépendances est le processus par lequel les dépendances d'un service sont extraites à partir d'un descripteur de services. Ce processus aboutit à un graphe de dépendances incluant les propriétés du service et celles de ses dépendances. Les nœuds du graphe sont les services et composants, et les arcs sont les dépendances extraites. Les services et composants sont décrits dans le dépôt d'une manière unitaire. Nous réalisons donc l'extraction récursivement en parcourant les dépendances de chaque nœud et en les incluant au fur et à mesure dans le graphe de dépendances globales. Lorsque l'ensemble des dépendances est tracé, la construction du graphe est alors finie (descripteur commun de services, Figure 10) et se présente sous forme d'un

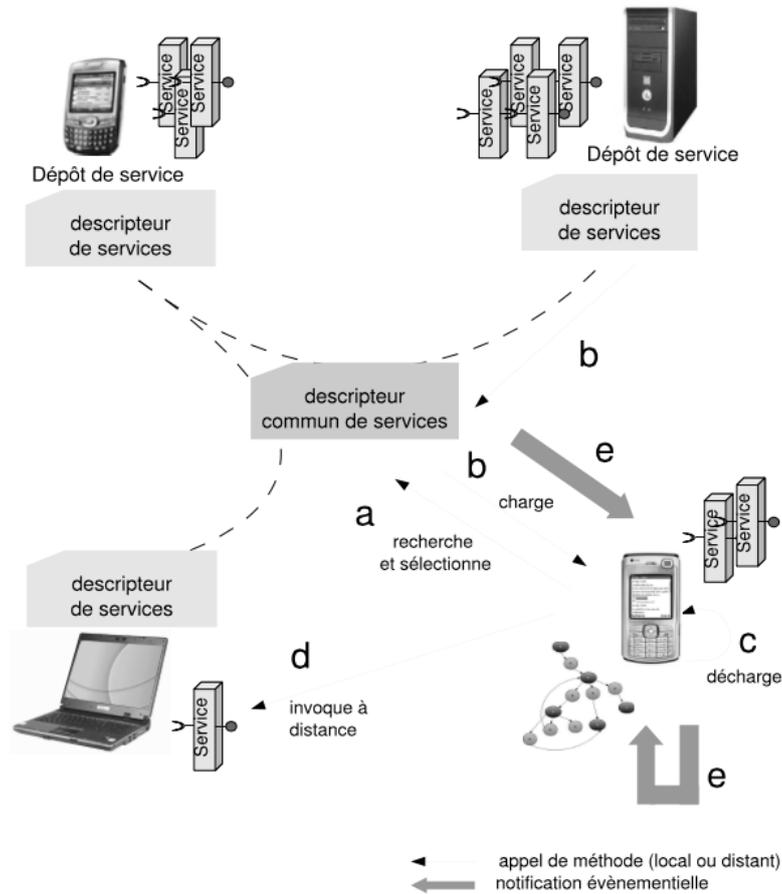


FIGURE 10 – Architecture d’AxSEL - Construction d’un descripteur commun

graphe bidimensionnel de services (exécution) / composants (déploiement) (Figure 11),

2. la décision du chargement est l’opération de parcours du graphe de dépendances du service en confrontant le contexte requis par celui-ci avec le contexte fourni par les terminaux mobiles. Pour chaque service du graphe de dépendances, une décision de chargement ou non, selon les contraintes, est prise (étape a, Figure 10),
3. le chargement/déchargement est l’étape qui applique la décision précédente en effectuant techniquement le téléchargement des composants, réalisant son installation, le démarrant et publiant le service sur la plate-forme de services (étape b, Figure 10). Le déchargement est également possible, par exemple après modification du contexte, et correspond à une désinstallation du composant de la plate-forme (étape c, Figure 10). Un déchargement peut s’accompagner, si le service est indispensable ou en cours d’utilisation, d’une redirection des appels vers un service distant (étape d, Figure 10),
4. l’adaptation contextuelle est réalisée lorsque des changements sont observés dans le contexte (étape e, Figure 10). Ces notifications peuvent provenir du dispositif lui-même à partir des moniteurs sur le matériel (diminution ou augmentation de la mémoire, activation ou désactivation d’interfaces réseau, etc.) et

également à partir des descripteurs de dépôt avec des notifications de nouvelles versions, des ajouts ou des suppressions d'un ou plusieurs services ou composants. Cette étape déclenche une nouvelle prise de décision avec comme conséquences d'éventuels chargements/ déchargements (étapes a, b, c, Figure 10).

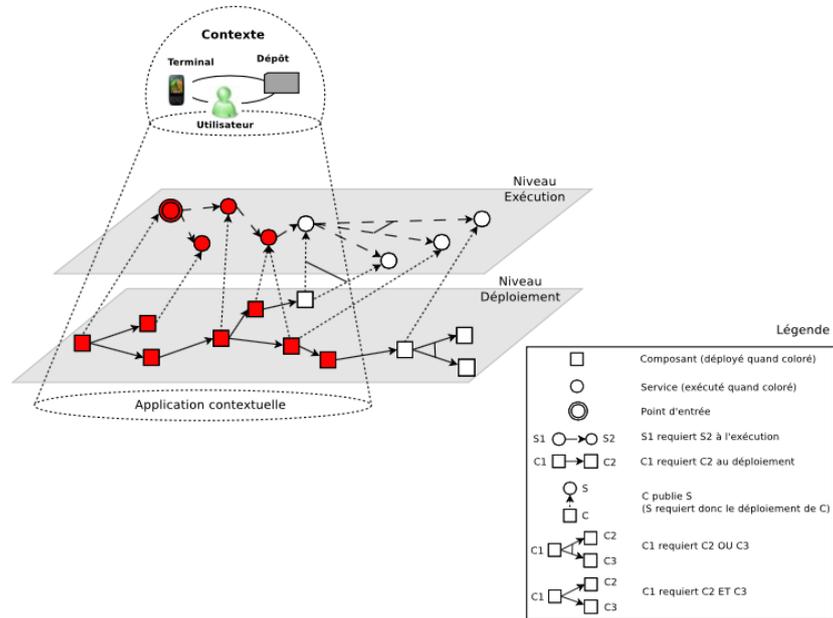


FIGURE 11 – Projection du contexte sur le graphe de services / composants AxSeL

Projection du contexte sur le graphe bidimensionnel Comme le montre la Figure 11, un point-clef pour la prise en compte du contexte est que celui-ci est projeté sur le graphe afin d'obtenir une vue contextuelle du déploiement des composants et services. Les aspects contextuels liés aux composants et services, comme leurs priorités définies par l'utilisateur ou leurs occupations mémoire surveillées sur le terminal, sont projetés sur le graphe ; d'autres éléments contextuels, comme par exemple l'occupation mémoire globale ou la batterie du terminal, restent externalisés et surveillés par le terminal. Cette surveillance du contexte se base sur des mécanismes de capture des événements provenant des sources définies, ensuite sur le traitement des informations collectées et enfin par un mécanisme de notification. Afin d'éviter qu'AxSeL ne gère des données hétérogènes recueillies à partir de trop de sources d'information non pertinentes, nous utilisons un mécanisme qui masque les sondes. En effet, pour chaque source, nous associons un adaptateur (*Wrapper*) permettant à travers l'interaction avec des APIs spécifiques de récupérer les données à partir des sondes. Pour les ressources matérielles, les APIs utilisées sont celles fournies par la plate-forme d'exécution, dans notre cas nous reposons sur une machine virtuelle permettant la récupération des valeurs de la mémoire. Les données relatives aux services et aux composants sont extraites à partir de l'API correspondante du dépôt. Au moment de la projection de contexte, chaque nœud du graphe - composant ou service - enregistre alors un mécanisme d'écoute sur les adaptateurs pour être notifié en cas de changement. Il est important de noter que chaque nœud enregistre

termes de consommation mémoire et de temps d'exécution à petite échelle. Nous avons voulu observer ces coûts dans le cas de plus grands graphes pour évaluer le passage à l'échelle d'AxSeL. Nous réalisons des simulations sur des graphes que nous générons automatiquement. Les graphes générés ont pour chaque nœud entre une et cinq dépendances au niveau composant et trois dépendances au niveau service. Nous observons ensuite les performances relatives à l'exécution d'AxSeL avec et sans adaptation.

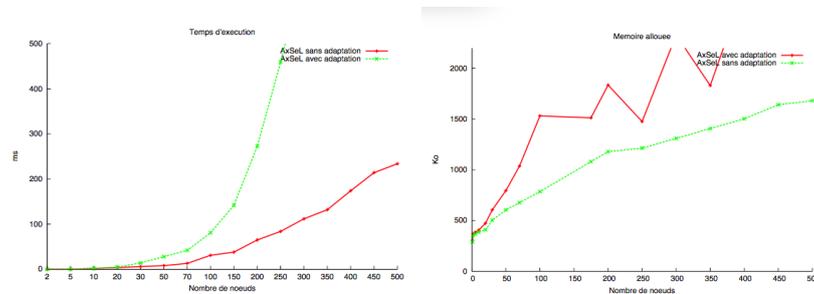


FIGURE 13 – Performances d'AxSeL en fonction du nombre de nœuds du graphe

La courbe 13 (de gauche) illustre le coût en temps d'exécution observé avec et sans adaptation. Sans surprise, AxSeL est plus lent avec adaptation que sans. Le temps total d'exécution pour un graphe de 250 nœuds passe de 100 ms à 500 ms. Ce temps reste très raisonnable et ne pénalise pas l'utilisateur avec un temps d'attente très long. De plus, le processus d'adaptation est un processus ponctuel qui n'est pas réalisé constamment. Cette courbe présente toutefois un profil exponentiel, et il conviendrait d'étudier d'autres algorithmes pour une complexité moindre. La courbe 13 (de droite) représente le coût en termes de mémoire en fonction du nombre de nœuds. La courbe d'usage mémoire sans adaptation a un comportement linéaire allant de 400 Ko de RAM et n'excédant pas 1500 Ko dans le pire des cas (graphe à 500 nœuds). Quant à la courbe d'adaptation, nous notons un surcoût raisonnable de 500 à 1000 ko, mais des oscillations et un écart type important de l'ordre de 40 %. Ces variations importantes sont difficiles à anticiper car le processus de recherche des dépendances pour chaque nœud étant plus important, le ramasse-miettes intervient alors inopinément pour nettoyer et désallouer de la mémoire. Les simulations ont toutefois été réalisées en considérant le pire des cas qui est la recoloration totale du graphe. Dans des cas plus communs nous ne procédons pas à la recoloration du graphe en entier mais à une partie uniquement. Si nous nous plaçons dans un cas contraint, par exemple avec notre smartphone Nokia 6630, notre plate-forme AxSeL, avec des applications usuelles avec une centaine de nœuds en moyenne, ne dépassera pas la valeur de 2000 Ko avec ou sans adaptation, soit 2,7 % de la mémoire du Nokia. Ce coût est raisonnable. Après un usage intensif par chargement des applications de calendrier, de la liste des contacts, de l'appareil photo et caméra et d'un navigateur web, nous avons bien observé un taux important d'occupation de la mémoire de 97 %. Dans le cas de non-utilisation d'AxSeL, l'utilisateur ne peut plus exécuter d'applications et doit en arrêter manuellement une application pour en exécuter une nouvelle. Dans le cas d'utilisation d'AxSeL, la désinstallation de l'application la plus gourmande s'effectue automatiquement, sans intervention de l'utilisateur, en exécutant la nouvelle application.

4.2 ACOMMA

OBJECTIFS L'explosion de l'usage des smartphones fait de ceux-ci l'outil de communication le plus efficace et prisé de ces dernières années - avec une augmentation de 50% de leur nombre en 2013 par exemple. Bien que les smartphones présentent des capacités croissantes en terme de fonctionnalités - GPS, Wifi, caméra, et en terme de puissance de calcul et mémoire, les applications deviennent également de plus en plus complexes et consommatrice de ressources et donc de batterie - jeux vidéo, navigation, réalité augmentée, etc.

Une solution pour obtenir de bonnes performances en préservant la batterie est de profiter d'environnements puissants en exécutant à distance tout ou parties des applications (*offloading*). L'utilisation d'environnements d'informatique dans les nuages à cet effet (*Mobile Cloud Computing*) a permis d'enrichir l'offre applicative des smartphones mais présente le nouveau défaut d'une latence réseau importante avec ce nuage lointain. Pour adresser ce problème, plusieurs approches cherchent à rapprocher ces environnements puissants au plus proche en terme de réseau et également avec une proximité géographique. Ces environnements proches présentent par contre souvent des contraintes de dynamisme, par exemple avec de la mobilité de périphériques. Les partitionnements planifiés de manière optimale mais statiquement ne peuvent généralement pas répondre à ces contraintes.

Nous proposons ACOMMA - *an Ant-inspired Collaborative Application Offloading Middleware for Mobile Applications* - un intergiciel d'exécution automatique d'applications dans un nuage de proximité créé spontanément avec les autres smartphones à proximité. ACOMMA effectue une exécution distante à grain-fin au niveau de chaque méthode d'une application en décidant du partitionnement avec un algorithme inspiré des colonies de fourmis. Cet algorithme bio-inspiré permet à ACOMMA de s'adapter à différentes situations de l'environnement.

Les principales contributions d'ACOMMA sont : (1) la transformation automatique des méthodes des applications en services, (2) un algorithme de partitionnement avec prise de décision bi-objectifs et multi-destinations pour l'exécution distante, et (3) un apprentissage collaboratif à base de caches collaboratifs pour ne pas avoir besoin de reprendre une décision de partitionnement si une situation similaire s'est déjà présentée.

CONTRIBUTION ET RÉSULTATS L'architecture d'ACOMMA est présentée dans la Figure 14. L'exécution distante peut s'effectuer dans un nuage distant, tout comme dans un nuage proche créé spontanément avec les périphériques du voisinage (*Spontaneous Proximity Cloud*).

ACOMMA est composé de plusieurs services permettant de découvrir (*Discovery Service*) et de maintenir la connaissance de l'environnement (*Context Monitoring Service*). Les deux services les plus importants sont le service responsable de l'exécution distante (*Offloading Manager Service*) - décidant des appels devant être redirigés, et le service de gestion des collaborations (*Collaboration Service*) - conservant ces décisions.

Transformation d'applications Pour pouvoir effectuer un partitionnement à grain-fin, pour chaque application avant son exécution, ACOMMA effectue une transformation de chaque appel de méthode

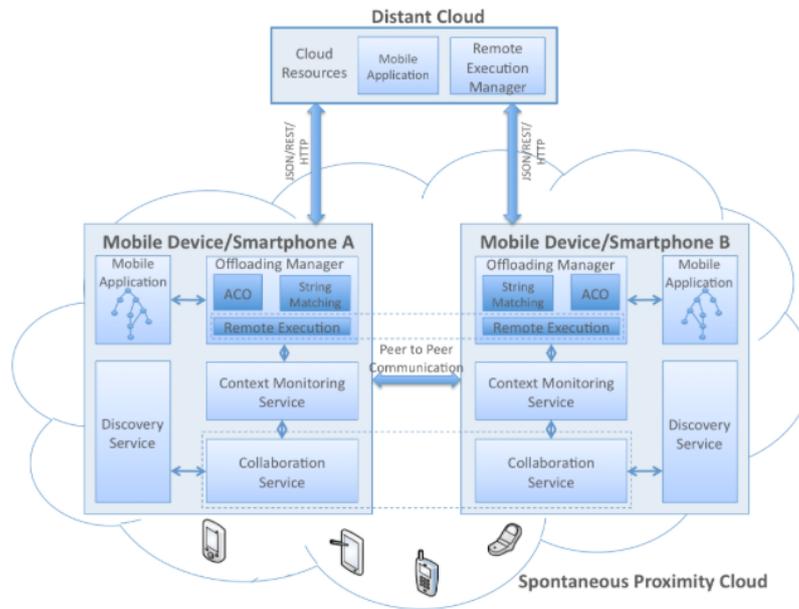


FIGURE 14 – Architecture générale d'ACOMMA

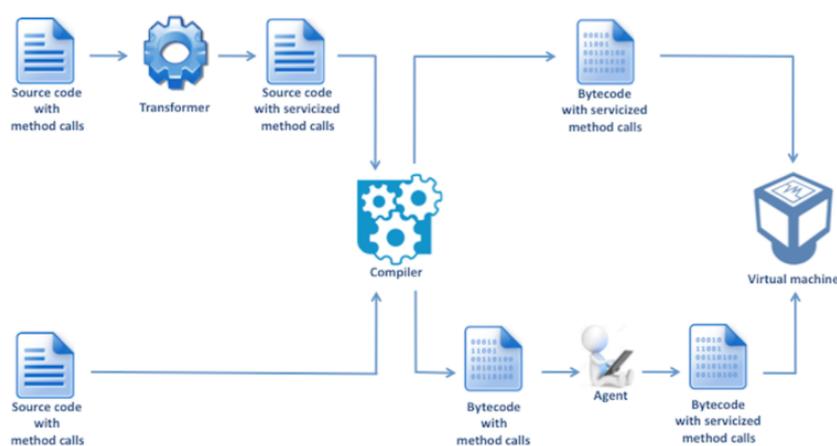


FIGURE 15 – Transformation automatique des appels de méthodes en services dans ACOMMA

en une invocation de service - avec une API REST permettant au flux de données de transiter à travers tout pare-feu. Comme le montre la Figure 15, dans le cas d'une application dont les sources sont connues, le transformateur génère un nouveau code source pouvant être compilé. Dans le cas d'applications ne souhaitant pas modifier leur code source ou ne fournissant pas leur code source, nous avons utilisé un agent branché à la machine virtuelle interceptant dynamiquement les appels et effectuant la réécriture du bytecode sous forme d'invocation de service. Dans notre cas, cette transformation à la volée est effectuée avec Jooflux connecté à une VM Dalvik (cf Section 3.1).

Prise de décision dynamique bi-objectifs et multi-destinations L'objectif des approches effectuant une exécution distribuée est de prendre le graphe d'appels de méthodes d'une application et de le partitionner - i.e. de la couper - si possible de manière optimale pour décider de

quels appels seront effectués localement et à distance. Pour pouvoir s'adapter à des conditions très changeantes sur le graphe d'appels - i.e. ajout, retrait de nœuds, valeur de poids évolutives - nous avons considéré des graphes non pas à partitionner en deux mais en plusieurs destinations, avec des intérêts entre les destinations pouvant être différents et pouvant donc être multi-critères. Nous sommes intéressés aux algorithmes bio-inspirés, particulièrement aux algorithmes de colonies de fourmis (*Ant Colony Optimization*) car ceux-ci présentent une heuristique efficace sur le multi-critère. Les fourmis parcourant le graphe utilisent un système de phéromones à effet "mémoire" ayant l'avantage de préserver le meilleur chemin trouvé dans le graphe si celui-ci est fortement emprunté. Ces phéromones pouvant s'évaporer, les fourmis possèdent la faculté d'emprunter un nouveau chemin lorsque de nouvelles conditions se présentent.

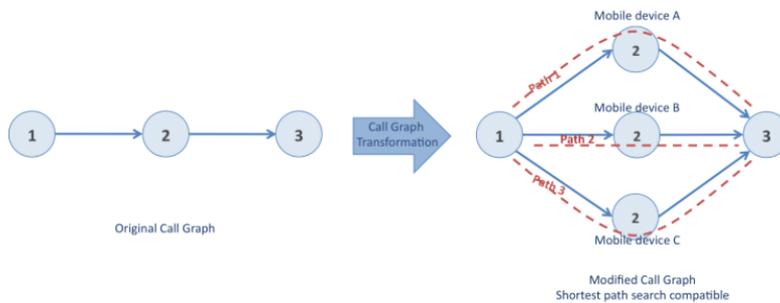


FIGURE 16 – Problème du plus court chemin appliqué dans ACOMMA pour la décision d'exécution distante

L'originalité de notre approche consiste à transformer le problème de partitionnement du graphe d'appels en un calcul de plus-court chemin avec l'algorithme de colonie de fourmis. La Figure 16 illustre cela avec un graphe à 3 nœuds. La question est de savoir si le nœud 2 doit s'exécuter localement ou sur une autre destination. Nous enrichissons le graphe en dupliquant chaque nœud autant de fois qu'il y a de destinations possibles - ici 2A pour une exécution sur le mobile A, 2B pour une exécution sur le mobile B, 2C pour une exécution sur le mobile C. Nous ajoutons des étiquettes n-uplet de coût (CPU, temps d'exécution) sur les arcs et calculons le plus court chemin entre les nœuds 1 et 3. Les fourmis partent au hasard sur le graphe, puis conservent les chemins les plus empruntés correspondant à l'ensemble des solutions non-dominées.

Taux de succès et gains Pour évaluer le gain à effectuer le choix de l'exécution à distance, nous avons testé des micro- et macro-jeux de tests sur des smartphones Android Samsung Galaxy et Google Nexus avec une machine Macbook Pro comme nuage distant. Les micro-jeux de tests sont des fonctions de calcul mathématique (Fibonacci, multiplication de matrices, calcul de déterminant, calcul d'intégrale) et les macro-jeux de tests sont les applications classiques 'offloading' (reconnaissance faciale - avec l'algorithme Eigenface, et jeu d'échec - avec l'algorithme Monte-Carlo).

Comme le montre le Tableau 3, l'algorithme de colonie de fourmis est bien une heuristique et peut se tromper. Le taux moyen de succès pour Fibonacci et multiplication de matrices est respectivement de 62%

		Fibonacci			Multiplication			Determinant			Integrate		
		Success (%)	Time (%)	CPU (%)	Success (%)	Time (%)	CPU (%)	Success (%)	Time (%)	CPU (%)	Success (%)	Time (%)	CPU (%)
Galaxy S2	Serie 1	60	40.54	20	48	7.6	-16.66	72	31.43	66.67	96	93.97	86.55
	Serie 2	64	29.7	18.75	60	11.93	0	100	66.20	93.73	96	96.99	99.49
	Serie 3	52	27.25	15.38	52	13.06	0	100	92.82	97.36	96	97.78	99.55
	Serie 4	56	31.08	7.14	52	8.37	15.38	100	98.23	99.54	96	98.59	99.81
Nexus 7 Tablette	Serie 1	76	13.34	10.53	48	9.74	-8.33	88	32.88	30.30	96	96.84	98.44
	Serie 2	52	18.98	23.08	44	7.03	0	96	83.78	82.97	96	98.39	98.65
	Serie 3	80	24.38	53.67	56	4.30	-7.14	96	96.23	98.53	96	98.85	99.18
	Serie 4	56	13.23	13.69	48	3.72	8.33	100	98.86	99.30	96	99.15	99.55

TABLE 3 – Performances d’ACOMMA sur des micro-benchmarks

et 59,5%. Ce taux monte à 94% et 96% pour les calculs de déterminants et d’intégrales, et même - dans le Tableau 4 - à 98% et 97% pour la reconnaissance faciale et l’algorithme de Monte-Carlo. Ce résultat nous montre que plus la complexité du graphe est grande, plus l’algorithme de colonie de fourmis a des chances de succès.

Fibonacci et multiplication de matrices n’ayant pas vraiment de forts calculs à effectuer, les gains en CPU et temps d’exécution ne sont pas très élevés et fluctuants. Pour le calcul de déterminant, les gains peuvent être très bons 98% et 99% lorsque le calcul demandé est important (série 4). Pour le calcul d’intégrales, les gains sont de suite intéressants.

		Face Recognition			MonteCarlo		
		Success (%)	Time (%)	CPU (%)	Success (%)	Time (%)	CPU (%)
Galaxy S2	Serie 1	100	83.92	81.43	96	81.66	99.74
	Serie 2	100	87.89	60.47	100	83.72	97.74
	Serie 3	100	88.52	75.23	96	94.68	99.92
	Serie 4	100	89.98	77.89	96	96.05	99.88
Nexus 7 Tablette	Serie 1	96	83.74	89.68	96	95.28	96.73
	Serie 2	92	82.09	91.71	100	94.39	97.36
	Serie 3	100	80.77	84.82	96	98.64	99.15
	Serie 4	96	78.40	75.42	96	99.01	99.34

TABLE 4 – Performances d’ACOMMA sur des macro-benchmarks

Pour les macro-jeux de tests, les gains vont être plus réalistes entre 78% et 89%.

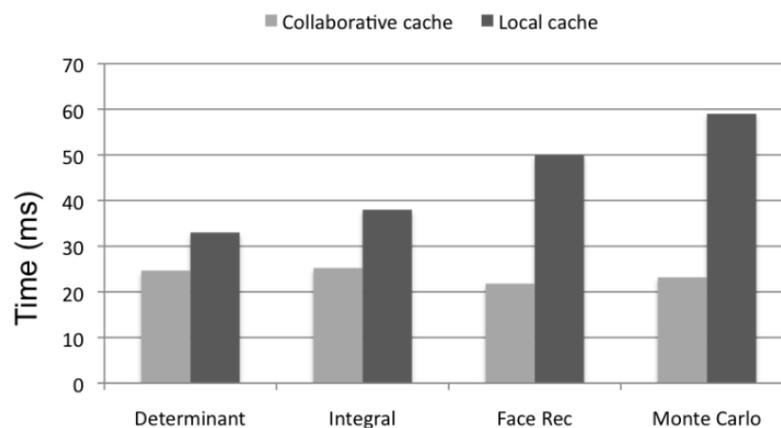


TABLE 5 – Performances d’ACOMMA avec cache collaboratif

Un des points les plus intéressant était de voir si une collaboration entre les smartphones de proximité permettait une convergence plus rapide des décisions avec un cache collaboratif. Pour cela, nous avons

augmenté le nombre de smartphones à 2 Galaxy Samsung SII et 3 Google Nexus et comparé quand ceux-ci prennent une décision uniquement sur la base du cache local ou quand ils partagent les décisions. La Figure 5 présente les résultats. Les applications les moins complexes - calcul de déterminants et d'intégrales - présentent 30% à 40% de gain (10ms) et les plus complexes - reconnaissance faciale et Monte-Carlo - jusqu'à 60% (35ms).

4.3 MYSIM

OBJECTIFS Construire une application ambiante peut s'avérer une tâche difficile. En effet, le développeur d'application peut avoir à connaître des protocoles de bas niveau - correspondants à des interfaces matérielles - tout comme des protocoles de niveau applicatif. C'est cette logique qui a amené le développement des Architectures Orientées-Services, avec la possibilité d'un système distribué faiblement couplé où chaque développeur peut se concentrer uniquement sur le développement métier de son service.

Une infrastructure où "tout est service" présente plusieurs possibilités et défis pour développer des applications. Une des manières naturelles de le faire est d'effectuer de l'intégration de service [Ishikawa et al. 2005], en en créant certains, combinant ou adaptant d'autres pour effectuer une tâche précise. Ces capacités d'adaptation deviennent alors cruciales pour les intergiciels pour faciliter cette création d'application.

De nombreux intergiciels offrent ce type de mécanismes mais ils sont pour l'essentiel centrés sur le développeur ou l'utilisateur et ne font que répondre aux requêtes et besoins de ceux-ci. Nous pensons qu'un intergiciel peut se baser sur les buts à atteindre (*Task-Oriented*) et effectuer automatiquement plusieurs de ces tâches d'adaptation.

Pour cela, nous proposons MySIM - *a Spontaneous Integration Middleware*, pour les environnements ambiants. Les trois principales contributions sont : (1) de définir l'intégration comme une combinaison de trois aspects fonctionnels : la transformation de services, la composition de services et l'adaptation de services, (2) de proposer un intergiciel - transparent pour l'utilisateur - basé sur une réaction aux événements d'apparition et disparition de services, et (3) de définir une métrique considérant les aspects fonctionnels des services (description syntaxique et sémantique) et non-fonctionnels (qualité de service) pour comparer, évaluer et trouver la meilleure combinaison de services.

CONTRIBUTION ET RÉSULTATS Dans les Architectures Orientées-Services, le développement d'application repose sur l'utilisation de services - briques logicielles de base - qui vont être décrites par leurs interfaces. L'application consiste donc 'juste' à enchaîner des appels sur ces interfaces et construire le graphe d'appels. Les services possèdent la propriété d'être faiblement couplés - i.e. que l'interface et l'implantation peuvent être décorréliées. Comme le montre la Figure 17, l'implantation des services peut alors être gérée par l'intergiciel d'exécution et dynamiquement modifiée sans que l'application n'en soit perturbée.

Dans MySIM, nous effectuons cette intégration spontanée avec une approche basée sur les événements et réagissons à deux événements particuliers :

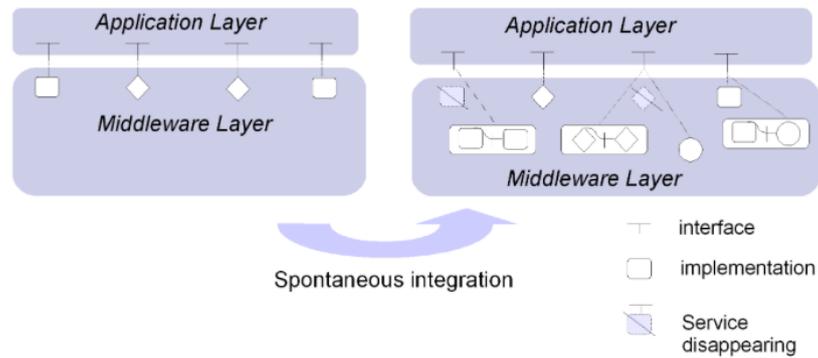


FIGURE 17 – Intégration spontanée de services avec MySIM

- La disparition de service : lorsqu'un service disparaît de l'environnement ambiant (arrêt de machine, mobilité, etc.), MySIM cherche dans l'environnement un service fournissant la même fonctionnalité et le remplace de façon transparente pour l'application. Ce nouveau service peut présenter une ou plusieurs implantations différentes et fournir également une qualité de service différente.
- L'apparition de service : quand un nouveau service apparaît dans l'environnement, il va proposer une nouvelle interface et implantation, mais celles-ci ne sont pas forcément compréhensibles pour une application qui a été développée sur d'autres bases d'interfaces. La première tâche de MySIM est de transformer ce service dans un formalisme compréhensible de toutes applications. MySIM cherche ensuite si ce nouveau service est profitable aux applications existantes, en cours d'exécution, en vérifiant si ses interfaces correspondent syntaxiquement et peuvent fournir une sémantique plus riche/appropriée ou si celui-ci peut offrir une qualité de service plus importante. Cette recherche s'effectue en essayant les différentes combinaisons possibles avec les services de l'environnement en les transformant, composant et adaptant.

Cette recherche s'effectue avec un cycle de vie et une condition d'arrêt décrits dans [Ibrahim et al. 2007b].

Intégration transparente L'intégration spontanée repose sur trois mécanismes de modifications de services - cf Figure 18 :

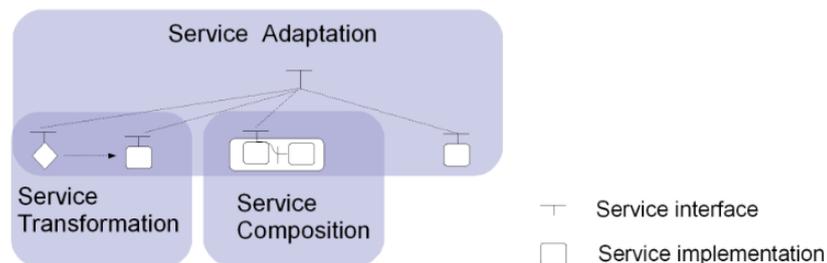


FIGURE 18 – Méthodes d'intégration avec MySIM

- La transformation de service : lorsqu'un service apparaît, pour qu'il puisse être utilisé par une application, il faut que la tech-

nologie utilisée soit compatible avec celle de l'application. Nous le transformons alors dans un modèle de service générique et pouvons ensuite le 'mapper' à différentes technologies avec une couche d'enrobage (*Wrapper*). Comme le montre la Figure 18, la transformation de service agit donc sur la partie implantation des services.

- L'adaptation de service : celle-ci garantit de façon transparente l'exécution de l'application en cherchant des interfaces de services équivalentes, si de nouveaux évènements de services interviennent. Comme le montre la Figure 18, cette adaptation agit donc par de la substitution d'interfaces.
- La composition de services : celle-ci fournit de nouveaux services - des services *composés* - en créant un nouveau flot d'invocation entre des services deux à deux et en publiant une nouvelle interface qui est le point d'entrée de ce nouveau service. Comme le montre la Figure 18, cette composition agit donc sur le flot du graphe d'appels entre services.

Architecture de MySIM MySIM est composé de quatre services (cf Figure 19) : (1) *Translator Service* - responsable de la transformation des services, (2) *Generator Service* - responsable des propriétés fonctionnelles - les relations d'équivalences, (3) *Evaluator Service* - responsable des propriétés non-fonctionnelles - la qualité de service et enfin (4) *Builder Service* - mettant en place l'intégration technique réelle, enregistrant et surveillant les services.

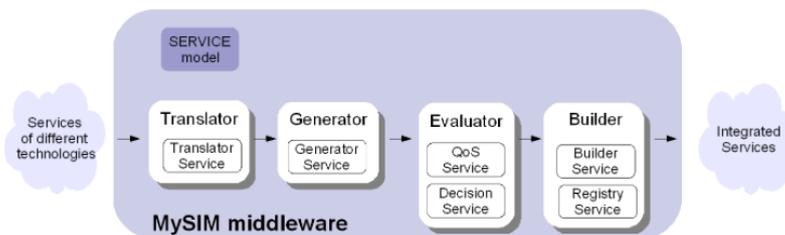


FIGURE 19 – Architecture de MySIM

Le service *Translator* transforme les services de différentes technologies dans notre modèle générique de service. Cette opération extrait les signatures des opérations, leur description sémantique, les implantations des opérations, et la description de la qualité de service. Dépendant de la technologie, cette opération peut être plus ou moins difficile. Nous proposons dans MySIM la correspondance entre les services OSGi [Alliance 2007], les WebServices [Mockford 2004] et les composants Fractal [Bruneton et al. 2006].

Le service *Generator* est responsable des correspondances entre les services au niveau des interfaces (cf Figure 20). Il vérifie toutes les relations d'équivalence entre services - composés, transformés, substitués - aussi bien syntaxiquement avec les signatures des opérations, que sémantiquement sur les ensembles d'opérations. Deux opérations sont syntaxiquement (respectivement sémantiquement) équivalentes si leurs entrées, sorties et concepts sont syntaxiquement (respectivement sémantiquement) équivalents. L'équivalence syntaxique est une simple correspondance de type sur les entrées/sorties et une correspondance de noms d'opérations pour les concepts (*String Matching*).

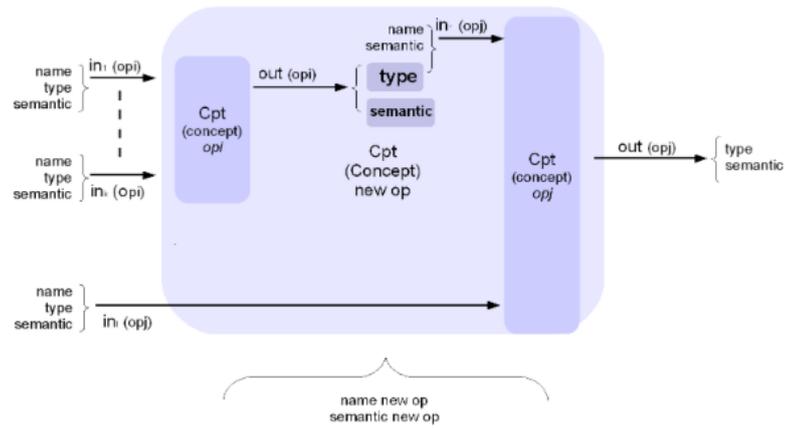


FIGURE 20 – Correspondance syntaxique et sémantique d’interfaces de services dans MySIM

L’équivalence sémantique utilise la relation sémantique définie par Paolucci [Paolucci et al. 2002]. Elle définit quatre niveaux d’équivalence sémantique entre deux concepts : *Exact*, *Plug In*, *Subsume*, et *Fail*. Nous stipulons qu’un concept peut se substituer à un autre dans le cas *Exact* et *Plug In*. Deux services peuvent donc se composer ou substituer si syntaxiquement (et potentiellement sémantiquement) les sorties d’une opération peuvent être utilisées comme entrées d’une autre opération.

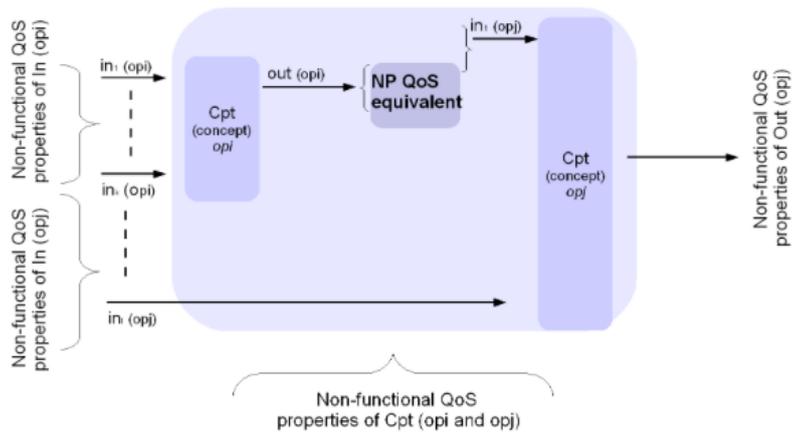


FIGURE 21 – Correspondance de qualité de service non-fonctionnelle dans MySIM

Le service *Evaluator* est chargé d’évaluer les relations d’équivalence précédentes en analysant leurs propriétés non-fonctionnelles de qualité de service. Cette évaluation s’effectue pour toute substitution, adaptation, composition, mais nous portons une attention particulière à la composition. En effet, une composition de services qui peut être fonctionnellement correcte peut très bien ne pas finalement correctement s’exécuter car les propriétés non-fonctionnelles ne s’enchaînent pas dans la composition. Nous définissons une métrique $QoS_{Degree}(opi, opj)$

comme mesure à quel point une opération est proche d'une autre en terme de qualité de service.

$$QoS(op_i, op_j) = \sum_{k=1}^{|\mathcal{N}p_{op_i}|} w_k * deg(npk_{op_i}, npk_{op_j})$$

où w_k est le poids assigné à chaque propriété non-fonctionnelle avec la condition $\sum_{k=1}^{|\mathcal{N}p_{op_i}|} (w_k) = 1$

Le plus proche w_k est de zéro, le plus important est la propriété npk_{op_i} . Ce degré est normalisé entre 0 et 1 en utilisant la standardisation z-score. Le z-score indique de combien et dans quelle direction, une propriété dévie de sa distribution moyenne. Nous utilisons le z-score pour pouvoir comparer tout type de propriétés non-fonctionnelle, quelle que soient leurs échelles de valeurs.

Evaluations Nous avons développé MySIM avec la plate-forme OSGi Apache Felix [Felix]. Nous y avons implanté notre modèle générique de service et avons construit les règles de traduction vers les services OSGi [Alliance 2007], les WebServices [Mockford 2004] et les composants Fractal [Bruneton et al. 2006]. Les descriptions sémantiques des services OSGi et Fractal sont comprises dans les descripteurs de *bundles* - composants de déploiement. La description sémantique des WebServices est faite via leur description WSDL [Christensen et al. 2001]. La correspondance sémantique est effectuée en interne en utilisant un raisonneur et des ontologies OWL-S [Martin et al. 2004].

Nous avons évalué un cas d'utilisation avec un scénario réel sur 3 laptops incluant 10 services (*Webcam, storage service, printer service* avec différentes implantations) et nous l'avons ensuite étendu à 100 services pour évaluer l'extensibilité.

La génération et composition technique (*Builder Service*) est rapide et prend de l'ordre de 1s.

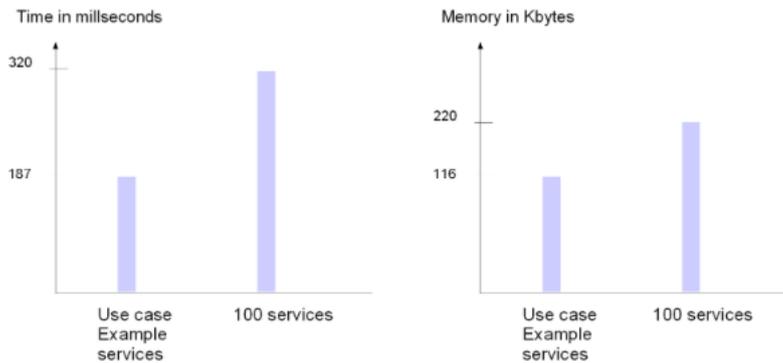


FIGURE 22 – Temps d'exécution et consommation mémoire de l'équivalence syntaxique dans MySIM

La vérification d'équivalence syntaxique est également efficace - cf Figure 22, en quelques centaines de millisecondes et 200 ko de mémoire et peut donc tout à fait s'exécuter dans des environnements contraints en ressources.

La comparaison sémantique est plus lente - cf Figure 23 : 12 secondes pour 10 services et 55 secondes pour 100 services. La consommation mémoire n'est pas négligeable non plus avec un besoin de 7M.

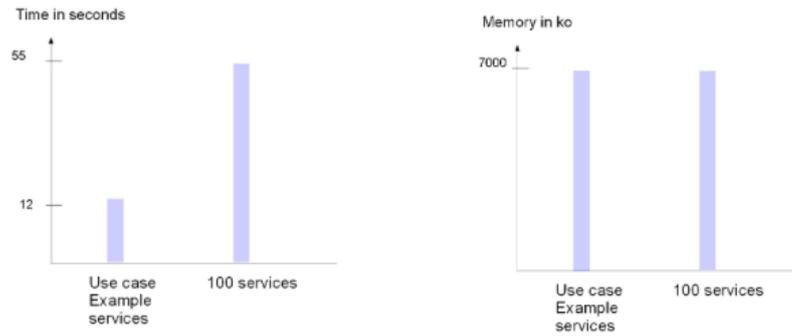


FIGURE 23 – Temps d'exécution et consommation mémoire de l'équivalence sémantique dans MySIM

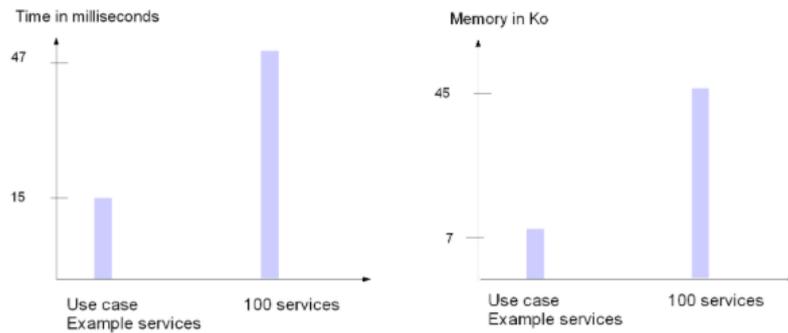


FIGURE 24 – Temps d'exécution et consommation mémoire du calcul de degré QoS dans MySIM

Le calcul du degré de qualité de service est lui efficace avec seulement 50 ms et 50 ko nécessaires pour trouver le meilleur compromis.

Ces résultats nous montrent qu'une substitution ou composition de services syntaxique avec qualité de service est tout à fait possible à faible coût pour des environnements contraints. Pour une substitution ou composition de services sémantique, il faut prévoir des terminaux un peu plus puissant - smartphones, gateways, RPI, etc. - et ne pas cibler une mobilité forte - piétonne ou véhiculaire - car, bien que ces adaptations soient transparentes, sans discontinuité de service et que l'utilisateur ne perçoive pas le coût, le bénéfice de l'intégration n'arrive qu'au bout d'une minute.

VERS DES INTERGICIELS DISTRIBUÉS, PERFORMANTS ET LARGE-ÉCHELLE

*A distributed system is one in which the failure of a computer
you didn't even know existed
can render your own computer unusable.*

— Leslie Lamport

Avec l'évolution du nombre de smartphones, avec le nombre croissants de capteurs annoncés, le passage à l'échelle est actuellement un des grands enjeux des environnements ambiants. Pendant un an et demi, j'ai travaillé comme professeur invité à SHANGHAI JIAO TONG UNIVERSITY sur ce thème. Collaborant avec le professeur JIAN CAO, j'ai eu la chance de pouvoir officier comme encadrant de deux de ses doctorants chinois et de pouvoir complètement les guider sur ces sujets et réaliser des travaux de très haut niveau, ceux-ci n'étant que résumés ici :

- Pri-REIN - partie des travaux de thèse de SHIYOU QIAN [Qian et al. 2015a] - présente, en Section 5.1, un intergiciel orienté-message à base de publications / abonnements intégrant efficacité, passage à l'échelle et qualité de service.
- CANDS - partie des travaux de thèse de DINGYU YANG [Yang et al. 2014] - présente, en Section 5.2, un système distribué permettant de gérer d'important flux de données sur de très grands graphes, tout en préservant un temps de réponse de centaine de millisecondes.

5.1 PRI-REIN

OBJECTIFS Dans les environnements ambiants - en raison de la mobilité et de la nature versatile des services à disposition - les applications effectuent souvent leur diffusion d'information en utilisant des messages asynchrones (*Message-Oriented Middleware*). Un des paradigmes existants pour développer facilement une application asynchrone est le mécanisme d'éditeurs / abonnés (*Publishers / Subscribers*). Les éditeurs publient des messages rentrant dans une certaine catégorie (*topic*) ; les abonnés s'inscrivent à la catégorie et reçoivent les messages correspondant à celle-ci.

Lorsqu'un nœud du réseau reçoit un message, un des défis est donc de savoir si celui-ci correspond aux différentes catégories d'un ou de plusieurs abonnés pour les leur rediriger (Event Matching). Avec Pri-REIN, nous nous intéressons à la notion de qualité de service dans cette correspondance de messages. Nous proposons : (1) une métrique permettant rapidement de déterminer la correspondance des catégories et (2) l'ajout d'une priorité aux différents abonnés et la garantie de délais croissants - pour le temps de calcul de correspondance - respectant l'ordre de priorité.

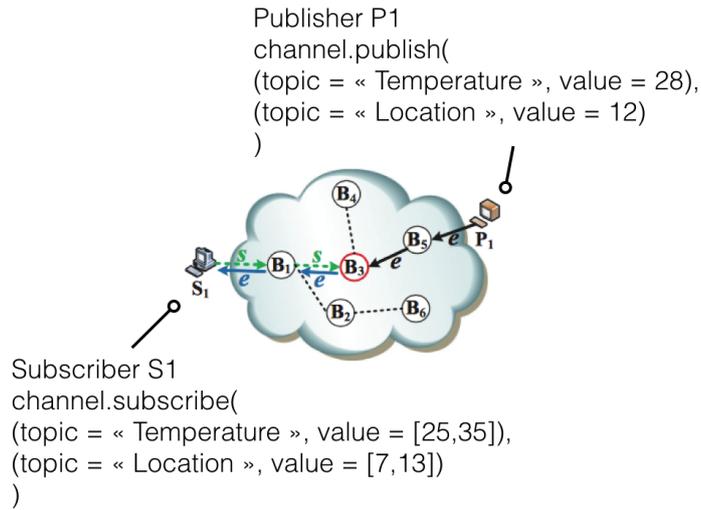
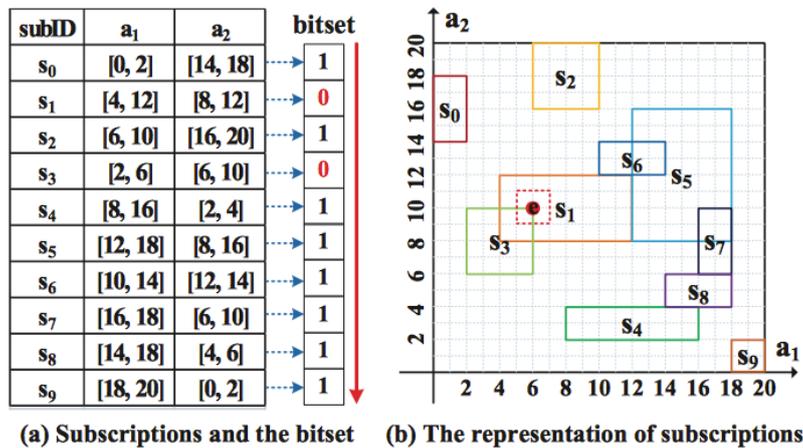


FIGURE 25 – Pri-REIN - exemple

CONTRIBUTION ET RÉSULTATS La Figure 25 présente un exemple de correspondance entre message et catégories que nous souhaitons effectuer. Nous avons modélisé les catégories avec un identifiant *topic*, et surtout un intervalle de valeurs. L’abonné S1 s’intéresse ainsi aux températures entre 25 et 35°C aux endroits identifiés entre 7 et 13. L’éditeur P1 se trouve à l’endroit 12 et publie une valeur de température de 28°C. Pour les nœuds de réseau - B1 et B3 ayant reçu la souscription -, la tâche est donc de savoir si ce message intéresse S1.



(a) Subscriptions and the bitset (b) The representation of subscriptions

FIGURE 26 – Problème géométrique du point d’inclusion dans REIN

Problème géométrique du point d’inclusion Comme le montre la Figure 26(a), sur un nœud du réseau, nous avons une liste d’abonnés (s₀, s₁, ...) avec leurs intervalles de valeurs d’intérêt pour chaque catégorie (a₁, a₂, ...). Et lorsqu’un message arrive, la méthode naïve consiste à parcourir cette table pour déterminer les lignes - et donc les abonnés - qui correspondent ou non. Cette méthode ne passe évidemment pas bien à l’échelle, aussi nous avons modélisé cette recherche par la résolu-

tion géométrique du point d'inclusion dans un rectangle [Atallah 1983]. Comme présenté dans la Figure 26(b), les intervalles de valeurs peuvent être représentés par des rectangles et - par exemple -, lorsque le message ($a_1=6, a_2=10$) arrive, il s'agit de déterminer dans quel rectangle ce point est inclus. L'exemple donné ici est en 2D pour deux contraintes d'intervalles de valeurs; il serait en n dimensions pour n contraintes.

Nous avons résolu ce problème avec un arbre équilibré d'intervalles (*H-tree*). Dans l'exemple donné, supposons que les rectangles soient équi-répartis avec une racine de l'arbre au centre géométrique ($a_1=10, a_2=10$), dès que je compare mon point à la racine de l'arbre, je ne vais aller que dans la branche d'intervalle $a_1=[1-10]$ et peux donc éliminer - avec une seule comparaison - tout les rectangles de la partie droite (s_5, s_7, s_8, s_9). Donc plutôt que de parcourir la liste pour trouver les correspondances, nous parcourons rapidement en arbre en identifiant les non-correspondances (bitset=1 dans la Figure 26).

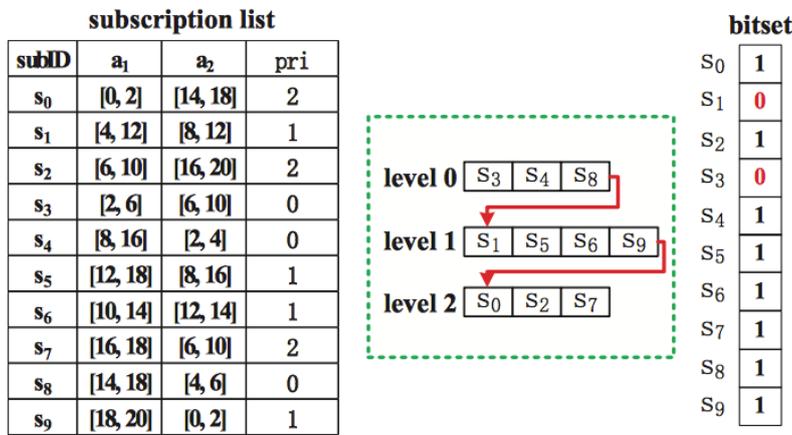


FIGURE 27 – Ajout de la priorité dans la liste d’abonnés Pri-REIN

Nous ajoutons ensuite une priorité sur les abonnés - cf Figure 27, ce qui nous permet de leur donner une qualité de service adaptée. Dans l'exemple précédent où les abonnés s_1 et s_3 correspondent au message, s_3 ayant une priorité supérieure (level 0), il sera servi en premier.

Performance et passage à l'échelle Comme le montre les Figures 28 et 29, les performances de REIN - sans l'ajout de la priorité - sont déjà très intéressantes. Comparé à SIENA [Carzaniga et Wolf 2003], TAMA [Zhao et Wu 2011] et même un H-tree classique qui présentent des courbes linéaires, REIN reste quasiment constant en fonction du nombre d'abonnés (linéaire avec une pente très faible). En fonction du nombre de contraintes, REIN gagne un ordre de magnitude en allant 10 fois plus vite que SIENA et TAMA.

En ajoutant les priorités - dans la Figure 30, les temps de calculs de correspondance - qui étaient fixes pour REIN - vont évoluer en fonction du niveau de priorité. Les abonnés Pri-REIN en priorité 0 gagnent 10% par rapport à REIN. Le point de croisement des courbes s'effectue au niveau de priorité 3.

Pri-REIN-S est une petite optimisation qui consiste - quand deux abonnés ont la même priorité - à servir celui qui s'intéresse aux catégories les plus demandées plutôt que de garder l'ordre du premier-arrivé

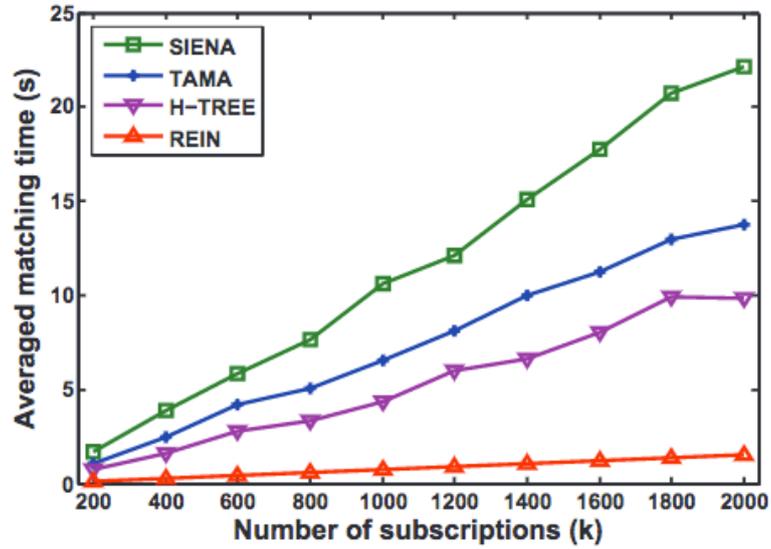


FIGURE 28 – Performance de REIN en fonction du nombre de souscriptions

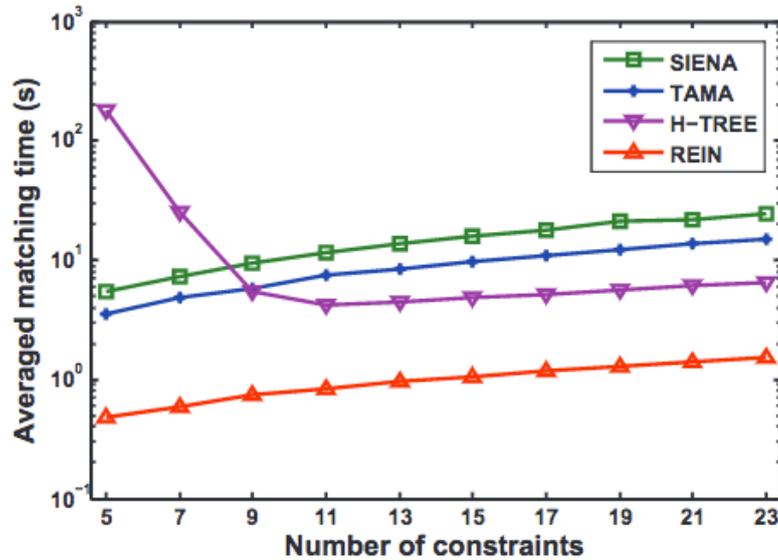


FIGURE 29 – Performance de REIN en fonction du nombre de contraintes

premier-servi. Cette optimisation permet aux abonnés de priorité o de gagner 15% par rapport à REIN et d'avoir plus de souplesse avec un niveau de priorité de plus en dessous de REIN - le point de croisement s'effectuant au niveau 4.

5.2 CANDS

OBJECTIFS Une des applications critiques dans notre société moderne est la gestion du trafic routier en temps-réel. Les systèmes de transports intelligents (*Intelligent Transportation Systems - ITS*) s'appuient sur une large remontée de données GPS pour optimiser les

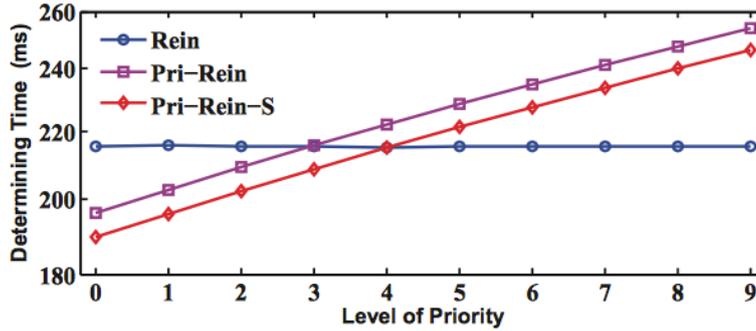


FIGURE 30 – Ajout de la priorité avec Pri-REIN

trajets, mais ce large volume de données pose des problèmes à traiter [Guerrero-Ibáñez et al. 2012]. Les systèmes de navigation, de recommandation - et encore plus des services comme les taxis, les bus, les ambulances ou les pompiers - ont besoin de notifications et alertes en temps réel pour adapter leurs trajets de façon optimale en fonction du trafic. Les systèmes actuels de gestion de graphes distribués soit ne supportent pas les graphes dynamiques, soit permettent des techniques d'approximation mais ne peuvent pas garantir des plus courts chemins.

Nous proposons CANDS - *Continuous Optimal Navigation via Distributed Stream Processing*, dont les contributions principales sont (1) une solution distribuée avec mécanismes asynchrone pour répondre en temps réel à des requêtes de plus court chemin dans un graphe dynamique, et (2) une solution permettant de trouver rapidement les requêtes affectées par des modifications de trafic, et de mettre à jour leurs plus court chemins tout en garantissant l'optimalité du processus.

CONTRIBUTION ET RÉSULTATS L'architecture de CANDS est déployée sur un cluster de machines comme présenté dans la Figure 31. Elle consiste en un ensemble de *Processing Element* (PE) pouvant s'exécuter de manière distribuée - par exemple, un PE par cœur processeur - et communiquant par messages asynchrones.

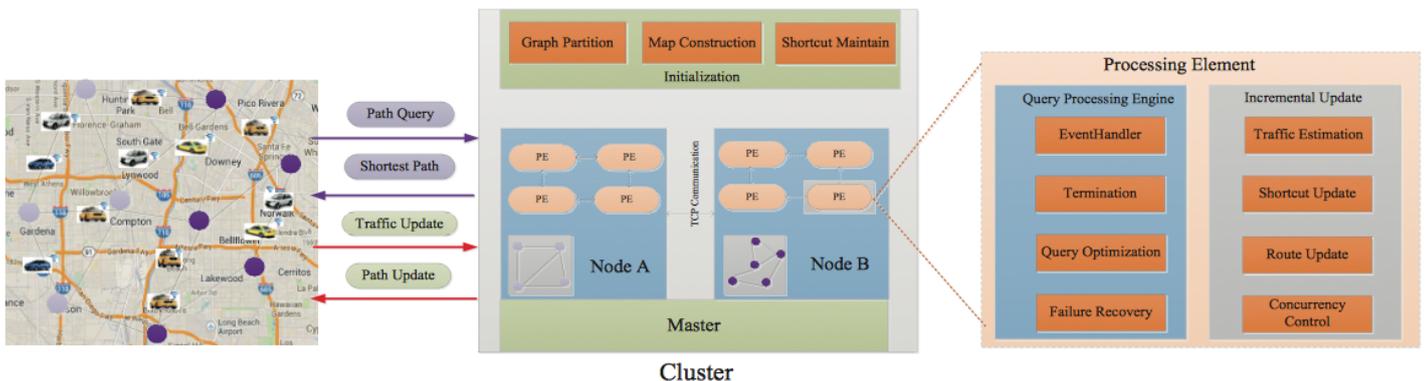


FIGURE 31 – Architecture de CANDS

CANDS permet de gérer deux types de flux de données : (1) les requêtes de navigation - requêtes de plus court chemin d'un point à

un autre (*Time Dependent Single-Source Short Path queries* - TD-SSSP), et (2) les mises à jour de localisation de véhicules (*Reverse Short Path for edge update* - RSP). Les premières requêtes calculent un chemin sur la base des conditions de trafic actuelles, et les secondes mettent à jour l'estimation de ce trafic et impliquent donc également des mises à jour des plus courts chemins concernés par ces informations.

Partitionnement de graphe et index Dans un premier temps, l'ensemble du réseau routier est découpé par un partitionnement de graphe en M sous-graphes en utilisant l'algorithme d'équilibrage METIS [Karypis et Kumar 1998], où M est le nombre de machines cibles. Cette étape est critique pour la performance. Car, premièrement, le temps de traitement d'une requête est déterminé par rapport à la tâche la plus lente et cet équilibrage élimine ce goulot d'étranglement. Et deuxièmement, la communication entre arc du même sous-graphe s'effectue sur la même machine, réduisant les entrées/sorties réseau. Dans un deuxième temps, chaque sous-graphe est également découpé en N partitions plus petites et chacune de ses partitions est assignée à un *Processing Element* pour permettre à chaque ressource de calcul de chaque machine d'être pleinement utilisée. Ce nombre N est un paramètre utilisateur dont la meilleure valeur va être estimée en fonction du graphe.

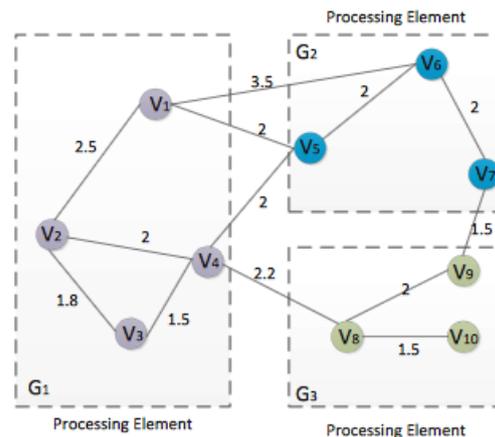


FIGURE 32 – Partitionnement de graphe dans CANDS

La Figure 32 présente un exemple de graphe découpé en trois partitions. Pour chaque partition G_i , une liste de paires permet de conserver les arcs de bords reliés aux partitions voisines. Par exemple, pour la partition G_1 , les arcs conservés sont pour les nœuds $\{v_1, v_4\}$ et la liste est $L_1 = \{ \langle v_1, v_6, G_2 \rangle, \langle v_1, v_5, G_2 \rangle, \langle v_4, v_8, G_3 \rangle, \langle v_4, v_5, G_2 \rangle \}$.

Un ensemble de raccourcis est également maintenu pour faciliter et accélérer le temps de traitement. Un raccourci (*Graph Shortcut*) est le plus court chemin entre nœuds de bord à l'intérieur d'une partition. Entre chaque nœud de bord, nous pré-calculons un plus court chemin avec Dijkstra et conservons les résultats. Dans l'exemple de la Figure 32, le plus court chemin $v_1 \rightarrow v_2 \rightarrow v_4$ avec une distance de 4.5 est maintenu dans G_1 . A noter, ce plus court chemin ne garantit qu'une optimalité locale et non globale. Un meilleur résultat peut parfaitement être trouvé en utilisant des arcs d'autres partitions. Par exemple, $v_1 \rightarrow v_5 \rightarrow v_4$ avec une distance de 4 est le réel plus court chemin entre v_1

et v_4 . Quand une mise à jour intervient dans le statut d'une route, les raccourcis sont recalculés pour conserver cette propriété d'optimalité locale.

Optimisation par combinaison de messages Après avoir traversé une partition pour y calculer des chemins, des messages sont envoyés aux partitions voisines en utilisant les nœuds de bord. Pour chaque nœud de bord, un message doit être généré et envoyé à son homologue connecté dans la partition voisine. Il est possible qu'une partition ait plusieurs arcs connectés à une même partition voisine. Par exemple, dans la Figure 32, il y a deux nœuds de bord v_1 et v_4 dans la partition G_1 . v_1 doit envoyer deux messages à la partition G_2 , et v_4 doit envoyer 1 message à G_2 et 1 message à G_3 . Cet envoi multiple de message est trop coûteux au niveau du réseau. Pour diminuer ce coût, nous proposons un 'combineur' de message (*Message Combiner*) qui plutôt qu'émettre plusieurs fois, regroupe ensemble les messages à même partition destination. Dans la Figure 32, nous combinons les messages de calcul de route entre G_1 et G_2 en un seul message et l'envoyons à G_2 .

Optimisation par broadcast Une autre technique d'optimisation est de diminuer l'inondation de messages en broadcastant le plus court chemin courant à tout les *Processing Element* quand un meilleur résultat est trouvé dans la partition G_t . De cette manière, chaque partition G_p maintient une variable $\delta_{s \rightarrow t}$, indiquant la meilleure distance courante. Le but est que les partitions 'lointaines' d'un nœud s peuvent apprendre cette information avant de recevoir les messages de leurs partitions voisines. Alors, la propagation de message peut s'arrêter plus rapidement sans impliquer un grand nombre de partitions. Nous définissons la distance d'un nœud s à une partition comme suit :

$$d(s, G_p) = \min_{v \in G_p.\text{border}} d(s, v)$$

Si $d(s, G_p) > \delta_{s \rightarrow t}$, la propagation peut s'arrêter. Bien que cette condition ne soit pas toujours satisfaite du fait que les communications soient asynchrones, cela permet d'empêcher l'envoi de résultats partiels pires que ceux de la partition source G_s jusqu'à n'importe quelle autre partition. Les partitions lointaines peuvent alors ne pas répondre aux messages de calculs de route si le résultat trouvé est pire que le résultat reçu par broadcast. Si la partition G_p ne reçoit pas de message de calcul de route, nous avons alors $d(s, G_p) \geq d(s, t)$. Cela signifie que cette partition est trop lointaine du nœud s et que le calcul du plus court chemin devra passer par les partitions.

Expérimentations avec traces réelles CANDS a été testé avec plusieurs jeux de données : traces synthétiques de San Francisco (320k nœuds, 800k arcs), Florida (1M nœuds, 2M arcs), California et Nevada (1.8M nœuds, 4.6M arcs), Eastern USA (3.5M nœuds, 8.7M arcs), et Central USA (14M nœuds, 34.3M arcs) [DIMACS] et trace réelle de Beijing (155k nœuds, 337k arcs) [Zheng et al. 2011]. Seuls les résultats de la trace réelle seront présentés ici. Cette trace comprend les trajectoires de 10.3k taxis du 2 au 8 février 2008 à Beijing. Les coordonnées GPS sont plaquées sur les nœuds du graphe et les trajectoires originales sont calculées. Nous calculons les poids sur les arcs par calcul du temps

moyen de voyage des taxis sur un intervalle de temps donné. Avec les points de départ des passagers, nous pouvons générer des requêtes de destination - 200 requêtes de longueur moyenne de 200 nœuds sont extraites.

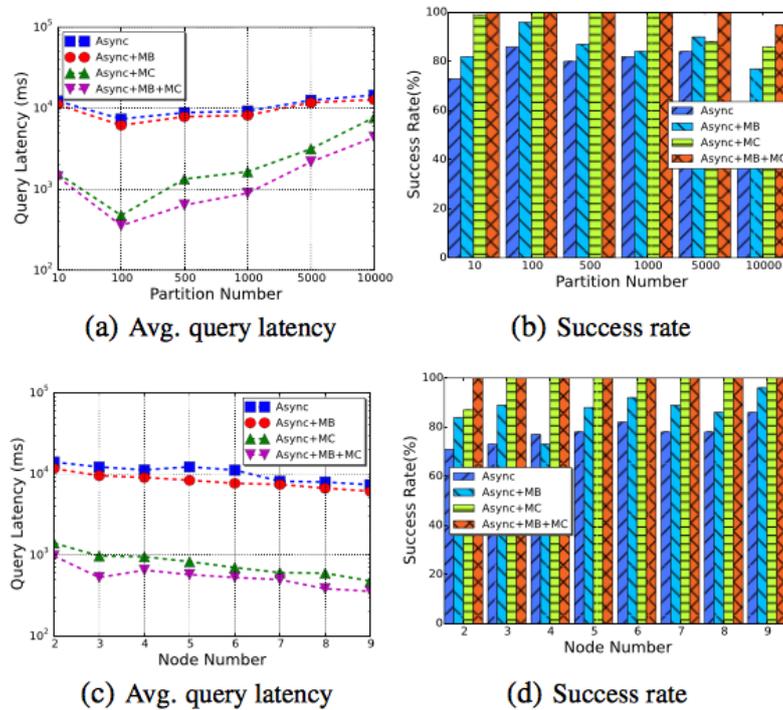
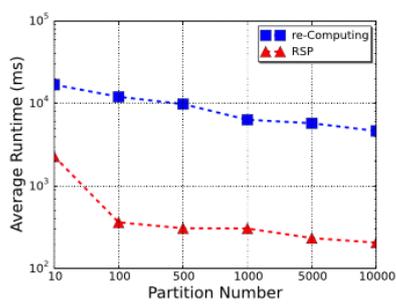


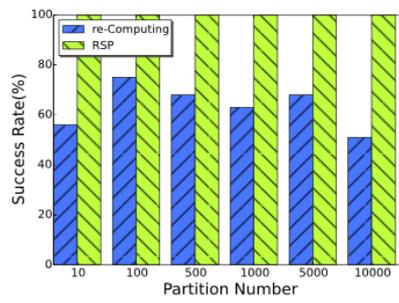
FIGURE 33 – Evaluation de performance des requêtes SSSP dans CANDS

La Figure 33 montre la latence et le taux de succès des requêtes de calcul de plus court chemin en fonction du nombre de machines et du nombre de partition dans le graphe. Le meilleur nombre de partition est de 100 avec des performances de 500ms et un taux de succès de 100%. CANDS arrive à supporter ~10000 requêtes simultanées tout en conservant un taux de succès proche de 100%. Le nombre de machine dans le cluster permet de gagner un facteur 2 en passant de 1000ms pour 2 machines à 500ms pour 9 machines. Le gain obtenu avec l'optimisation en message broadcast (MB) est léger en terme de latence et un peu plus importante en terme de taux de succès. C'est par contre l'optimisation avec message combineur (MC) qui améliore grandement les performances - de 10000ms à 600ms en latence et 100% de taux de réussite. Les deux optimisations combinées obtiennent les meilleures performances.

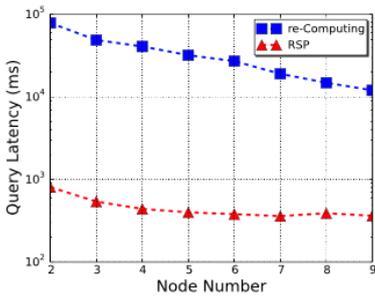
Nous avons également évalué les performances de CANDS en cas de mises à jour de requêtes lorsque les conditions de trafic changent (cf Figure 34). Nous comparons ces performances au recalcul complet de la requête de plus court chemin. Le gain est significatif avec des temps de latence de 500ms et un taux de succès de 100% par rapport à une latence de 10000ms et un taux de succès de 60-70%.



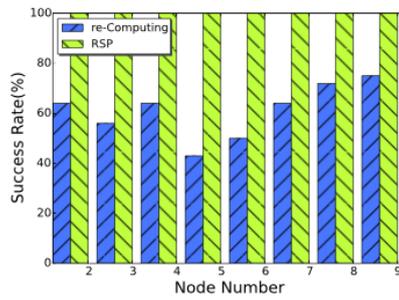
(a) Avg. query latency



(b) Success rate



(c) Avg. query latency



(d) Success rate

FIGURE 34 – Evaluation de performance des requêtes RSP dans CANDS

*Computer languages of the future will be more concerned with goals
and less with procedures specified by the programmer.*

— Marvin Minsky [Minsky 1970]

Pour aller plus loin que des intergiciels efficaces et distribués permettant de passer à l'échelle, la dernière étape consiste à rendre les intergiciels complètement autonomes. Chaque nœud distribué possède alors ses données, effectue ses calculs et prend seul ses décisions - potentiellement en tenant compte de données et décisions environnantes - mais présente un comportement indépendant. Les travaux présentés dans cette section examinent cela dans le cadre de la Ville Intelligente. Un sous-ensemble des travaux de thèse de MARIE-ANGE LÈBRE [Lèbre 2016] présente un modèle de prise de décision autonome avec de l'information partielle distribuée et partagée entre véhicules.

6.1 CROWDSOURCING VÉHICULAIRE

OBJECTIFS Comme nous l'avons précédemment vu dans la Section 5.2, la fluidité du trafic véhiculaire est un enjeu majeur. Dans le réseau urbain, les temps de parcours des sections de route varient constamment au cours d'une journée et des événements non prévisibles, comme des bouchons fantômes ou des accidents, peuvent le perturber. La première préoccupation est donc d'éviter ces zones de congestion, nous parlons du SPP (*Shortest Path Problem*). Les solutions statiques du SPP ne prennent en compte, ni les informations trafic en temps réel, ni les parcours de l'ensemble des véhicules (ce qui peut les amener tous au même endroit à un instant t). Nous identifions ainsi deux problèmes : l'accès à une information en temps réel et la coopération entre les acteurs du réseau. Des modèles centralisés répondent à cette problématique. Ils possèdent une connaissance de la totalité du réseau mais cela implique de traiter une quantité importante de données, rendant une réponse en temps réel difficile (mais pas impossible comme vu dans la Section 5.2). Les systèmes décentralisés sont également beaucoup étudiés. Parmi ceux-ci, nous nous intéressons aux solutions bio-inspirées. Le modèle le plus utilisé a été introduit par Marco Dorigo et Gianni Di [Dorigo et Di Caro 1999] et s'inspire du comportement naturel des fourmis. Elles utilisent des phéromones pour guider la colonie vers une source de nourriture. Les fourmis parcourent différents chemins et échangent leurs informations avec leurs semblables. Les insectes procèdent à ce qu'on appelle du *Crowdsourcing* : c'est l'utilisation de la créativité, de l'intelligence et du savoir-faire d'un grand nombre d'individus pour réaliser certaines tâches. L'algorithme des fourmis appelé ACO (*Ant Colony Optimization*) a été utilisé dans différents domaines, comme la coloration des graphes ou encore le problème du voyageur de commerce [Dorigo et Gambardella 1997]. Dans le SPP, ACO s'utilise en amont du parcours, à l'instar de l'algorithme de Dijkstra et de A*.

Nous proposons un modèle véhiculaire de crowdsourcing où le véhicule est assimilé à une fourmi cherchant son chemin en temps réel. Au cours de leurs déplacements, les véhicules échangent leur connaissance du réseau avec les autres véhicules. Ils calculent donc leur trajet avec une information partielle du réseau. De plus, ces données, assimilées à des phéromones, possèdent un délai et un degré de fiabilité. Nous nous intéressons ici à la communication V2V et la diffusion en pair-à-pair. Cela nous permet de créer un réseau de grande envergure en connectant les véhicules entre eux.

Les deux principales contributions de ces travaux sont (1) la proposition et l'étude de trois modèles véhiculaires de crowdsourcing - inspirés et adaptés des algorithmes de fourmis, et prenant en compte une quantité et proximité d'informations différentes, et (2) d'établir des scénarios d'évaluation de ces modèles permettant de se confronter à des situations imprévues.

CONTRIBUTION ET RÉSULTATS L'algorithme des fourmis est inspiré par leur comportement pour trouver des sources de nourriture dans un environnement inconnu. A chaque itération de l'algorithme, chaque fourmi voyage d'un nœud à un autre nœud selon une probabilité. Cette probabilité de choisir le chemin est définie par la qualité du lien déterminée par la quantité de phéromone déposée. Contrairement à la fourmi, le véhicule ne cherche pas un lieu, il connaît sa destination. Dans notre modèle, il est guidé par sa carte de phéromone à chaque pas de temps (cf Figure 35). La qualité d'un lien est définie également par le taux de phéromone qui indique sa fluidité. La fiabilité de l'information est représentée par l'évaporation. Lorsque deux véhicules se croisent, ils échangent leurs cartes de phéromones en conservant les valeurs les plus à jour.

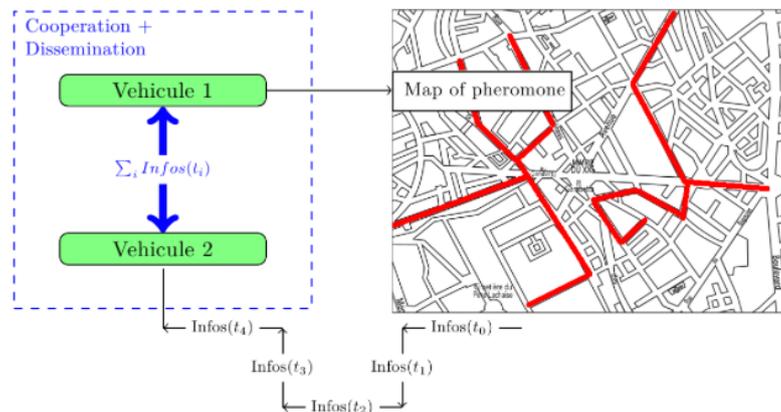


FIGURE 35 – Echange de cartes de phéromones dans un réseau véhiculaire

Phéromones et trafic *Définition : carte des phéromones*

Le graphe G - graphe orienté représentant les intersections et routes du réseau routier - est pondéré et chaque véhicule connaît une partie de G au temps présent t . Nous notons $G^m(t)$ le graphe connu par le véhicule m au temps t . C'est un sous-graphe de G : $E^m \subset E$. Les poids sur les liens de l'ensemble E^m représentent l'information en phéromone connue par le véhicule m au temps t . Nous notons $w_{ij}^m(t_0)$, le temps

de parcours sur le lien e_{ij} enregistré au temps t_0 par le véhicule m . Le taux de phéromone sur le lien e_{ij} est alors noté :

$$\tau_{ij}^m = \frac{w_{ij}}{w_{ij}^m(t_0)}$$

w_{ij} est le temps de parcours du lien e_{ij} avec la vitesse maximale autorisée. Le couple de données (τ_{ij}^m, t_0) est associé à chaque lien e_{ij} . τ_{ij}^m est le taux de phéromone sur le lien e_{ij} , il représente la qualité du lien en termes de temps de parcours et t_0 correspond au moment où le véhicule m enregistre la valeur. Ce temps représente la validité de la phéromone. Plus le taux de phéromone est proche de 1, plus le temps de parcours est faible sur le lien, et plus il est proche de 0, plus le lien est congestionné. Enfin, plus $(t - t_0)$ est élevé, plus l'information est ancienne.

Au début, tous les liens ont un taux de phéromone initiale fixé à $\tau_0 = 0.5$. Cette valeur va permettre l'équilibre entre le cas où le véhicule ne possède pas d'information sur le lien et le cas où le lien n'est pas vraiment congestionné, ni vraiment fluide.

Le véhicule m est capable d'échanger ses informations avec les autres véhicules, comme une fourmi. Quand le véhicule traverse un lien, il enregistre le temps de parcours sur son graphe $G^m(t)$ en termes de phéromone, ensuite il échange ce dernier avec les autres véhicules qu'il croise. Le véhicule complète donc son savoir grâce à la connaissance des autres véhicules.

Définition : Évaporation de la phéromone

Quand le temps passe, la fiabilité de l'information diminue, nous définissons ce procédé par l'évaporation. Soit t le temps présent, t_0 le temps où le véhicule a enregistré la phéromone sur le lien e_{ij} , $(t - t_0)$ représente alors le temps écoulé depuis l'enregistrement des informations sur e_{ij} . Plus cette valeur est élevée plus l'information est ancienne et devient alors inutile pour le véhicule m . L'évaporation est définie à chaque pas de temps par :

$$\begin{cases} (\tau_{ij}, t_0) \leftarrow (\varphi \cdot \tau_{ij}, t_0) \\ (\tau_{ij}, t_0) \leftarrow (0.5, -1) \quad \text{si } (t - t_0) \geq T \end{cases}$$

avec

$$\varphi = \frac{t_0}{t}$$

φ représente le gradient de l'évaporation. T est la durée de vie du temps de parcours mesuré. Quand $(t - t_0) \geq T$, le véhicule initialise le lien à 0.5, cela signifie que la valeur enregistrée était trop ancienne. Dans tous les cas, notre modèle pénalise les informations des liens congestionnés, qu'elles soient anciennes ou récentes, grâce à τ . Ensuite, le modèle pénalise les informations anciennes grâce à φ .

Modèles de crowdsourcing Nous proposons trois modèles de collaboration et prise de décision - PKP, KPP et PPE - pour trouver son chemin à l'instar d'une fourmi. Dans les deux premiers modèles, le véhicule utilise toutes les informations rassemblées sur sa carte pour trouver son chemin jusqu'à sa destination. Dans le dernier modèle, le véhicule utilise seulement les valeurs des phéromones localisées autour de lui.

Phéromone sur k-Parcours : PKP

Si un véhicule choisit son parcours lien par lien, sans considérer sa destination, comme une fourmi, l'exploration va être longue voire infructueuse. Le véhicule connaît son arrivée, c'est pourquoi nous introduisons un algorithme de k plus courts chemins. Ces chemins vont donner la direction vers la destination et ainsi éviter des explorations inutiles trop loin de l'arrivée du véhicule. Nous utilisons l'algorithme de Yen [Luettel et al. 2012]. Il emploie n'importe quel algorithme connu afin de trouver le plus court chemin, ensuite il trouve le second plus court et ainsi de suite, jusqu'à k chemins. Ici nous utiliserons l'algorithme de Dijkstra. Notre premier modèle fonctionne de la manière suivante, pour un véhicule m au temps t :

- Calcul de k parcours entre la position du véhicule et sa destination D^m , avec les coûts sur les liens égaux à w_{ij} (temps de parcours sur le lien e_{ij} avec la vitesse maximale).
- Calcul de la probabilité de choisir le parcours, $P_n = \{R_1, \dots, R_n\}$ parmi les k chemins :

$$p(P_n) = \frac{\sum_{i=1}^n \frac{\gamma \cdot \tau_i}{n \cdot L(P_n)}}{\sum_n \sum_i \frac{\gamma \cdot \tau_i}{n \cdot L(P_n)}}$$

avec

$$\gamma = \begin{cases} 1 & \text{si } \tau_i \text{ est connu} \\ \beta & \text{si } \tau_i \text{ est inconnu} \end{cases}$$

$\beta < 1$ permet donc de donner moins d'importance au valeur de phéromone inconnue par le véhicule (c'est-à-dire égale à 0.5). Le véhicule va choisir le chemin ayant la plus haute probabilité parmi les k plus courts chemins. Ensuite, si deux parcours possèdent la même probabilité, l'algorithme choisit le chemin ayant la plus faible variance en phéromone. Nous considérons qu'un véhicule va préférer prendre deux sections de route ayant une fluidité moyenne plutôt qu'une section libre suivie d'une section complètement congestionnée.

K-Phéromone Parcours : KPP

Dans cette seconde méthode, l'algorithme calcule k plus courts parcours directement sur la carte de phéromone. Les poids considérés dans l'algorithme, sur les liens e_{ij} , sont les suivants :

$$\frac{w_{ij}}{\gamma \cdot \tau_{ij}}$$

avec

$$\gamma = \begin{cases} 1 & \text{si } \tau_i \text{ est connu} \\ \beta & \text{si } \tau_i \text{ est inconnu} \end{cases}$$

$\beta < 1$ permet donc de donner moins d'importance au valeur de phéromone inconnue par le véhicule (c'est-à-dire égale à 0.5). Ensuite la probabilité pour choisir le chemin P_n est donnée par :

$$p(P_n) = \frac{\sum_{i=1}^n \frac{\tau_i}{n \cdot L(P_n)}}{\sum_n \sum_{i=1}^n \frac{\tau_i}{n \cdot L(P_n)}}$$

Comme dans la méthode précédente *PKP*, si deux chemins possèdent la même probabilité, l'algorithme choisit le chemin ayant la variance la plus faible en phéromone.

Exploration des Parcours de Phéromone : PPE

La troisième méthode est proche du comportement des fourmis. Dans le modèle ACO, la probabilité de bouger sur un lien e_{ij} pour une fourmi dépend de deux valeurs, tout d'abord l'attractivité du lien qui est calculée par une heuristique, ensuite la trace de phéromone laissée par les autres fourmis indiquant l'efficacité des mouvements passés. Dans les algorithmes, le taux de phéromone peut être mis à jour une fois la fourmi arrivée à destination, ou à chaque pas de temps. Ainsi le taux de phéromone augmente ou diminue indiquant les bonnes ou les mauvaises décisions respectivement.

Dans cette méthode, le véhicule agit comme une fourmi, en explorant son environnement proche. L'algorithme *PPE* calcule tous les chemins possibles dans toutes les directions autour du véhicule, selon la condition suivante : $TT(P) < T$. Nous rappelons que T est la durée de vie de la phéromone enregistrée et $TT(P)$ est le temps de parcours du chemin P avec la vitesse maximale autorisée. Nous obtenons alors un ensemble de chemin : $\{P_l\}_l$. La probabilité de choisir un de ces chemins $P_l = \{R_i, \dots, R_s\}$ est donnée par :

$$\frac{\sum_{j=i}^s \xi \cdot \tau_j^\alpha \cdot (1 - \xi) \cdot \eta_{is}}{\sum_{P_k} (p(P_k))}$$

avec

$$\eta_{is} = \frac{L(R_i \rightarrow D^m)}{L(P_l) + L(R_s \rightarrow D^m)}$$

R_i est la position actuelle du véhicule m . R_s est la fin du chemin P_l . $L(R_s \rightarrow D^m)$ correspond à la distance euclidienne du plus court chemin entre R_s et la destination du véhicule $m : D^m$. Et $L(P_l)$ correspond à la distance euclidienne du chemin P_l .

Si la valeur de la phéromone est connue par le véhicule alors $\alpha = 1.5$, sinon $\alpha = 1$, nous donnons ainsi plus d'importance aux valeurs connues par le véhicule. ξ permet de donner plus ou moins d'importance à la valeur de la phéromone ou la longueur du chemin.

Scénarios et simulations Nous testons les trois modèles sur une carte réelle. Pour un scénario réaliste, nous utilisons les données du projet TAPAS Cologne [Uppoor et Fiore 2011], reproduisant le trafic urbain dans la ville de Cologne. Nous nous focalisons dans un quartier de dimension 1200 mètres par 800 mètres, sur la durée de 6h00 à 6h15, ce qui représente 1200 véhicules. Nous utilisons le simulateur SUMO (version 0.22) ainsi que l'interface de contrôle TraCI afin de contrôler les véhicules en temps réel [Krajzewicz et al. 2012]. Nous comparons nos modèles avec PDLAIS pour *Partial, Decentralized and Locally Autonomous Information Strategy* - un autre modèle que nous avons développé pour rendre plus intelligente les intersections les plus congestionnées dans la ville [Lèbre et al. 2015b]. Nous les comparons également aux cas extrêmes, à savoir le modèle sans système de guidage (WOK) et le modèle centralisé (CS). Dans ce dernier, les véhicules possèdent l'information trafic du réseau en entier, à chaque pas de temps.

Scénario de trafic classique

Comme le montre le Tableau 6, nos trois modèles de crowdsourcing véhiculaire PKP, KPP et PPE présentent de bons résultats et améliorent le temps de parcours et le temps d'attente par rapport à une situation sans information (WOK). Nos modèles sont équivalents à PDLAIS et présentent des résultats proches de la solution centralisée (CS). Notons

Real Map	WOK	PKP	KPP	PPE	CS	PDLAIS
Duration (min)	1.64	1.58	1.51	1.61	1.45	1.58
Duration SD (min)	8.02	7.56	7.02	8.8	6.19	7.58
Duration Max (min)	8.31	9.11	7	10.3	6.36	9.23
Speed (km/h)	34.74	34.76	35.33	36.17	36.14	35.20
Waiting Time (s)	22.14	19.86	17.50	15	15.21	19.46
Waiting Time SD (s)	37.66	33.14	29.66	24.6	23.73	34.53
Waiting Time Max (s)	327.0	332.0	290	294.0	181	387
Route Length (km)	0.839	0.819	0.809	0.786	0.818	0.837
Route Length SD (km)	12,36	11,76	11,58	12.39	11.83	11.84

TABLE 6 – Comparaisons de modèles de prise de décision pour le guidage en ville - scénario de trafic classique

que le modèle KPP est le meilleur en temps de parcours et PPE est le meilleur en temps d'attente.

Scénario d'accident dans le trafic

Un incident se réfère à n'importe quoi qui pourrait stopper ou ralentir le flux du trafic. Dans le réseau urbain, cela peut-être un accident de voiture ou des mauvaises conditions météorologiques par exemple. Pour simuler un accident de voiture, il suffit d'arrêter une ou des voitures pendant une durée limitée durant leurs parcours [Smith et al. 2014]. Nous stoppons donc plusieurs véhicules sur une route à trois voies possédant un flux véhiculaire important, sur la carte de Cologne.

Nous calculons le temps de parcours et le temps d'attente des véhicules qui ne se trouvent pas bloqués dans l'accident. Nous relevons également le nombre de véhicules arrivant à destination selon le modèle simulé, ce paramètre permet d'évaluer l'efficacité du modèle. En effet des véhicules vont être bloqués par l'accident, et étant donné que la route bloquée possède une forte affluence, des routes annexes peuvent aussi être encombrées. Ainsi plus le nombre de véhicules capable de contourner la zone est grand, plus le modèle est efficace. Et parmi ces derniers, si leur temps de voyage et leur temps d'attente est faible alors le modèle est performant.

Real Map	WOK	KPP	PKP	PPE	CS	PDLAIS
Percentage of vehicles that end their travel	76	89	78	85	98	78
Duration (min)	1.30	1.65	1.27	1.54	1.79	1.30
Duration SD (min)	5.06	11.11	4.93	7.34	9.92	4.92
Waiting Time (s)	11.26	23.47	11.05	15	25.59	11.43
Waiting Time SD (s)	19.07	58.77	18.57	22.53	47.50	18.92

TABLE 7 – Comparaisons de modèles de prise de décision pour le guidage en ville - scénario d'accident dans le trafic

Les modèles permettant au plus grand nombre de véhicule d'éviter la zone de l'accident et les embouteillages causés par ces derniers, sont

KPP et PPE, avec plus de 85% des véhicules arrivant à leur destination (les autres restent bloqués par l'accident) (cf Tableau 7). Cependant la moyenne et l'écart type du temps d'attente et de parcours sont plus mauvais pour KPP que PPE, ce qui signifie que les contournements trouvés par KPP sont plus longs que ceux trouvés par PPE. Les modèles PKP et PDLAIS ont des résultats proches de la situation sans système (WOK). Il est important de noter que leurs moyennes sont plus faibles que les autres car le nombre de véhicules arrivés est plus faible. PDLAIS mise sur la position de certaines intersections communicantes dans la ville, si ces dernières ne sont pas proches de l'accident, le modèle ne peut être performant au niveau de cette zone. PKP, lui, calcule les k chemins sans information sur le trafic en temps réel au départ, ceci peut créer un handicap pour les véhicules passant dans cette zone. En augmentant la valeur de k , nous pourrions peut-être améliorer la performance de ce modèle sur une carte plus grande mais la complexité de l'algorithme de Yen et son temps de calcul augmentera également.

Scénario de terrorisme

Il existe plusieurs études concernant la modélisation du transport dans des situations extrêmes comme celle du 11 septembre 2001. Plusieurs explorent les stratégies d'évacuations en prenant compte de la distribution de la population et du comportement des véhicules au sein de tel scénario [Peeta et Hsu 2009, Wein et al. 2010, Alazawi et al. 2011]. Nous utilisons les observations et les analyses de [Lambert et al. 2013] afin de simuler une attaque terroriste dans un quartier de la ville de Cologne. Nous testons ainsi la capacité de nos modèles à propager l'information et évacuer les véhicules de la zone dangereuse. Les données utilisées dans [Lambert et al. 2013] proviennent d'un état de l'art composé d'une banque de données sur le comportement de la population, dans le cas hypothétique d'une attaque sur Washington DC. Leur article étudie 1 à 2 millions de voyages, cette quantité de données est trop importante pour nos simulations. Nous utiliserons donc leurs résultats et observations sur les processus d'évacuation pour reproduire sur une carte plus petite le même scénario. Ainsi durant une attaque, le nombre de voyage est doublé : la population dans les bâtiments où dans leur maison évacue la zone avec leur voiture. De plus, le pourcentage de véhicule quittant complètement la région est de 13% des voyages totaux. Nous appliquons ces observations sur la zone la plus empruntée par les véhicules en temps normal.

Comme le montre le Tableau 8, le modèle PPE, utilisant des informations locales en phéromone autour du véhicule, obtient les meilleurs résultats en temps d'attente malgré des écarts types plus grands, ce qui signifie que certains véhicules sont lésés par rapport aux autres. KPP et PDLAIS ont de bons résultats avec les meilleurs écarts types. Ils sont les plus proches de la solution centralisée. Cependant PDLAIS est dépendant de l'infrastructure, si celle-ci se trouve être positionnée dans la zone dangereuse, ces résultats ne peuvent être assurés. Les modèles de crowdsourcing véhiculaire, eux ne dépendent pas de l'infrastructure et peuvent être performants quelque soit le lieu de l'explosion.

Scénario de catastrophe naturelle

Une catastrophe naturelle peut être un tsunami, une éruption volcanique ou un tremblement de terre par exemple. Les questions d'évacuation et les facteurs qui les influencent sont très étudiées, particulièrement en science sociale [Wolshon et McArdle 2009, MT et Wolshon 2013]. Durant un tremblement de terre, la localisation

Real Map	WOK	PKP	KPP	PPE	CS	PDLAIS
Duration (min)	1.82	1.77	1.60	1.50	1.49	1.64
Duration SD (min)	10.21	9.33	8.09	12.73	6.69	8.35
Duration Max (min)	10.11	9.43	12.22	15.88	6.5	9.43
Speed (km/h)	33.63	33.39	34.71	34.24	35.96	34.84
Waiting Time (s)	28.98	24.24	20.14	23.12	16.4	20.73
Waiting Time SD (s)	51.96	38.43	38.12	47.12	27.14	37.9
Waiting Time Max (s)	439	371	531	471	268	417
Route Length (km)	0.857	0.862	0.825	0.776	0.823	0.853
Route Length SD(km)	12.56	13.51	11.99	19.18	12.3	12.18

TABLE 8 – Comparaisons de modèles de prise de décision pour le guidage en ville - scénario de terrorisme

des cassures sur les routes ne peut être prédite à l'avance, cependant les auteurs de l'article [Wisetjindawat et al. 2015] proposent un modèle probabiliste sur ces cassures, selon l'intensité du tremblement de terre. Nous utiliserons ce modèle pour simuler un état de catastrophe sur la zone urbaine de Cologne. La probabilité qu'une section de route r du réseau soit passable est défini par :

$$p_r = e^{-\lambda \cdot l_r}$$

p_r est la probabilité que la route r soit praticable par un véhicule après le tremblement de terre. λ est la constante représentant le taux de cassure de la section de route, selon l'intensité du tremblement de terre (locations/km) et l_r est la longueur (km) de la section de route r . La valeur de λ est obtenue à partir du rapport d'un bureau au Japon gérant les catastrophes [oj 2013]. Une intensité de 4 sur l'échelle de l'agence météorologique et sismique du japon (JMA) ne cause pas de cassure, une intensité de 5 JMA peut causer entre 0.035 et 0.11 cassures par kilomètre, 6 JMA en cause autour de 0.16 et enfin 7 JMA en cause 0.48.

Dans le scénario que nous proposons, les véhicules vont agir comme les fourmis, et propager l'information des liens disponibles ou inutilisables après la catastrophe. Grâce à ces informations un véhicule d'urgence essaiera de se frayer un chemin à travers la ville. Une intensité de 7 JMA (intensité maximale sur l'échelle) est une situation trop catastrophique pour être exploitée. Nous utilisons une valeur de 0.11 qui correspond à une intensité de 5 JMA. Nous appliquons ainsi la probabilité de cassure ci-dessus, sur le quartier de la ville de Cologne utilisé précédemment. Dans ce genre de situation, l'important est de pouvoir faire circuler un véhicule d'urgence. Nous évaluons donc l'impact en temps de parcours de ce véhicule traversant la zone urbaine sinistrée. Dans nos modèles, ce dernier obtient assez d'information, il sera amené à faire des détours pour arriver à destination. Mais il se peut qu'il n'obtienne pas assez d'information et reste coincé au milieu du trafic arrêté par les cassures. Étant donné que c'est une probabilité qui est appliquée, nous considérons plusieurs simulations du même

scénario. Enfin nous supposons que les feux de circulation ne sont pas disponibles après la catastrophe.

Simulation Iteration	1	2	3	4	5
PKP	no	no	135s	no	no
KPP	no	no	123s	no	no
PPE	110s	no	119s	115s	112s
PDLAIS	103s	no	101s	no	no

TABLE 9 – Comparaisons de modèles de prise de décision pour le guidage en ville - scénario de catastrophe naturelle

Le Tableau 9 montre si le véhicule d'urgence est arrivé à destination selon le modèle, en cas de tremblement de terre. S'il ne parvient pas à sa destination, la valeur *no* est enregistré. Nous remarquons que PKP et KPP ne sont pas performants, cela provient du calcul des k chemins jusqu'à la destination qui limitent les possibilités (ces k chemins sont longs et donc la possibilité de rencontrer des chemins impraticables est grande). Le modèle PDLAIS, plus performant, utilise de l'information locale, mais il est fortement dépendant de l'état de l'infrastructure et de la position des intersections communicantes par rapport au trajet de l'ambulance. Enfin, le modèle PPE, utilisant la phéromone localement, donne le plus de chance à l'ambulance d'arriver à destination. En effet, il permet une plus grande possibilité de chemin, choisi pas à pas par le véhicule, de manière locale. La solution locale PPE est la plus robuste dans ce cas précis d'événement catastrophique.

Troisième partie

EN RÉSUMÉ

CONCLUSIONS

J'ai présenté dans cette Habilitation à Diriger des Recherches treize ans de recherche que je vais synthétiser en trois points - en mettant en avant les contributions et surtout les leçons tirées.

MACHINES VIRTUELLES Nous avons créé Jooflux - un intergiciel permettant la mise à jour et l'insertion d'aspects pendant l'exécution d'applications. Nous avons également créé ConGolo - un langage de programmation permettant de gérer le dynamisme de l'environnement dans une application. L'élément commun de ces deux approches est de reposer sur une machine virtuelle permettant d'intégrer et d'optimiser les mécanismes de redirection dynamique d'appels de méthodes. Supporter ce dynamisme dans une application a un coût. Soit ce coût est très important en terme d'ingénierie logicielle pour implanter ces mécanismes à très bas niveau et présente de très bonnes performances ; soit ce coût est présent en terme de moindre performance mais s'effectue de manière transparente au travers de mécanismes fournis par un intergiciel. Nous avons opté pour cette dernière solution générique et pensons que l'évolution des plates-formes matérielles et le foisonnement de nouveaux langages de programmation nous a donné raison et que cette perte de performance nécessite encore du travail d'optimisation (*Scheduling and Just-in-time compilation*) mais devient de moins en moins significative.

ARCHITECTURES ORIENTÉES-SERVICES Nous avons créé AxSeL et ACOMMA pour optimiser le chargement et déchargement de fonctionnalités sous forme de services dans des environnements contraints. Nous avons intégré spontanément avec MySIM des services sur des critères de qualité de services. Les architectures orientées-services sont maintenant une technologie mûre, présentant de nombreux standards, et obtenant de bonnes performances. Ces architectures vont parfaitement pouvoir mettre en place une informatique de bordure (*Edge and Fog Computing*) en déployant des services dans de petits centres de données de proximité (*Cloudlet and Mobile Cloud Computing*). Ces architectures vont toutefois devoir évoluer si elles veulent pouvoir s'appliquer à l'Internet des Objets car elles reposent sur des mécanismes de découvertes de services ne gérant pas l'incertitude. Et typiquement avec des propriétés de passage à l'échelle, de connexions / déconnexions, et potentiellement de mauvais fonctionnement de capteurs, les architectures orientées-service vont devoir s'orienter vers des solutions probabilistes.

INTERGICIELS LARGE-ÉCHELLE AUTONOMES Nous avons mis en place CANDS - une plate-forme permettant de gérer un très grand flux de données, garantissant des calculs optimaux tout en obtenant des temps de réponse très rapides. Nous l'avons également étendu pour prendre en compte le contenu des données et propager les messages selon une priorité de ces contenus (*Content-Centric Middleware*). Notre approche manipule de très grands graphes, avec de la dynamique sur

les poids des arêtes, mais pour l'instant pas encore avec de la dynamique structurelle sur ces graphes. Les études menées sur l'analyse de trace et les expériences dans la Ville Intelligente nous montrent que l'application considérée et son environnement revêtent une importance capitale. Par exemple, l'analyse d'une trace de mobilité peut très bien nous montrer que le comportement - même réel - n'est rien de plus qu'un graphe aléatoire. Ou par exemple, le guidage en ville par calcul de chemin dans un graphe peut être optimal mais très peu résilient à un accident ou une catastrophe naturelle. Ces résultats nous laissent à penser qu'une approche centrée-application (*Application-Centric Middleware*) serait préférable pour la mise en œuvre d'intergiciels large-échelle autonomes.

Les travaux présentés dans cette Habilitation se concentrent sur les aspects intergiciels, mais cette liste de travaux n'est pas exhaustive de toute la recherche menée sur cette période. La thématique de la mobilité a également pris une bonne place dans ma recherche avec la création d'une trace de micro-mobilité véhiculaire dans la thèse de MARIE-ANGE LÈBRE, et son exploitation pour la mise au point de rond-point autonome [Lèbre et al. 2015, Lèbre et al. 2015a, Lèbre et al. 2015b, Lèbre et al. 2015c, Lèbre et al. 2016], l'analyse de traces de mobilité existantes en collaboration avec PEDRO VAZ DE MELO, ALINE VIANA, MARCO FIORE et KATIA JAFFRÈS exhibant les relations sociales machine-à-machine [Vaz De Melo et al. 2012, Vaz De Melo et al. 2013, Vaz De Melo et al. 2015], le mélange de mobilité véhiculaire urbaine et de diffusion pair-à-pair pour le smart parking dans la thèse de TRISTA LIN, co-encadrée par HERVÉ RIVANO [Lin 2015][Lin et al. 2013, Lin et al. 2014, Lin et al. 2014a, Lin et al. 2014, Lin et al. 2014b, Lin et al. 2015, Lin et al. 2015], et enfin, avec SHIYOU QIAN et JIAN CAO, l'analyse de la mobilité des taxis de Shanghai pour proposer un système de recommandation équitable [Qian et al. 2015b]. Ces échanges d'idées furent passionnants - et le seront encore j'espère! Ces travaux ont grandement alimenté ma réflexion et m'ont conforté dans mon idée que les intergiciels large-échelle autonomes doivent caractériser chaque propriété du dynamisme au regard des applications.

*Most creativity is a transition from one context
into another where things are more surprising. [...]
Our job is to remind us that
there are more contexts
than the one that we're in
— the one that we think is reality.*

— Alan Key [Kay 2004]

8.1 PERSPECTIVES À COURT TERME

Mes perspectives de recherche à court terme sont celles que j'anime dans l'équipe de recherche DYNAMID - l'intégration de systèmes de systèmes (*Systems-of-Systems*) pour les environnements dynamiques. Mes deux principaux domaines d'application visés sont les Villes Intelligentes et l'Internet des Objets, toujours dans la suite des travaux précédents. Ces travaux portent vraiment leurs fruits actuellement avec le démarrage d'une chaire avec l'entreprise SPIE sur l'Internet des Objets au département TÉLÉCOMMUNICATIONS et une pleine participation de l'équipe DYNAMID.

Les trois points durs scientifiques que je vais développer sont autour des intergiciels et de l'Internet des Objets :

PROVISIONNEMENT AUTOMATIQUE ET RÉSILIENT DE SYSTÈMES IOT LARGE-ÉCHELLE Le développement de l'IoT est annoncé avec des milliards d'objets connectés [Gartner 2015]¹. Dans ce contexte, la configuration de ces objets ne pourra plus aucunement se faire manuellement à l'installation. Nous allons nous intéresser à la phase initiale de configuration (*bootstrap*) de manière et ce qu'un objet puisse démarrer avec le moins de prérequis d'installation, de manière à ce que tout ce dont il va avoir besoin en terme de logiciel - selon le rôle endossé - soit automatiquement installé (*initial provisioning*) et que toute erreur de fonctionnement ou mise à jour puisse permettre de manière sûre et sécurisée de pouvoir redémarrer quoi qu'il se passe (*dependable mechanisms*) [Abiteboul et al. 2007, Schonwalder et al. 2008].

SURVEILLANCE, DÉTECTION DE COMPORTEMENTS ET ADAPTATION DE DÉPLOIEMENT POUR SYSTÈMES IOT À LONGUE DURÉE DE VIE Le développement actuel de l'IoT, avec des déploiements large-échelle, soulève de nouveaux enjeux de gestion de ces systèmes sur le long terme - avec des durées de vie de dizaines d'année. Un déploiement IoT est en effet spécifique avec des contraintes de durée de vie de batterie, souvent localisé dans des endroits où l'intervention humaine est coûteuse voire impossible et pourtant doit pouvoir répondre à une adaptation aux avancées technologiques rapides. Nous

1. Les chiffres sont vraiment à nuancer car très différents selon les études, mais reflètent tout de même une très large-échelle.

allons nous intéresser plus particulièrement au cas d'installation d'une nouvelle machine dans une fabrique, avec tous les impacts que cela peut avoir sur une chaîne logistique (intervention de technicien de maintenance, réparations, mises à jour, etc.) pour automatiser le processus. Ce travail aura pour but d'analyser dynamiquement le comportement (*behavior checking*) pour adapter de manière autonome le déploiement, par exemple en prenant des décisions de groupes entre machines pour éviter des pannes (*local consensus / distributed decision-making*), en activant / désactivant des droits d'accès ou en changeant la production en fonction de qui se trouve à proximité de la machine [Li et al. 2012].

ÉVALUATION DE SÉCURITÉ D'UN SYSTÈME IOT Le développement rapide de l'IoT, avec du matériel hétérogène dans des environnements distribués et peu cloisonnés [Gubbi et al. 2013], pose de nombreuses questions quant à la sécurisation de ces systèmes. Les défis associés sont à l'intersection de problèmes de sécurité réseau, logiciel et systèmes distribués et nécessitent une approche globale [Riahi et al. 2013]. Nous allons nous intéresser plus particulièrement à un cas d'usage lors d'un audit de sécurité ponctuel (*auditing and practices*). Ce travail aura pour but de proposer un modèle (une représentation intermédiaire) des systèmes IoT étudiés, de modéliser de manière semi-automatique les infrastructures existantes et enfin d'analyser des propriétés de sécurité des matériels et logiciels sur ces réseaux afin d'assister l'auditeur à détecter des situations à risque.

Ces sujets font l'objet de démarrage de thèses, la première co-encadrée avec JULIEN PONGE dans le cadre d'une thèse CIFRE avec l'entreprise REDHAT; la seconde co-encadrée avec NICOLAS STOULS dans le cadre d'une thèse CIFRE avec l'entreprise ECONOCOM; la troisième co-encadrée avec FRANÇOIS LESUEUR dans le cadre d'une thèse CIFRE avec l'entreprise ALGOSECURE. Ces travaux vont avoir comme théâtre d'expérimentation commun une plate-forme IoT que je suis en train de mettre en œuvre au rez-de-chaussée du bâtiment CLAUDE CHAPPE du département TÉLÉCOMMUNICATIONS. Cette plate-forme permettra - j'espère dans un futur proche - d'inciter ces entreprises partenaires - conjointement avec la chaire IoT SPIE - à des accords de consortium pour des réponses à appels à projets.

8.2 PERSPECTIVES À LONG TERME

A plus long terme - après une évolution avant les années 2000 basée sur les intérêts économiques des domaines applicatifs et des évolutions technologiques, et - après une évolution après les années 2000 basée sur les intérêts sociétaux sans trop mesurer les enjeux des technologies applicatives créées, je me mets à espérer une recherche et développement intégrant des questionnements et des allers-retours entre les deux. En effet, de grandes évolutions sont annoncés - cf Figure 36 - et ont déjà leurs fondements scientifiques créés, comme l'informatique quantique (*Quantum Computing*) [Chow et al. 2014], les réseaux sur le corps humain (*Body-Area Network - BAN*) [Seyedi et al. 2013], les nano-technologies et les bio-technologies appliquées à l'être humain (*Human Enhancements, Neural Connectivity*) [Bostrom et Sandberg 2009, Kosmyna et al. 2016] ou à l'environnement (*Smart Dust*) [Qu et al. 2013, Ilyas et Mahgoub 2016]. Plus que les tech-

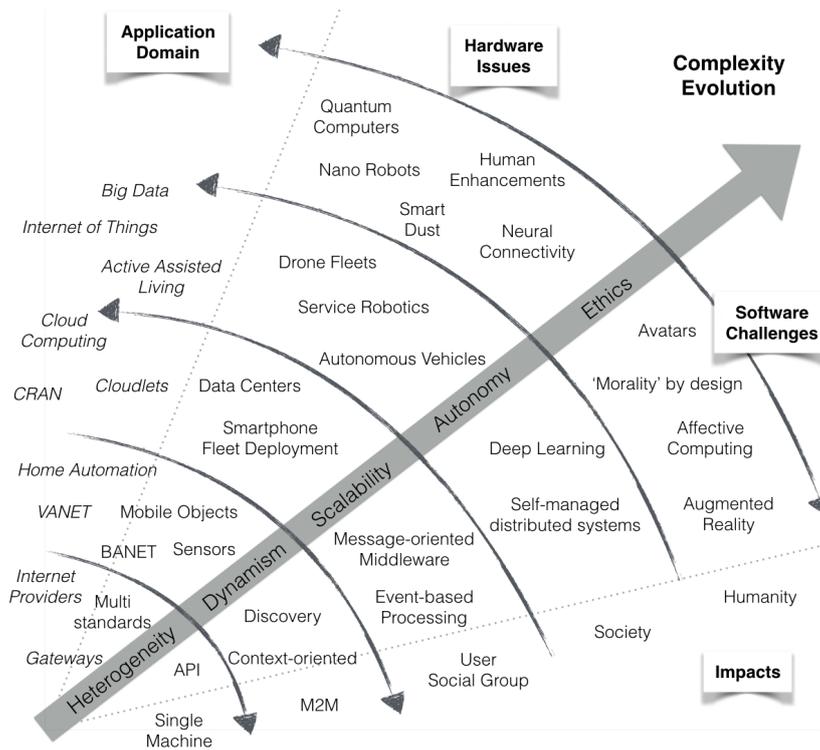


FIGURE 36 – Vers du logiciel éthique

nologies précédentes s'intégrant à notre quotidien, ces nouvelles technologies peuvent s'interconnecter avec l'être humain et l'environnement. Après l'Internet des Machines, l'Internet des Objets, le scénario de l'humain augmenté et d'un Internet d'Humains connectés (*Internet of People*) est techniquement une possibilité ne relevant plus de la science-fiction [Miranda et al. 2015].

En charge d'un transport intelligent des données, de traitement distribués sur ces données, les intergiciels prennent de plus en plus de décisions de manière autonome. Au-delà des questions de responsabilité et de gouvernance - déjà difficiles - se pose la question de l'éthique quand à ces décisions. En effet, bien que la définition de machine diffère, les nano- ou bio-technologies sont programmables et ce seront bien les 'logiciels' - et plus précisément les 'intergiciels' réutilisables et extensibles avec leurs algorithmes qui y auront été implantés qui prendront des décisions. Mais ces intergiciels agiront-ils bien ou mal sur l'être humain ?

J'entrevois trois pistes d'évolution de l'intelligence ambiante où les intergiciels vont jouer un rôle central :

LA RÉALITÉ AUGMENTÉE (AUGMENTED REALITY) Avec l'intégration de périphériques augmentant notre perception du monde - comme les lunettes ou casques à réalité augmentée (*Google Glass*), les montres connectés (*iWatch*), etc. - se pose la question de ce que va présenter l'intergiciel comme informations à l'utilisateur. L'intergiciel en a-t-il le droit ? Est-ce bien pour l'utilisateur ? Pour tous les utilisateurs ? Doit-il cacher des informations ? Où se place le curseur ?

L'exemple de Pokémon Go illustre assez bien ces propos [Hern 2016] : un logiciel en créant un monde virtuel affiche des animaux légendaires (*Pokémons*) à attraper virtuellement mais dans un lieu physiquement géo-localisé, et - d'un côté - amène les utilisateurs à une activité physique en marchant et à la découverte culturelle de nouveaux lieux, mais - d'un autre côté - peut provoquer des mouvements de foule dans des lieux publics ou des intrusions dans des propriétés privées.

L'INFORMATIQUE AFFECTIVE (AFFECTIVE COMPUTING) ET L'INFORMATIQUE PERSUASIVE (PERSUASIVE COMPUTING) L'interconnexion de périphériques directement dans notre corps affecte l'être humain physiquement, mais également psychologiquement. Beaucoup d'éléments logiciels influencent déjà notre quotidien, et prenons-nous bien la mesure des sentiments mis en jeu et de cette influence? [Fritz et al. 2014, Chen et al. 2015]

Un étude récente montre que cet aspect psychologique peut réellement changer notre comportement en bien, mais également avec des effets négatifs [Duus et Cooray 2015] : “une étude avec 200 femmes qui portaient un capteur d'activité Fitbit [... pour surveiller l'activité physique ...] a révélé que la plupart des utilisatrices ont adopté le dispositif comme s'il était parti d'elles-mêmes et ont cessé de le traiter comme une technologie externe. Presque toutes les participantes ont pris une route plus longue pour augmenter leur nombre de pas et la quantité d'exercice hebdomadaire réalisée. [...] La plupart des femmes de l'étude ont pensé qu'il était important de quantifier leurs activités quotidiennes et vérifier leur tableau de bord de progression plus de deux fois par jour. Mais [...] lorsque nous avons demandé aux femmes comment elles se sentaient sans leur Fitbit, beaucoup ont déclaré se sentir « nue » et que les activités qu'elles menaient, gaspillées. Certains se sentaient même moins motivés pour faire de l'exercice. Beaucoup se sont senties sous pression pour atteindre leurs objectifs quotidiens et ont ressenti que leurs routines quotidiennes étaient contrôlées par Fitbit.”

AVATAR, APPRENTISSAGE (MACHINE LEARNING) ET INTELLIGENCE ARTIFICIELLE (ARTIFICIAL INTELLIGENCE) L'évolution des intergiciels fournit une capacité de collecte et stockage de données et une capacité de calcul inégalées jusqu'alors. Ces capacités permettent à des domaines comme l'apprentissage automatisé (*Machine Learning* et *Deep Learning*) d'atteindre des profondeurs de raisonnement inégalées pour une machine jusqu'alors [Schmidhuber 2015, Goodfellow et al. 2015].

A la différence de DeepBlue d'IBM battant Kasparov - le champion d'échec - en 1996 grâce seulement à la profondeur de calcul [Newborn 2012], AlphaGo - logiciel de jeu de go développé par Google [Silver et al. 2016] - avec un apprentissage profond, a battu Fan Hui - champion européen - puis Lee Sedol - champion du monde - en apprenant progressivement de ses erreurs. Le point intéressant est que AlphaGo fut *surpris* par un coup - “J'ai jeté un œil rapide aux registres internes : AlphaGo estimait à moins de 1 sur 10000 la probabilité du brillant 78e coup de Lee, donc AlphaGo a trouvé celui-ci vraiment surprenant.” – Demis Hassabis [Audureau 2016]. Et l'autre point intéressant est qu'AlphaGo fit des coups *innovants* que les meilleurs joueurs ont considérés mauvais sur le moment mais ce sont révélés payants sur plus long terme “La vieille croyance à la créativité humaine et à nos

connaissances traditionnelles sur le go ont été mises à l'épreuve par mon expérience, et je me rends compte que j'ai du travail à faire dans cette direction." – Lee Sedol.

Maintenant peut-on confier une opération chirurgicale à un intergiciel s'il peut être surpris? Mais fera peut-être mieux qu'un humain? Et comment mesurer si mon intergiciel a été bien entraîné et raisonne 'suffisamment profondément'? Qui supervise le développement d'un intergiciel? Un intergiciel peut-il nous représenter par le biais d'un avatar? [Castelvecchi 2016]

Tout comme actuellement la notion de '*Privacy by design*' [Spiekermann 2012] commence à imprégner le développement de systèmes d'information, je pense qu'une future voie de recherche est de fournir des outils de conception des logiciels permettant de voir par qui et comment est programmé un intergiciel - le '*Morality by design*' [Verbeek 2011]. Et encore plus que la conception de logiciel [Bloom 2013, Bloom 2014], les intergiciels pouvant avoir des durées d'exécutions longues avec des comportements changeants - s'adaptant intelligemment à l'environnement, il serait intéressant d'avoir des outils qualifiant dynamiquement d'éventuels changements moraux – '*Moral Software Life-cycle & Ethical Machine Learning*'.

Quatrième partie

ANNEXES

Curriculum Vitæ

Frédéric LE MOUËL

41 years old, married, two children

Address :

DynaMid Team / INRIA CITI Laboratory

INSA Lyon

Bâtiment Claude Chappe

6 avenue des Arts

F-69621 Villeurbanne Cedex

Tel. : +33 472 436 485

Fax : +33 472 436 227

E-mail : frederic.le-mouel@insa-lyon.fr

Web : <http://www.le-mouel.net>

<http://perso.citi.insa-lyon.fr/flemouel>

<http://www.linkedin.com/in/fredericlemouel>

Current Position

- **Associate Professor** since September 1st 2004 in the TC Department (*Telecommunications, Services and Uses*) of INSA Lyon, France.
- **Researcher** in the INRIA CITI Laboratory (*Innovation Center in Telecommunications and Services Integration*) of INSA Lyon, France :
 - INRIA ARES Project (*Architectures of Networks of Services*) (2004-2008),
 - INRIA AMAZONES Project (*Ambient Architectures : Service-Oriented, Networked, Efficient, Secure*) (2008-2012),
 - and currently head of the DynaMid Project (*Dynamic Software and Distributed Systems for the Internet of Things*) (2012-) (team of 4 associate professors, 1 contractual professor, 1 ATER, 3 PhD students - <http://dynamid.citi-lab.fr>).
- Computer Science Section 27.
- Holder of the French Doctoral Supervision and Research Award (PEDR) (2007-2011)(2015-2019) and of the French Scientific Excellency Award (PES) (2011-2015).
- Head of the joint teaching SPE-T Program (*Specific Program of Engineering in Telecommunication*) between INSA Lyon, SJTU and EM Lyon in Shanghai, China (2013-2016).
- Founder of the double PhD degree between INSA Lyon and SJTU, Shanghai, China.

Former Positions

- 2013 - 2014 :** **Visiting Professor** in the CS Department (*Computer Science*) of SJTU (*Shanghai Jiao Tong University*). (National CRCT. Region mobility grant. Head - for the Rhône Alpes Region - of the joint research COOPERA Project and the joint teaching SPE-T Program. International collaboration with Professor Jian Cao. PhD supervisions), Shanghai, China.
- 2001 - 2003 :** **Assistant Professor** at the EMN (*École des Mines de Nantes*), member of the OCM Team (*Objects, Components, Models*) (Lectures on “*Systems and Networks Administration*”, “*Systems*”, “*Networks*” and “*Middlewares*”), France.
- 1999 - 2000 :** **Lecturer** at the IFSIC (*Institut de Formation Supérieure en Informatique et Communications*), University of Rennes 1 (Lectures on “*Object Programing*”, “*Software Engineering*”, “*Compilation*”, and “*Systems and Networks*”), France.
- 1998 - 2001 :** **Ph.D. Student (INRIA grant)**, member of the INRIA SOLIDOR Project (*Design of Distributed Operating Systems*) at the IRISA laboratory (*Institut de Recherche en Informatique et Systèmes Aléatoires*) and at the University of Rennes 1, France.
- 1996 - 1997 :** **Part-time Lecturer** at the IFSIC (*Institut de Formation Supérieure en Informatique et Communications*), University of Rennes 1 (Lectures on “*Algorithmic*” and “*Programing*”), France.

Education

- 1998 - 2003 :** **Ph.D. in Computer Science with honors**, IRISA Laboratory (*Institut de Recherche en Informatique et Systèmes Aléatoires*), University of Rennes 1, France.
Title : *Adaptive Environment for Distribution of Applications in a Mobile Context*.
Keywords : *Adaptive Systems, Mobile Computing, Distributed Computing*.
Defended, in December 1st 2003, in front of the committee :
President : D. HERMAN (Univ. of Rennes 1)
Reviewers : H. GUYENNET (Univ. of Franche Comté) and I. DEMEURE (ENST Paris)
Members : B. FOLLIOU (Univ. Pierre et Marie Curie) and M.-T. SEGARRA (ENST Bretagne)
Supervisor : F. ANDRÉ (Univ. of Rennes 1)
- 1994 - 1997 :** **Diplôme d’Ingénieur en Informatique et téléCommunication, cum laude**, French equivalence of a Master’s Degree in Computer Science and Telecommunications, IFSIC (*Institut de Formation Supérieure en Informatique et Communications*), University of Rennes 1, France.
Speciality : *Languages and Operating Systems*.
Title : *Smart Retrieval of Information on Internet and Intranet*.
Keywords : *Smart Agents, Information System, Document Indexation, Document Retrieval, Stochastic and Genetic Algorithms*.
Supervisor : E. COLAVITI (Alcatel Alsthom Research, Marcoussis).

Teaching Activities

- Master lecture, together with Julien Ponge and Yves Caniou, on “*Distributed Computing : From Principles to Pervasive, Grid and Cloud Computing*” at the University of Lyon, France (2012-2016).
- Master lectures, together with Stéphane Frénot, on “*Pervasive Systems and Peer-to-Peer Computing*” at the University of Lyon, France (2006-2011).
- Master lectures, together with Frédérique Laforest, on “*Pervasive Information Systems*” at the University of Jendouba, Tunisia (2004-2007), at the University of Kairouan, Tunisia (2006-2007) and at the Benemérita Universidad Autónoma de Puebla, Mexico (2005-2006).
- 5th INSA Year, lectures on “*Cloud and Internet of Things*” (leader, 2016-), “*The Internet of Things : the application view - The Ambient Intelligence*” (leader, 2010-2016) and “*Pervasive Systems*” (leader, 2005-2011) in the TC Department, INSA Lyon, France.
- 4th INSA Year, lectures on “*Modeling and Software Engineering*” (2005-2010), “*Modeling and Object Programming*” (2004-2005) and “*Middleware*” (2004-2016) in the TC Department, INSA Lyon, France.
- 3th INSA Year, lectures on “*Dynamic Web*” (leader, 2004-2016), “*Java*” (co-leader, 2004-2012) in the TC Department, INSA Lyon, France.

All these lectures represents ~275h eq TD per year.

Supervisions

Name	Degree	University	Years	Subject
Marie-Ange Lèbre	Ph.D	INSA Lyon (Valeo CIFRE grant)	2012 2016	Impact of a Local and Autonomous Decision on Intelligent Transportation Systems at different Scales.
Roya Golchay	Ph.D.	INSA Lyon (Iranian CIFRE grant)	2011 2016	From Mobile to Cloud : Using Bio-inspired Algorithms for Collaborative Offloading.
Trista Lin	Ph.D	INSA Lyon (Region grant)	2012 2015	Smart Parking : Network, Infrastructure and Urban Service.
Amira Ben Hamida	Ph.D.	INSA Lyon & ENSI (Tunisia) (Amigo grant)	2006 2010	A Middleware for a Contextual and Autonomous Deployment of Services in Pervasive Environments.
Noha Ibrahim	Ph.D.	INSA Lyon (Libanian grant & Amigo grant)	2004 2008	Spontaneous Integration of Services in Pervasive Environments.

Ferit Bayer Elhuseyni	Master	INSA Lyon & University of Lyon	2011	A Cross-layer Approach to Achieve Efficient Service Delivery in Mobile Delay-Tolerant Networks with a Service-Oriented Network Coding.
Etienne Gili	Master	INSA Lyon & University of Lyon	2009	Modelisation of a Spontaneous Orchestration of Services.
Ali Tounetki	Master	INSA Lyon & University of Lyon	2009	Ambient Aggregation of Service Repositories.
Cédric Lévy-Bencheton	Master	University of Lyon	2007	Trust Model integrating Services' Semantics.
Ines Barrak	Master	INSA Lyon & University of Kairouan (Tunisia)	2007	A Dynamic Colouring Algorithm for a Constrained-based Service Loading.
Mouna Khlifi	Master	INSA Lyon & University of Kairouan (Tunisia)	2007	Towards a Context-aware Orchestration of Web Services.
Hajer Chamekh	Master	INSA Lyon & University of Jendouba (Tunisia)	2006	Semantic Deployment of Services in a Pervasive Environment.
Máté Soós	Master	INSA Lyon & University of Budapest	2005	Trust Model Simulation for Ambient Networks and a Protocol to Exchange Credentials between Peers.
Nabil Benanbar	Master	University of Lyon	2005	Smart Strategies for Using Pervasive Services.
Amira Ben Hamida	Master	INSA Lyon & University of Jendouba (Tunisia)	2005	Spontaneous Proximity Documents.
Michael Ion & Javier Lallana	IST Semester	INSA Lyon	2014	Automatic Discovery Protocols Use in Context-Oriented Programming for the Internet of Things.
Michael Genkin	IST Semester	INSA Lyon	2012	Control and Management Interface for Java 7 JooFlux Agent
Wei Liang Choo	IST Semester	INSA Lyon	2010	Studying network coding behavior in mobile social networks - WSN implementation.

+ 20 PFE : End of study internships from 2005 to 2016.

Research Activities

Committees

Journals

– Program Committee of the International Journal On Advances in Internet Technology, 2012.

- Co-editor, with *Frédérique Laforest*, of the *ISI journal special edition on “Adaptation and Context Management”*, *Revue d’Ingénierie des Systèmes d’Information (ISI)*, vol. 11, no.5, 2006.

Conferences

- *Program Committee of the International Conference on Mobile Systems and Pervasive Computing (MobiSPC’2016)*.
- *Program Committee of the International Conference on Intelligent Systems and Applications (INTELLI’2012-2016)*.
- *Program Committee of the International Conference on Systems (ICONS’2012-2016)*.
- *Program Committee of the International Conference on Pervasive and Embedded Computing (PECCS’2011-2013)*.
- *Program Committee of the International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM’2008-2016)*.
- *Program Committee of the 11th International Conference on New Technologies in Distributed Systems / NOuvelles TEchnologies de la REpartition (NOTERE’2012)*.
- *Program Committee of the 8^{ème} journées francophones Mobilité et Ubiquité (UbiMob’2012)*.
- *Program Committee of the 6th-8th International Conference on Ubiquitous Intelligence and Computing (UIC’2009-2011)*.
- *Organization Committee of European Conference on Computer Systems (EuroSys’2010)*.
- *Program Committee of the 11th IEEE International Conference on High Performance Computing and Communications (HPCC’2009), Pervasive/ubiquitous computing and intelligence Track*.
- *Program Committee of the International Conference on Mobile Communications and Pervasive Computing (MCPC’2009)*.
- *Program Committee of the 14th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA’2008), Ubiquitous Computing Track*.
- *Program Committee of the International Conference on Pervasive Services (ICPS’2008)*.
- *Program Committee of the International Symposium on Ubiquitous Computing Systems (UCS’2007)*.
- *Organization Committee of the Objects, Components, Models Conference (OCM’2002) (including website creation and proceedings edition, <http://www.emn.fr/x-info/ocm-2002/>)*.

- *Organization Committee of the 1st French Conference on Operating Systems (CFSE'1)* (including website creations of Rennes'99 and CFSE'1, <http://ren99.irisa.fr/> and <http://www.irisa.fr/manifestations/1999/cfse1/>).

Workshops

- *Program Committee of the On The Move Academy Workshop (OTM Academy'2009-2016)* in conjunction with OTM Federated Conferences.
- *Program Committee of the International Symposium of Internet of Ubiquitous and Pervasive Things (IUPT'2011-2013)* in conjunction with Ambient Systems, Networks and Technologies Conferences (ANT'2011-2013).
- *Chairman of the 9^{ème} atelier sur l'Évolution, la Réutilisation et la Traçabilité des Systèmes d'Information (ERTSI'2010)* (including website creation, <http://ertsi2010.conf.citi-lab.fr/>).
- *Chairman of the 1st-4th IEEE/ACM International Workshop on Services Integration in Pervasive Environments (SIPE'2006-2009)* in conjunction with the *IEEE/ACM International Conference on Pervasive Services (ICPS'2006-2009)* (including website creation and proceedings edition, <http://sipe09.conf.citi.insa-lyon.fr/>).
- *Program Committee of the 1st-3rd Workshop on Middleware-Application Interaction (MAI'2007-2009)* in conjunction with EuroSys'2007 and DisCoTec federated conferences.
- *Organization Committee of the Colloque PRIAM : Les technologies au service des droits : opportunités, défis, limites (PRIAM'2008)* (including website creation, <http://priam08.conf.citi.insa-lyon.fr/>).
- *Program Committee of the 19th IFIP/IEEE International Workshop on Distributed Systems : Operations and Management (DSOM'2008)* part of MANWEEK'2008.
- *Program Committee of the 2nd IEEE WoWMoM Workshop on Adaptive and Dependable Mission-critical Mobile Systems (ADAMUS'2008)* in conjunction with WoWMoM'2008.
- *Program Committee of the 6^{ème} Atelier sur les Objets, Composants et Modèles dans l'ingénierie des Systèmes d'Information (OCM-SI'2007)* and of *7th Atelier sur l'Evolution, la Réutilisation et la Traçabilité des Systèmes d'Information (ERTSI'2008)*, in conjunction with INFORSID'2007-2008.
- *Program Committee of the International Workshop on Self-Adaptive Software Engineering (SASE'2008)* in conjunction with IEEE CSMR'2008.
- *Organization Committee of the 1st Workshop on Software Organisation and Monitoring of Ambient Systems (SOMITAS'2008)*, in conjunction with Ambi-sys'2008.
- *Program Committee of the International Workshop on Functional Architecture for Assisted Living Services (FunCAAL'2007)*, in conjunction with Pervasive'2007.

- *Program Committee of the French OSGi Workshop*, in conjunction with UbiMob'2006.

Referees

- *Referees for the journal : Ad hoc Networks (2012).*
- *Referees for the conferences : International Conference on Service Oriented Computing (IC-SOC'2011), International Conference on Software Composition (SC'2009), International Conference on Embedded Computer Systems : Architectures, Modeling and Simulation (IC-SAMOS'2009), NOuvelles TEchnologies de la REpartition (NOTERE'2009, NOTERE'2005), Mediterranean Ad Hoc Networking (Med-hoc-net'2005), Symposium on Applied Computing (SAC'2001).*

PhD Jury

- *Member (Supervisor) of the PhD Jury of Ms. Roya Golchay, "From Mobile to Cloud : Using Bio-inspired Algorithms for Collaborative Offloading", University of Lyon, INSA Lyon, France, January 26th 2016.*
- *Member (Supervisor) of the PhD Jury of Ms. Marie-Ange Lèbre, "Impact of a Local and Autonomous Decision on Intelligent Transportation Systems at different Scales", University of Lyon, INSA Lyon, France, January 25th 2016.*
- *Member (Supervisor) of the PhD Jury of Ms. Trista Lin, "Smart Parking : Network, Infrastructure and Urban Service", INSA Lyon, France, December 17th 2015.*
- *Member of PhD Jury of Mr. Gregory Van Seghbroeck, "Design-Time Validation and Execution of Service Choreographies", University of Ghent, Belgium, June 21th 2011.*
- *Member of the PhD Jury of Mr. Aitor Urbieta, expert for the "Doctor Europeus" PhD mention, "An integrated approach for a context-aware and 'cause-effect'-oriented modelling, matching and composition of semantic services for intelligent environments", Mondragon Goi Eskola Politeknikoa University, Spain, June 2010.*
- *Member (Supervisor) of the PhD Jury of Ms. Amira Ben Hamida, "AxSeL : A Middleware for a Contextual and Autonomic Deployment of Services in Pervasive Environments", INSA Lyon, France, February 1st 2010.*
- *Member (Supervisor) of the PhD Jury of Ms. Noha Ibrahim, "Spontaneous Integration of Services in Pervasive Environments", INSA Lyon, France, September 23th 2008.*
- *Member of PhD Jury of Mr. Frederic Van Quickenborne, "Design and Efficient Management of Aggregation Networks for Fast Moving Users", University of Ghent, Belgium, November 22nd 2007.*

Contracts

- Leader of a regional project for structuring collaborations with SJTU in Shanghai, China - *Smart Cities and IT Technologies for a Sustainable Development* - also supported with a mobility grant [COOPERA+Explora'Pro], 2013-2015 (15+25 k€).
- Leader of a bilateral contract with Valeo - *Service Deployment and Invocation under Delay Constraints in Vehicular Networks* [Type INSAVALOR+CIFRE], 2012-2015 (110 k€).
- Co-leader, with Hervé Rivano, of a regional project in collaboration with Lyon Urban Agency (UrbaLyon) - *Mesuring the Urban Mobility for a Citizenship Service Cartography* [Type ADR-ARC 7], 2012-2015 (32 k€).
- Leader for INSA Lyon of the Smart Chappe Building Project - *The Smart Chappe Building : A Context-aware Service Platform* [Type BQF], 2009-2011 (20 k€). (<http://telecom.insa-lyon.fr/content/plateforme-smart-chappe-building>).
- Leader for INRIA Rhône-Alpes of the European Project Amigo - *Ambient Intelligence for the Networked Home Environment* [Type IP]. Participation to Work Packages 2, 3, 6 and 9. Leader for the 3.4 task “Programming and Deployment Framework for Amigo Services”, 2004-2008 (220 k€) (<http://www.hitech-projects.com/euprojects/amigo/>).
- Member of the INRIA Cooperating Research Action PRIAM (*Privacy Issues and Ambient intelligence*) [Type ARC]. Leader for the information management system task, 2007-2009 (<http://priam.citi.insa-lyon.fr/>).
- Member of the French Ministerial Project KAA (*Knowledge Authentication Ambient - Multi Disciplinary Project for Trust Management Scheme in Spontaneous Networks*) [Type ACI], 2004-2007 (<http://kaa.citi.insa-lyon.fr/>).
- Member of the French Ministerial Project Arcad (*Architecture Répartie extensible pour Composants ADaptables - Distributed Architecture Extensible to Adaptive Components*) [Type RNTL], 2001-2003 (<http://arcad.essi.fr/arcad/>).

Patents

- J. Ponge, F. Le Mouël, “JooFlux”, Déclaration d’invention - Software Patent, CITI Lab, INSA Lyon, September 2012.

Scientific Animation

- Participation to the Zhang Heng program about “Smart Cities and Internet of Things” (**French Embassy**) : Visit of Chinese Academy of Sciences, National Natural Science Foundation of China, Beihang University, Shanghai Jiao Tong University (2012).
- Co-leader, with Karine Samuel, of the axe 2 “Mobilités et flux, connaître et agir ensemble : pour des organisations innovantes - Mobility, Flows, Knowledge : act together for innova-

ting organizations” of the Regional Academic Research Community Program “Innovations, Mobilités, Territoires et Dynamiques Urbaines - Innovations, Mobility, Territories and Urban Dynamism” (ARC 7) (2011-2015).

- Co-leader, with Cécile Favre and Jean-Paul Jamont, of the Summer School about “Internet and Web of Things” of the Regional Academic Research Community Program “Technologies de l’Information et de la Communication et Usages Informatiques Innovants - Technologies for New and Innovative Information, Communication and Use” (ARC 6) (2013).
- Scientific Seminars Manager of the CITI Laboratory (2010-2013) (<http://www.citi-lab.fr/news/>).
- Leader for INSA Lyon of the Apple on Campus and iOS Developer University Program (2009-2013).

Seminaries (out of conferences presentations)

- Les objets connectés : au service de qui ?
28^{ème} Entretiens Jacques Cartier, Showroom Smart Electric, EDF, Lyon, France, December 2nd 2015.
- Objet, composant, service : quel model pour les environnements pervasifs ?
Invited Talk, 6^{ème} Atelier sur les Objets, Composants et Modèles dans l’ingénierie des Systèmes d’Information (OCM-SI’2007), Perros-Guirec, France, May 22th 2007.
- Ambient Bundle Repository (ABR) : une plateforme OSGi contextuelle et sémantique pour le déploiement de services en environnement ambiant
Laboratory Seminary, LIUPPA, Bayonne, France, April 26th 2007.
- Deployment, Integration and Management of Services in Pervasive Environments
Invited Talk, 1st Workshop China-France on Telecommunications, Xi’An, China, October 24th 2006.
- Le futur de l’Internet
Round Table, 9^{ème} Rencontres Internet d’Autrans, Autrans, France, January 14th 2005.
- Systèmes adaptatifs : étude, classification, solution pour le cas de la distribution en environnements mobiles
SimCITI Laboratory Seminary, INSA, Lyon, France, November 18th 2004.
- Adaptive Environment for Distribution of Applications in a Mobile Context
Invited Talk, Facultad de Ciencias de la Computación, Benemérita Universidad Autónoma de Puebla, Puebla, Mexico, December 19th 2003.
- AeDEn : un cadre général pour une distribution adaptative des applications en environnements mobiles

Journée de travail du projet RNTL ARCAD, École des Mines de Nantes, Nantes, France, September 13th 2001.

- Environnement adaptatif d'exécution distribuée d'applications dans un contexte mobile
Journée Parallélisme, Répartition et Objets (PRO), Université des Sciences et Technologies de Lille, Lille, France, November 25th 1999.

Open Source Software / Open Data

- Microscopic vehicular mobility trace of Europarc roundabout, Creteil (Creative Commons Licence) (2015) (<http://vehicular-mobility-trace.github.io>).
- Shanghai Taxi Trace - Contact Analysis (Apache Licence) (2014) (<https://github.com/flemouel/taxi-trace>).
- Golo - Fondation Eclipse (Eclipse Public Licence) (2012) (<https://github.com/eclipse/golo-lang>), and ConGolo (2015) (<https://github.com/bmaingret/contextual-golo-lang>).
- Jooflux (Software Patent - Mozilla Licence) (2012) (<https://github.com/dynamid/jooflux>).

Administrative Responsibilities

- Administrative Council of the Regional Academic Research Community Program “Innovations, Mobilités, Territoires et Dynamiques Urbaines - Innovations, Mobility, Territories and Urban Dynamism” (ARC 7) (2011-2015).
- Elected member of the TC Department Council (2005-2012).
- Elected member of the CITI Laboratory Council (2009-2012).
- Elected expert for faculty hiring committee (CS) (Computer Science Section 27) (2009-2013).
- Industrial Training Officier for students in 3rd year of the TC Department (2004-2013).
- Social Media Director of the CITI Laboratory (2015-2016).
- Creation of TC Department promotional leaflet, INSA, Lyon, France, 2004.

Memberships

- EuroSys Member (*European ACM-SIGOPS Association, Special Interest Group on Operating Systems*) since its creation in May 2005.
- ASF Member (*French ACM-SIGOPS Association*) since 1999. ASF Board (2008-2012), information system and research days (2008-2011).

- ESRGroups Member (*Open Community of Engineers and Researchers*) since 2006.
- OSGi French User Group Member since its creation in December 2005.
- Vice-treasurer of ADOC Association (*Association of Ph.D. Students of the University of Rennes I*), Rennes, France (1998-2000).

PUBLICATIONS

B.1 LIVRES ET CHAPITRES DE LIVRE

[Frénot et al. 2012] S. FRÉNOT, F. LE MOUËL, J. PONGE et G. SALAGNAC, *Various Extensions for the Ambient OSGi Framework, Computer Engineering : Concepts, Methodologies, Tools and Applications*, chapter 712, p. 1799–1810, *Computer Engineering : Concepts, Methodologies, Tools and Applications*, IGI Global, mai 2012.

[Ibrahim et al. 2010] N. IBRAHIM, F. LE MOUËL et S. FRÉNOT, *Middleware Technologies for Ubiquitous Computing, Handbook of Research on Next Generation Networks and Ubiquitous Computing*, chapter 12, p. 122–131, *Handbook of Research on Next Generation Networks and Ubiquitous Computing*, IGI Global, 2010.

B.2 EDITIONS DE LIVRES OU ACTES

[Laforest et Le Mouël 2006] F. LAFOREST et F. LE MOUËL (éd.), *Adaptation et gestion du contexte, Networking and Information Systems / Ingénierie des Systèmes d’Information (ISI)*, vol. 11, Hermès, octobre 2006, Special edition.

[Le Mouël et Frénot 2006] F. LE MOUËL et S. FRÉNOT (éd.), *Proceedings of the 1st IEEE International Workshop on Services Integration in Pervasive Environments (SIPE’2006)*, Lyon, France, IEEE Press, juin 2006.

[Le Mouël et Frénot 2007] F. LE MOUËL et S. FRÉNOT (éd.), *Proceedings of the 2nd IEEE International Workshop on Services Integration in Pervasive Environments (SIPE’2007)*, Istanbul, Turkey, IEEE Press, juillet 2007.

[Le Mouël et Frénot 2008] F. LE MOUËL et S. FRÉNOT (éd.), *Proceedings of the 3rd ACM International Workshop on Services Integration in Pervasive Environments (SIPE’2008)*, Sorrento, Italy, ACM Press, juillet 2008.

[Le Mouël et Frénot 2009] F. LE MOUËL et S. FRÉNOT (éd.), *Proceedings of the 4th ACM International Workshop on Services Integration in Pervasive Environments (SIPE’2009)*, London, UK, ACM Press, juillet 2009.

B.3 BREVETS

[Ponge et Le Mouël 2012] J. PONGE et F. LE MOUËL, *Jooflux*, Software Patent / Déclaration d’invention, France, septembre 2012.

B.4 JOURNAUX INTERNATIONAUX

[Frénot et al. 2011] S. FRÉNOT, F. LE MOUËL, J. PONGE et G. SALAGNAC, « Various Extensions for the Ambient OSGi Framework », *Interna-*

tional Journal of Adaptive, Resilient and Autonomic Systems (IJARAS), vol. 2, n°3, juillet 2011, p. 1–12.

- [Ibrahim et al. 2014] N. IBRAHIM, F. LE MOUËL et S. FRÉNOT, « Semantic Service Substitution in Pervasive Environments », *International Journal of Services, Economics and Management (IJSEM)*, vol. 6, n°4, 2014, p. 283–309.
- [Ibrahim et Le Mouël 2009] N. IBRAHIM et F. LE MOUËL, « A Survey on Service Composition Middleware in Pervasive Environments », *International Journal of Computer Science Issues (IJCSI)*, vol. 1, août 2009, p. 1–12, Special Issue on Pervasive Computing.
- [Le Mouël et al. 2002] F. LE MOUËL, F. ANDRÉ et M.T. SEGARRA, « AeDÉn : An Adaptive Framework for Dynamic Distribution over Mobile Environments », *Annals of Telecommunications*, vol. 57, n°11–12, novembre 2002, p. 1124–1148.
- [Lin et al. 2015] T. LIN, H. RIVANO et F. LE MOUËL, « Urban Infrastructure Deployment for Wireless On-Street Parking Sensor Networks », *Procedia Engineering*, vol. 115, août 2015, p. 29–36.
- [Vaz De Melo et al. 2015] P.O.S. VAZ DE MELO, A. C. VIANA, M. FIORE, K. JAFFRÈS-RUNSER, F. LE MOUËL, A. A. F. LOUREIRO, L. ADDEPALLI et G. CHEN, « RECAST : Telling Apart Social and Random Relationships in Dynamic Networks », *Performance Evaluation*, vol. 87, n°0, mai 2015, p. 19–36, Special Issue : Recent Advances in Modeling and Performance Evaluation in Wireless and Mobile Systems.
- [Yang et al. 2014] D. YANG, D. ZHANG, K.-L. TAN, J. CAO et F. LE MOUËL, « CANDS : Continuous Optimal Navigation via Distributed Stream Processing », *PVLDB*, vol. 8, n°2, 2014, p. 137–148.

B.5 JOURNAUX NATIONAUX

- [Ben Hamida et al. 2008] A. BEN HAMIDA, F. LE MOUËL, S. FRÉNOT et M. BEN AHMED, « Une approche pour un chargement contextuel de services dans les environnements pervasifs », *Networking and Information Systems / Ingénierie des Systèmes d'Information (ISI)*, vol. 13, n°3, juin 2008, p. 59–82, Special edition.
- [Ben Hamida et al. 2011] A. BEN HAMIDA, F. LE MOUËL, S. FRÉNOT et M. BEN AHMED, « Déploiement adaptatif d'applications orientées services sur environnements contraints », *Computer Science and Technology / Technique et Science Informatiques (TSI)*, vol. 30, n°1, 2011, p. 59–91.
- [Le Mouël et André 2001] F. LE MOUËL et F. ANDRÉ, « AeDÉn : un cadre général pour une distribution adaptative et dynamique des applications en environnements mobiles », *Electronic Journal on Networks and Distributed Processing / Revue Électronique sur les Réseaux et l'Informatique Répartie (EJNDP/RERIR)*, vol. 11, mars 2001, p. 169–181.

B.6 CONFÉRENCES INTERNATIONALES

- [André et al. 2000] F. ANDRÉ, A.-M. KERMARREC et F. LE MOUËL, « Improvement of the QoS via an Adaptive and Dynamic Distribution of Applications in a Mobile Environment », *Proceedings of the 19th IEEE Symposium on Reliable Distributed Systems (SRDS'2000)*, p. 21–29, Nürnberg, Germany, octobre 2000.
- [Ben Hamida et al. 2008a] A. BEN HAMIDA, F. LE MOUËL, S. FRÉNOT et M. BEN AHMED, « Contextual Service Loading by Dependency Graph Colouring », *Proceedings of the 8th International Conference on New Technologies in Distributed Systems (NOTERE'2008)*, p. 182–187, Lyon, France, juin 2008.
- [Ben Hamida et al. 2008b] A. BEN HAMIDA, F. LE MOUËL, S. FRÉNOT et M. BEN AHMED, « A Graph-based Approach for Contextual Service Loading in Pervasive Environments », *Proceedings of the 10th International Symposium on Distributed Objects and Applications (DOA'2008), Lecture Notes in Computer Science*, vol. 5331, p. 589–606, Springer Verlag, Monterrey, Mexico, novembre 2008.
- [Ben Hamida et Le Mouël 2006] A. BEN HAMIDA et F. LE MOUËL, « Resurrection : A Platform for Spontaneously Generating and Managing Proximity Documents », *Proceedings of the IEEE International Conference on Pervasive Services (ICPS'2006)*, p. 333–336, Lyon, France, juin 2006.
- [Frénot et al. 2010] S. FRÉNOT, N. IBRAHIM, F. LE MOUËL, A. BEN HAMIDA, J. PONGE, M. CHANTREL et D. BERAS, « ROCS : a Remotely Provisioned OSGi Framework for Ambient Systems », *Proceedings of the 12th IEEE/IFIP Network Operations and Management Symposium (NOMS'2010)*, p. 503–510, Osaka, Japan, avril 2010.
- [Golchay et al. 2016a] R. GOLCHAY, F. LE MOUËL, J. PONGE et N. STOULS, « Automated Application Offloading through Ant-inspired Decision-Making », *Proceedings of the 13th International Conference on New Technologies in Distributed Systems (NOTERE'2016)*, Paris, France, juillet 2016.
- [Golchay et al. 2016b] R. GOLCHAY, F. LE MOUËL, J. PONGE et N. STOULS, « Spontaneous Proximity Clouds : Making Mobile Devices to Collaborate for Resource and Data Sharing », *Proceedings of the 12th EAI International Conference on Collaborative Computing : Networking, Applications and Worksharing (CollaborateCom'2016)*, Beijing, China, novembre 2016, Best Paper.
- [Ibrahim et al. 2007a] N. IBRAHIM, F. LE MOUËL et S. FRÉNOT, « Automatic Service-Integration Framework for Ubiquitous Environments », *Proceedings of the International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM'2007)*, p. 15–20, Papeete, French Polynesia (Tahiti), France, novembre 2007.
- [Ibrahim et al. 2007b] N. IBRAHIM, F. LE MOUËL et S. FRÉNOT, « C-ANIS : A Contextual, Automatic and Dynamic Service-Oriented Integration Framework », *Proceedings of the International Symposium on Ubiquitous Computing Systems (UCS'2007), Lecture Notes in Computer Science*, vol. 4836, p. 118–133, Springer Verlag, Tokyo, Japan, novembre 2007.

- [Ibrahim et al. 2009] N. IBRAHIM, F. LE MOUËL et S. FRÉNOT, « MySIM : A Spontaneous Integration Middleware for Pervasive Environments », *Proceedings of the ACM International Conference on Pervasive Services (ICPS'2009)*, p. 1–10, London, UK, juillet 2009.
- [Ibrahim et al. 2010] N. IBRAHIM, S. FRÉNOT et F. LE MOUËL, « User-Excentric Service Composition in Pervasive Environment », *Proceedings of the 24th IEEE International Conference on Advanced Information Networking and Applications (AINA'2010)*, p. 682–689, Perth, Australia, avril 2010.
- [Ibrahim et Le Mouël 2006] N. IBRAHIM et F. LE MOUËL, « ANIS : A Negotiated Integration of Services in Distributed Environments », *Proceedings of the 8th International Symposium on Distributed Objects and Applications (DOA'2006), Lecture Notes in Computer Science*, vol. 4276, p. 1467–1484, Springer Verlag, Montpellier, France, octobre 2006.
- [Jouve et al. 2007] W. JOUVE, N. IBRAHIM, L. RÉVEILLÈRE, F. LE MOUËL et C. CONSEL, « Building Home Monitoring Applications : From Design to Implementation into The Amigo Middleware », *Proceedings of the 2nd International Conference on Pervasive Computing and Applications (ICPCA'2007)*, Birmingham, UK, juillet 2007.
- [Le Mouël et al. 2000] F. LE MOUËL, M.T. SEGARRA et F. ANDRÉ, « Improving Mobile Computing Performance by Using an Adaptive Distribution Framework », *Proceedings of 7th International Conference on High Performance Computing (HiPC'2000), Lecture Notes in Computer Science*, vol. 1970, p. 479–488, Springer Verlag, Bangalore, India, décembre 2000.
- [Le Mouël et André 2000a] F. LE MOUËL et F. ANDRÉ, « AeDÉn : un cadre général pour une distribution adaptative et dynamique des applications en environnements mobiles », *Proceedings of the 3rd International Conference on New Technologies in Distributed Systems (NOTERE'2000)*, p. 171–182, Paris, France, novembre 2000.
- [Le Mouël et André 2000b] F. LE MOUËL et F. ANDRÉ, « Distribution over Mobile Environments », *Proceedings of 2000 ACM Symposium on Applied Computing (SAC'2000)*, p. 568–569, Villa Olmo, Como, Italy, mars 2000.
- [Lèbre et al. 2015a] M.-A. LÈBRE, F. LE MOUËL et E. MÉNARD, « On the Importance of Real Data for Microscopic Urban Vehicular Mobility Trace », *Proceedings of the 14th International Conference on ITS Telecommunications (ITST'2015)*, p. 22–26, Copenhagen, Denmark, décembre 2015.
- [Lèbre et al. 2015b] M.-A. LÈBRE, F. LE MOUËL et E. MÉNARD, « Partial and Local Knowledge for Global Efficiency of Urban Vehicular Traffic », *Proceedings of the IEEE 82nd Vehicular Technology Conference (VTC'2015-Fall)*, p. 1–5, Boston, MA, USA, septembre 2015.
- [Lèbre et al. 2015c] M.-A. LÈBRE, F. LE MOUËL, E. MÉNARD, A. GARNAULT, B. BRADAÏ et V. PICRON, « Real scenario and simulations on GLOSA traffic light system for reduced CO₂ emissions, waiting time and travel time », *Proceedings of the 22th Intelligent Transport Systems and Services World Congress (ITS World Congress'2015)*, Bordeaux, France, octobre 2015.

- [Lèbre et al. 2016] M.-A. LÈBRE, F. LE MOUËL et E. MÉNARD, « Resilient, Decentralized V2V Online Stop-Free Strategy in a Complex Roundabout », *Proceedings of the IEEE 83rd Vehicular Technology Conference (VTC'2016-Spring)*, p. 1–5, Nanjing, China, mai 2016.
- [Lin et al. 2014a] T. LIN, H. RIVANO et F. LE MOUËL, « How to Choose the Relevant MAC Protocol for Wireless Smart Parking Urban Networks? », *Proceedings of the 11th ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks (PE-WASUN'2014)*, p. 1–8, Montreal, QC, Canada, septembre 2014.
- [Lin et al. 2014b] T. LIN, H. RIVANO et F. LE MOUËL, « Router Deployment of Streetside Parking Sensor Networks in Urban Areas », *Proceedings of the Symposium "Towards integrated modelling of urban systems"*, Lyon, France, octobre 2014.
- [Liu et al. 2014] C. LIU, J. CAO et F. LE MOUËL, « A Low-Latency Service Composition Approach in Mobile Ad Hoc Networks », *Proceedings of the 29th Annual ACM Symposium on Applied Computing (SAC'2014)*, p. 509–511, Gyeongju, Republic of Korea, mars 2014.
- [Ponge et al. 2013] J. PONGE, F. LE MOUËL et N. STOULS, « Golo, a dynamic, light and efficient language for post-invokedynamic JVM », *Proceedings of the 11th International Conference on Principles and Practice of Programming in Java (PPPJ'2013)*, p. 153–158, Stuttgart, Germany, septembre 2013.
- [Qian et al. 2015a] S. QIAN, J. CAO, F. LE MOUËL, M. LI et J. WANG, « Towards Prioritized Event Matching in a Content-based Publish/Subscribe System », *Proceedings of the 9th ACM International Conference on Distributed Event-Based Systems (DEBS'2015)*, p. 116–127, Oslo, Norway, juin 2015.
- [Qian et al. 2015b] S. QIAN, J. CAO, F. LE MOUËL, I. SAHEL et M. LI, « SCRAM : A Sharing Considered Route Assignment Mechanism for Fair Taxi Route Recommendations », *Proceedings of the 21st ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'2015)*, p. 955–964, Sydney, Australia, août 2015.
- [Royon et al. 2006] Y. ROYON, S. FRÉNOT et F. LE MOUËL, « Virtualization of Service Gateways in Multi-provider Environments », *Proceedings of the 9th International SIGSOFT Symposium on Component-Based Software Engineering (CBSE'2006), Lecture Notes in Computer Science*, vol. 4063, p. 385–392, Springer Verlag, Mälardalen University, Västerås, Sweden, juin 2006.
- [Vaz De Melo et al. 2013] P.O.S. VAZ DE MELO, A. C. VIANA, M. FIORE, K. JAFFRÈS-RUNSER, F. LE MOUËL et A. A. F. LOUREIRO, « RECAST : Telling Apart Social and Random Relationships in Dynamic Networks », *Proceedings of the 16th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM'2013)*, p. 327–334, Barcelona, Spain, novembre 2013.

B.7 WORKSHOPS INTERNATIONAUX

- [Bryce et al. 2007] C. BRYCE, M.A.C. DEKKER, S. ETALLE, D. LE MÉTAYER, F. LE MOUËL, M. MINIER, J. MORET-BAILLY et S. UBÉDA,

« Ubiquitous Privacy Protection », *Proceedings of the 5th Workshop on Ubicomp Privacy in conjunction with the 9th International Conference on Ubiquitous Computing (UbiComp'2007)*, Innsbruck, Austria, septembre 2007, Position Paper.

- [Chamekh et Le Mouël 2007] H. CHAMEKH et F. LE MOUËL, « An Ontology-based Approach to Semantically Deploy Services in Pervasive Environments », *Proceedings of the 2nd IEEE International Workshop on Services Integration in Pervasive Environments (SIPE'2007) in conjunction with the IEEE International Conference on Pervasive Services (ICPS'2007)*, p. 403–408, Istanbul, Turkey, juillet 2007.
- [Frénot et al. 2010] S. FRÉNOT, F. LE MOUËL, J. PONGE et G. SALAGNAC, « Various Extensions for the Ambient OSGi Framework », *Proceedings of the 4th International workshop on Adaptive and Dependable Mobile Ubiquitous Systems (ADAMUS'2010) in conjunction with the ACM International Conference on Pervasive Services (ICPS'2010)*, Berlin, Germany, juillet 2010, Invited Paper.
- [Ibrahim et al. 2005] N. IBRAHIM, F. LE MOUËL et S. FRÉNOT, « Automatic Negotiated Integration of Services in Pervasive Environments », *Proceedings of the Middleware for Web Services Workshop (MWS'2005) in conjunction with the 9th International IEEE Enterprise Distributed Object Computing Conference (EDOC'2005)*, Enschede, The Netherlands, septembre 2005, "Most Promising Research" Award, NICTA.
- [Ibrahim et Le Mouël 2007] N. IBRAHIM et F. LE MOUËL, « Context-aware Specialization of Semantic Rules for choosing Services in Pervasive Environments », *Proceedings of the 2nd IEEE International Workshop on Services Integration in Pervasive Environments (SIPE'2007) in conjunction with the IEEE International Conference on Pervasive Services (ICPS'2007)*, p. 391–396, Istanbul, Turkey, juillet 2007.
- [LB et Le Mouël 2007] C. LÉVY-BENCHETON et F. LE MOUËL, « Trust protocol integrating services' semantics », *Proceedings of the 4th Workshop for Ubiquitous Networking and Enablers to Context-Aware Services (Ubiq NW) in conjunction with the 4th International Symposium on Ubiquitous Computing Systems (UCS 2007)*, Akihabara, Tokyo, Japan, novembre 2007.
- [Le Mouël et al. 2006] F. LE MOUËL, N. IBRAHIM, Y. ROYON et S. FRÉNOT, « Semantic Deployment of Services in Pervasive Environments », *Proceedings of the 1st International Workshop on Requirements and Solutions for Pervasive Software Infrastructures (RSPSI'2006) in conjunction with the Pervasive'2006 Conference*, Dublin, Ireland, mai 2006.
- [Lin et al. 2014] T. LIN, H. RIVANO et F. LE MOUËL, « Performance Comparison of Contention- and Schedule-based MAC Protocols in Urban Parking Sensor Networks », *Proceedings of the ACM International Workshop on Wireless and Mobile Technologies for Smart Cities (WiMobCity'2014) in conjunction with the 15th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'2014)*, p. 39–48, Philadelphia, Pennsylvania, USA, août 2014.
- [Maingret et al. 2015] B. MAINGRET, F. LE MOUËL, J. PONGE, N. STOULS, J. CAO et Y. LOISEAU, « Towards a Decoupled Context-Oriented Programming Language for the Internet of Things », *Proceedings*

of the 7th International Workshop on Context-Oriented Programming (COP'2015) in conjunction with the European Conference on Object-Oriented Programming (ECOOP'2015), Prague, Czech Republic, juillet 2015.

B.8 CONFÉRENCES NATIONALES

[Ben Hamida et Le Mouël 2006] A. BEN HAMIDA et F. LE MOUËL, « Middleware minimal et auto-extensible pour le déploiement de services en environnement pervasif », *Actes des 3ème Journées Francophones de la Mobilité et Ubiquité (UbiMob'2006)*, CNAM, Paris, France, septembre 2006.

[Golchay et al. 2011] R. GOLCHAY, F. LE MOUËL, S. FRÉNOT et J. PONGE, « Towards Bridging IoT and Cloud Services : Proposing Smartphones as Mobile and Autonomic Service Gateways », *Actes des 7ème Journées Francophones de la Mobilité et Ubiquité (UbiMob'2011)*, p. 45–48, Toulouse, France, juin 2011, Position Paper.

[Golchay et al. 2013] R. GOLCHAY, F. LE MOUËL, J. PONGE et N. STOULS, « Les smartphones comme passerelle de services : peuvent-ils relier l'Internet des choses (IoT) et la virtualisation dans les nuages (Cloud) ? », *Conférence d'informatique en Parallélisme, Architecture et Systèmes (ComPAS'2013) - Conférence Française en Systèmes d'Exploitation (CFSE'9)*, Grenoble, France, janvier 2013, Poster.

[Ibrahim et al. 2006] N. IBRAHIM, F. LE MOUËL et S. FRÉNOT, « Intégration négociée de services dans les systèmes distribués », *Actes des Journées Composants (JC'2006)*, Perpignan, France, octobre 2006.

[Le Mouël et al. 2005] F. LE MOUËL, N. IBRAHIM et S. FRÉNOT, « Interface Matching and Combining Techniques for Services Integration », *Proceedings of the 3er Congreso Nacional de Ciencias de la Computacion (CNCC'2005)*, FCC-BUAP, Puebla, Mexico, novembre 2005.

[Le Mouël 1999] F. LE MOUËL, « Amélioration du niveau de service par la distribution adaptative d'applications dans un environnement mobile », *Journée Jeunes Chercheurs en Systèmes (JCS'1999)*, dans les actes de la 1ère Conférence Française sur les Systèmes d'Exploitation (CFSE'1), p. 229–232, Rennes, France, juin 1999.

[Ponge et Le Mouël 2012a] J. PONGE et F. LE MOUËL, « Manipulation de bytecode : démocratisons la magie noire ! », *Mix-It "Java, Agilité, Web, Innovations"*, Lyon, France, avril 2012.

[Ponge et Le Mouël 2012b] J. PONGE et F. LE MOUËL, « Manipulation de bytecode : démocratisons la magie noire ! », *1ère Conférence Française pour les Développeurs (Devoux France'2012)*, Paris, France, avril 2012.

[Ponge et Le Mouël 2013] J. PONGE et F. LE MOUËL, « JooFlux : modification de code à chaud et injection d'aspects directement dans une JVM 7 », *Actes de la Conférence d'informatique en Parallélisme, Architecture et Systèmes (ComPAS'2013) - Conférence Française en Systèmes d'Exploitation (CFSE'9)*, Grenoble, France, janvier 2013.

[Vaz De Melo et al. 2012] P.O.S. VAZ DE MELO, A. C. VIANA, M. FIORE, K. JAFFRÈS-RUNSER, F. LE MOUËL et A. A. F. LOUREIRO, « Usando Redes Aleatorias na Analise de Mobilidade », *Proceedings of the 30th Brazilian Symposium on Computer Networks and Distributed Systems (SBRC)*, Ouro Preto, Brazil, avril 2012.

B.9 ATELIERS NATIONAUX

[Ben Hamida et al. 2007] A. BEN HAMIDA, F. LE MOUËL, S. FRÉNOT et M. BEN AHMED, « Approche pour un chargement contextuel de services sur des dispositifs contraints », *Actes du 6ème atelier sur les Objets, Composants et Modèles dans l'ingénierie des Systèmes d'Information (OCM-SI'2007) organisé conjointement avec INFORSID'2007*, Perros-Guirec, France, mai 2007.

[Golchay et al. 2011] R. GOLCHAY, F. LE MOUËL, S. FRÉNOT et J. PONGE, « Les smartphones comme passerelle de services : peuvent-ils relier l'Internet des choses (IoT) et la virtualisation dans les nuages (Cloud)? », *Journées Systèmes Embarqués (Semba'2011)*, Valence, France, octobre 2011, Poster.

[Ibrahim et al. 2006] N. IBRAHIM, F. LE MOUËL et S. FRÉNOT, « Techniques d'intégration de services dans les environnements distribués », *Actes du 5ème atelier sur les Objets, Composants et Modèles dans l'ingénierie des Systèmes d'Information (OCM-SI'2006) organisé conjointement avec INFORSID'2006*, Hammamet, Tunisie, mai 2006.

[Lèbre et al. 2013] M.-A. LÈBRE, F. LE MOUËL et E. MÉNARD, « Modèle multi-échelles pour les services dans les VANET », *Journées Nationales des Communications dans les Transports (JNCT'2013)*, Nevers, France, mai 2013, Poster.

[Lin et al. 2013] T. LIN, F. LE MOUËL et H. RIVANO, « Smart On-Street Parking Assistance System », *Journées Systèmes Embarqués (Semba'2013)*, Saint Germain au Mont d'Or, France, avril 2013, Poster.

[Lin et al. 2015] T. LIN, F. LE MOUËL et H. RIVANO, « Crowdsourcing Continuous Location-Aware Update Service for Parking Assistance », *Journées ResCom'2015*, Paris, France, janvier 2015.

B.10 RAPPORTS TECHNIQUES ET DE RECHERCHE

[Ahler et al. 2006] M. AHLER, M. ANAGNOSTOU, I. PAPAIOANNOU, I. ROUSSAKI, D. TSEMETZIS, L. RÉVEILLÈRE, W. JOUVE, F. LE MOUËL, N. IBRAHIM, C. CERISARA, C. MAGERKURTH, T. PRANTE, B. KLADIS, S. TOBIES, B. VAN DER WAL, J. M. MIRANDA, Á. RAMOS, V. D. JUANA, S. KALLIO, J. KELA et E. VILDJIOUNAITE, *Test Plans*, Deliverable n°6.1, European Amigo Project, avril 2006.

[Ben Mokhtar et al. 2005] S. BEN MOKHTAR, Y.-D. BROMBERG, N. GEORGANTAS, N. IBRAHIM, V. ISSARNY, F. LE MOUËL, D. SACCHETTI, A. GÉRODOLLE, M. ANAGNOSTOU, I. PAPAIOANNOU, I. ROUSSAKI, D. TSEMETZIS, J. PARRA, H. EERTINK, R. POORTINGA, A. TOKMAKOFF, S. C. MARTINEZ, J. M. MIRANDA, Á. RAMOS, J. ZUIDWEG et M. KARJALAINEN, *Detailed Design of the Amigo Middleware Core – Service Specification*, *Interoperable Middleware Core*, Deliverable n°3.1b, European Amigo Project, septembre 2005.

- [Ben Mokhtar et al. 2006a] S. BEN MOKHTAR, Y.-D. BROMBERG, N. GEORGANTAS, N. IBRAHIM, V. ISSARNY, W. JOUVE, F. LE MOUËL, L. RÉVEILLÈRE, D. SACCHETTI, F. TARTANOGLU, A. GÉRODOLLE, M. VALLÉE, M. ANAGNOSTOU, I. PAPAIOANNOU, I. ROUSSAKI, D. TSESMETZIS, J. PARRA, E. NAROSKA, R. MEVISSSEN, S. TOBIES, R. POORTINGA, P. PAWAR, A. TOKMAKOFF, J. GARCÍA, J. M. MIRANDA, M. PLANAGUMÀ, Á. RAMOS, D. ROLDÁN, J. KALAOJA, J. KANTOROVITCH, I. NISKANEN et T. PIIRAINEN, *Amigo Middleware Core Enhanced : Prototype Implementation & Documentation*, Deliverable n°3.3, European Amigo Project, octobre 2006.
- [Ben Mokhtar et al. 2006b] S. BEN MOKHTAR, Y.-D. BROMBERG, N. GEORGANTAS, N. IBRAHIM, V. ISSARNY, A. KAUL, F. LE MOUËL, D. SACCHETTI, F. TARTANOGLU, A. GÉRODOLLE, M. VALLÉE, M. ANAGNOSTOU, I. PAPAIOANNOU, I. ROUSSAKI, D. TSESMETZIS, J. PARRA, M. AHLER, R. MEVISSSEN, D. SCHAFFRATH, S. TOBIES, H. EERTINK, P. PAWAR, R. POORTINGA, A. TOKMAKOFF, J. GARCÍA, J. M. MIRANDA, M. PLANAGUMÀ, Á. RAMOS, S. SORRIBAS, J. KALAOJA et J. KANTOROVITCH, *Amigo Middleware Core : Prototype Implementation & Documentation*, Deliverable n°3.2, European Amigo Project, mars 2006.
- [Choo et al. 2011] W. L. CHOO, F. LE MOUËL, K. JAFFRÈS RUNSER et M. FIORE, *Adding Network Coding Capabilities to the WSNet Simulator*, Technical Report n°RT-0405, INRIA, 03 2011.
- [Le Mouël 1997] F. LE MOUËL, *Filtrage intelligent de documents textuels*, Research Report n°UAR/C/97/0280, Alcatel Alsthom Research, août 1997.
- [Lèbre et al. 2014] M.-A. LÈBRE, F. LE MOUËL, E. MÉNARD, J. DILLSCHNEIDER et R. DENIS, *VANET Applications : Hot Use Cases*, Research report, INRIA CITI Laboratory, INSA Lyon, juillet 2014.
- [Ledoux et al. 2001] T. LEDOUX et AL., *État de l'art sur l'adaptabilité*, Délivrable n°1.1, Projet RNTL ARCAD, décembre 2001.
- [Lin et al. 2014] T. LIN, H. RIVANO et F. LE MOUËL, *Traffic Modeling and Analysis in the Performance of Parking Sensor Networks*, Research Report n°RR-8480, INRIA, février 2014.
- [Ponge et al. 2015] J. PONGE, F. LE MOUËL, N. STOULS et Y. LOISEAU, *Opportunities for a Truffle-based Golo Interpreter*, Technical report, CITI - CITI Centre of Innovation in Telecommunications and Integration of services, avril 2015.
- [Ponge et Le Mouël 2012] J. PONGE et F. LE MOUËL, *JooFlux : Hijacking Java 7 InvokeDynamic To Support Live Code Modifications*, Research report, INRIA CITI Lab, INSA Lyon, octobre 2012.
- [Ramparany et al. 2005] F. RAMPARANY, J. PIERSON, T. FLURY, G. PRIVAT, A. GÉRODOLLE, H. NATIVIDAD, F. LE MOUËL, L. RÉVEILLÈRE, N. GEORGANTAS, S. BEN MOKHTAR, F. TARTANOGLU, V. ISSARNY, C. CERISARA, B. KLADIS, M. GILBERT, R. MEVISSSEN, B. VAN DER WAL, H. DE GROOT, M. VAN HARTSKAMP, P. VAN DER STOK, H. EERTINK, R. POORTINGA, T. BROENS, S. C. MARTINEZ, J. ARRIBAS, J. M. MIRANDA, G. VOM BÖGEL, P. KALLIO, J. ZHOU et J. KANTOROVITCHE, *State of the art analysis including assessment of system architectures for ambient intelligence*, Deliverable n°2.2, European Amigo Project, avril 2005.

[Roussaki et al. 2006] I. ROUSSAKI, I. PAPAIOANNOU, D. TSESMEZIS, M. ANAGNOSTOU, L. RÉVEILLÈRE, W. JOUVE, F. LE MOUËL, N. IBRAHIM, C. MAGERKURTH, T. PRANTE, B. KLADIS, J. M. MIRANDA, Á. RAMOS, M. BEL, F. Á. MELÉNDEZ et E. VILDJIOUNAITE, *Home Information and Entertainment Application Specification*, Deliverable n°6.2, European Amigo Project, juin 2006.

B.11 MÉMOIRES

[Le Mouël 1997] F. LE MOUËL, *Filtrage intelligent de documents textuels*, Thèse de Master, Université de Rennes 1, Rennes, France, septembre 1997.

[Le Mouël 2003] F. LE MOUËL, *Environnement adaptatif d'exécution distribuée d'applications dans un contexte mobile*, Thèse de doctorat, Université de Rennes 1, Rennes, France, décembre 2003.

B.12 MÉMOIRES - ENCADREMENT

[Ben Hamida 2010] A. BEN HAMIDA, *AxSeL : un intergiciel pour le déploiement contextuel et autonome de services dans les environnements pervasifs*, Thèse de doctorat, INSA Lyon et ENSI, Université de la Manouba, Lyon, France, février 2010.

[Golchay 2016] R. GOLCHAY, *From Mobile to Cloud : Using Bio-Inspired Algorithms for Collaborative Application Offloading*, Thèse de doctorat, Université de Lyon, INSA Lyon, Lyon, France, janvier 2016.

[Ibrahim 2008] N. IBRAHIM, *Spontaneous Integration of Services in Pervasive Environments*, Thèse de doctorat, INSA Lyon, Lyon, France, septembre 2008.

[Lèbre 2016] MARIE-ANGLE LÈBRE, *De l'impact d'une décision locale et autonome sur les systèmes de transport intelligent à différentes échelles*, Thèse de doctorat, Université de Lyon, INSA Lyon, Lyon, France, janvier 2016.

[Lin 2015] T. LIN, *Smart Parking : Network, Infrastructure and Urban Service*, Thèse de doctorat, Université de Lyon, INSA Lyon, Lyon, France, décembre 2015.

B.13 DIVERS

[Le Mouël 2004] F. LE MOUËL, *Calendrier prévisionnel du déroulement d'une thèse*, juin 2004, version 1.2.

[Lèbre et al. 2015] M.-A. LÈBRE, F. LE MOUËL et E. MÉNARD, *Microscopic vehicular mobility trace of Europarc roundabout, Creteil, France*, avril 2015, Open data trace, v1.0, Creative Commons Attribution-NonCommercial 4.0 International License.

BIBLIOGRAPHIE

- [Abiteboul et al. 2007] S. ABITEBOUL, I. DAR, R. POP, G. VASILE, D. VO-DISLAV et N. PREDA, « Large scale P2P distribution of open-source software », *Proceedings of the 33rd international conference on Very large data bases, VLDB Endowment*, p. 1390–1393, 2007.
- [Alazawi et al. 2011] Z. ALAZAWI, S. ALTOWAIJRI, R. MEHMOOD et M. B. ABDLJABAR, « Intelligent disaster management system based on cloud-enabled vehicular networks », *Proceedings of the 11th International Conference on ITS Telecommunications (ITST)*, IEEE, p. 361–368, 2011.
- [Alliance 2007] OSGI ALLIANCE, « Osgi service platform, core specification, release 4, version 4.1 », *OSGi Specification*, 2007.
- [Appeltauer et al. 2009] M. APPELTAUER, R. HIRSCHFELD, M. HAUPT, J. LINCKE et M. PERSCHIED, « A Comparison of Context-oriented Programming Languages », *Proceedings of the International Workshop on Context-Oriented Programming*, p. 6 :1–6 :6, ACM, New York, NY, USA, 2009.
- [Arnold 1999] K. ARNOLD, « The Jini Architecture : Dynamic Services in a Flexible Network », *Proceedings of the 36th Annual ACM/IEEE Design Automation Conference*, p. 157–162, ACM, New York, NY, USA, 1999.
- [Asbury et Weiner 2001] S. ASBURY et S. R. WEINER, *Developing Java Enterprise Applications*, John Wiley & Sons, Inc., New York, NY, USA, 2001, 2nd édition.
- [Atallah 1983] M. J. ATALLAH, « Dynamic computational geometry », 1983.
- [Audureau 2016] W. AUDUREAU, *Jeu de go : pour Lee Sedol, la victoire de la machine est moins tactique que psychologique*, Le Monde, mars 2016, http://www.lemonde.fr/pixels/article/2016/03/15/jeu-de-go-pour-lee-sedol-la-victoire-d-alphago-est-psychologique-plus-que-sportive_4883286_4408996.html.
- [Barr 2006] J. BARR, *Amazon EC2 Beta*, août 2006, https://aws.amazon.com/fr/blogs/aws/amazon_ec2_beta/.
- [Basten et al. 2003] T. BASTEN, L. BENINI, A. CHANDRAKASAN, M. LINDWER, J. LIU, R. MIN et F. ZHAO, « Scaling into Ambient Intelligence », *Proceedings of the Conference on Design, Automation and Test in Europe - Volume 1*, p. 10076–, IEEE Computer Society, Washington, DC, USA, 2003.
- [Bloom 2013] P. BLOOM, *Just babies : The origins of good and evil*, Crown, 2013.
- [Bloom 2014] P. BLOOM, *Did God Make These Babies Moral ?*, New Republic, janvier 2014, <https://newrepublic.com/article/116200/moral-design-latest-form-intelligent-design-its-wrong>.

- [Bostrom et Sandberg 2009] N. BOSTROM et A. SANDBERG, « The wisdom of nature : an evolutionary heuristic for human enhancement », *Human enhancement*, 2009, p. 375–416.
- [Box et al. 2000] D. BOX, D. EHNEBUSKE, G. KAKIVAYA, A. LAYMAN, N. MENDELSON, H. F. NIELSEN, S. THATTE et D. WINER, *Simple Object Access Protocol (SOAP) 1.1*, W3c note, mai 2000.
- [Bringhurst 2008] R. BRINGHURST, *The Elements of Typographic Style*, Hartley & Marks, Publishers, Point Roberts, WA, USA, 2008, Version 3.2. First published 1992, 3rd edition 2004, updated 2005 & 2008.
- [Bruneton et al. 2002] E. BRUNETON, R. LENGLET et T. COUPAYE, « ASM : a code manipulation tool to implement adaptable systems », *Adaptable and Extensible Component Systems (Journée ASF Systèmes à composants adaptables et extensibles)*, Grenoble, France, novembre 2002, <http://asm.ow2.org>.
- [Bruneton et al. 2006] E. BRUNETON, T. COUPAYE, M. LECLERCQ, V. QUÉMA et J.-B. STEFANI, « The fractal component model and its support in java », *Software : Practice and Experience*, vol. 36, n°11-12, 2006, p. 1257–1284.
- [Carzaniga et Wolf 2003] A. CARZANIGA et A. L. WOLF, « Forwarding in a content-based network », *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, ACM, p. 163–174, 2003.
- [Castelvecchi 2016] D. CASTELVECCHI, « Can we open the black box of AI? », *Nature*, vol. 538, n°7623, octobre 2016, p. 20–23.
- [Chen et al. 2008] G. CHEN, M. LI et D. KOTZ, « Data-centric Middleware for Context-aware Pervasive Computing », *Pervasive and Mobile Computing*, vol. 4, n°2, avril 2008, p. 216–253.
- [Chen et al. 2015] M. CHEN, Y. ZHANG, Y. LI, M. M. HASSAN et A. ALAMRI, « AIWAC : affective interaction through wearable computing and cloud technology », *IEEE Wireless Communications*, vol. 22, n°1, 2015, p. 20–27.
- [Chow et al. 2014] J. M. CHOW, J. M. GAMBETTA, E. MAGESAN, D. W. ABRAHAM, A. W. CROSS, B. R. JOHNSON, N. A. MASLUK, C. A. RYAN, J. A. SMOLIN, S. J. SRINIVASAN et M. STEFFEN, « Implementing a strand of a scalable fault-tolerant quantum computing fabric », *Nature Communications*, vol. 5, n°4015, avril 2014.
- [Christensen et al. 2001] E. CHRISTENSEN, F. CURBERA, G. MEREDITH et S. WEERAWARANA, *Web Service Definition Language (WSDL)*, W3c note, mars 2001.
- [Cisco et Frost & Sullivan 2008] CISCO et FROST & SULLIVAN, *Intelligent Middleware*, White Paper, 2008.
- [Clojure] CLOJURE, <http://clojure.org>.
- [Cook et al. 2009] D. J. COOK, J. C. AUGUSTO et V. R. JAKKULA, « Review : Ambient Intelligence : Technologies, Applications, and Opportunities », *Pervasive and Mobile Computing*, vol. 5, n°4, août 2009, p. 277–298.

- [Cox 1986] B. J. COX, *Object Oriented Programming : An Evolutionary Approach*, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1986.
- [Dahl 1968] O.-J. DAHL, *SIMULA 67 Common Base Language, (Norwegian Computing Center. Publication)*, 1968.
- [des Ligneris 2005] B. DES LIGNERIS, « Virtualization of Linux Based Computers : The Linux-VServer Project », *Proceedings of the 19th International Symposium on High Performance Computing Systems and Applications*, p. 340–346, IEEE Computer Society, Washington, DC, USA, 2005.
- [Desai et al. 2016] S. B. DESAI, S. R. MADHVPATHY, A. B. SACHID, J. P. LLINAS, Q. WANG, G. H. AHN, G. PITNER, M. J. KIM, J. BOKOR, C. HU, H.-S. P. WONG et A. JAVEY, « MoS₂ transistors with 1-nanometer gate lengths », *Science*, vol. 354, n°6308, 2016, p. 99–102.
- [DIMACS] DIMACS, <http://www.dis.uniroma1.it/challenge9/download.shtml>.
- [Dinn 2011] A. E. DINN, « Flexible, Dynamic Injection of Structured Advice Using Byteman », *Proceedings of the Tenth International Conference on Aspect-oriented Software Development Companion*, p. 41–50, ACM, New York, NY, USA, 2011, <http://www.jboss.org/byteman/>.
- [Dorigo et Di Caro 1999] M. DORIGO et G. DI CARO, « New Ideas in Optimization », chapitre The Ant Colony Optimization Meta-heuristic, p. 11–32, McGraw-Hill Ltd., UK, Maidenhead, UK, England, 1999.
- [Dorigo et Gambardella 1997] M. DORIGO et L. M. GAMBARDELLA, « Ant colony system : a cooperative learning approach to the traveling salesman problem », *IEEE Transactions on evolutionary computation*, vol. 1, n°1, 1997, p. 53–66.
- [Duus et Cooray 2015] R. DUUS et M. COORAY, *How we discovered the dark side of wearable fitness trackers*, The Conversation, juin 2015, <https://theconversation.com/how-we-discovered-the-dark-side-of-wearable-fitness-trackers-43363>.
- [Erl 2005] T. ERL, *Service-Oriented Architecture : Concepts, Technology, and Design*, Prentice Hall PTR, Upper Saddle River, NJ, USA, 2005.
- [Felix] FELIX, Apache, <http://felix.apache.org>.
- [Fielding 2000] R. T. FIELDING, *Architectural Styles and the Design of Network-based Software Architectures*, Thèse de doctorat, 2000, AAI9980887.
- [Frieder et Segal 1991] O. FRIEDER et M. E. SEGAL, « On dynamically updating a computer program : From concept to prototype », *Journal of Systems and Software*, vol. 14, n°2, 1991, p. 111 – 128.
- [Fritz et al. 2014] T. FRITZ, E. M HUANG, G. C MURPHY et T. ZIMMERMANN, « Persuasive technology in the real world : a study of long-term use of activity sensing devices for fitness », *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, p. 487–496, 2014.

- [Gall 2005] N. GALL, *Update on the origin of the term "middleware"*, juillet 2005, http://ironick.typepad.com/ironick/2005/07/update_on_the_o.html.
- [Gartner 2015] GARTNER, *Gartner Says 6.4 Billion Connected "Things" Will Be in Use in 2016, Up 30 Percent From 2015*, novembre 2015, <http://www.gartner.com/newsroom/id/3165317>.
- [Goldberg et Kay 1976] A. GOLDBERG et A. KAY, *Smalltalk-72 - Instruction Manual*, Xerox Palo Alto Research Center, mars 1976.
- [González et al. 2011] S. GONZÁLEZ, N. CARDOZO, K. MENS, A. CÁDIZ, J.-C. LIBBRECHT et J. GOFFAUX, *Subjective-C : Bringing Context to Mobile Platform Programming*, p. 246–265, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.
- [Goodfellow et al. 2015] I. J. GOODFELLOW, D. ERHAN, P. L. CARRIER, A. COURVILLE, M. MIRZA, B. HAMNER, W. CUKIERSKI, Y. TANG, D. THALER, D.-H. LEE *et al.*, « Challenges in representation learning : A report on three machine learning contests », *Neural Networks*, vol. 64, 2015, p. 59–63.
- [Gosling et al. 2013] J. GOSLING, B. JOY, G. L. STEELE JR., G. BRACHA et A. BUCKLEY, *The Java Language Specification, Java SE 7 Edition*, Addison-Wesley Professional, 2013, 1st édition.
- [Gosling 1995] J. GOSLING, « Java Intermediate Bytecodes : ACM SIGPLAN Workshop on Intermediate Representations (IR'95) », *SIGPLAN Notices*, vol. 30, n°3, mars 1995, p. 111–118.
- [Grammatikakis et al. 2003] M. GRAMMATIKAKIS, M. COPPOLA et F. SENSINI, *Software for Multiprocessor Networks on Chip*, p. 281–303, Springer US, Boston, MA, 2003.
- [Groovy] GROOVY, <http://groovy.codehaus.org>.
- [Grosso 2001] W. GROSSO, *Java RMI*, O'Reilly & Associates, Inc., Sebastopol, CA, USA, 2001, 1st édition.
- [Gubbi et al. 2013] J. GUBBI, R. BUYYA, S. MARUSIC et M. PALANISWAMI, « Internet of Things (IoT) : A vision, architectural elements, and future directions », *Future Generation Computer Systems*, vol. 29, n°7, 2013, p. 1645–1660.
- [Guerrero-Ibáñez et al. 2012] A. GUERRERO-IBÁÑEZ, C. FLORES-CORTÉS, JRG PULIDO, M. ANDRADE-ARÉCHIGA et P. DAMIAN-REYES, *Emerging technologies for urban traffic management*, INTECH Open Access Publisher, 2012.
- [Henning 2006] M. HENNING, « The Rise and Fall of CORBA », *Queue*, vol. 4, n°5, juin 2006, p. 28–34.
- [Heredia 2011] E. A. HEREDIA, *An Introduction to the DLNA Architecture : Network Technologies for Media Devices*, Wiley Publishing, 2011, 1st édition.
- [Hern 2016] A. HERN, *Pokémon Go : Who owns the virtual space around your home?*, *The Guardian*, juillet 2016, <https://www.theguardian.com/technology/2016/jul/13/pokemon-virtual-space-home>.

- [Hicks et Nettles 2005] M. HICKS et S. NETTLES, « Dynamic Software Updating », *ACM Transactions on Programming Languages and Systems*, vol. 27, n°6, novembre 2005, p. 1049–1096.
- [Hirschfeld et al. 2008] R. HIRSCHFELD, P. COSTANZA et O. NIERSTRASZ, « Context-Oriented Programming », *Journal of Object Technology*, vol. 7, n°3, mars 2008, p. 125–151.
- [Horstmann et Kirtland 1997] M. HORSTMANN et M. KIRTLAND, *DCOM Architecture*, Technical report, Microsoft Corporation, Redmond, WA, 1997.
- [Ilyas et Mahgoub 2016] MOHAMMAD ILYAS et IMAD MAHGOUB, *Smart Dust : Sensor network applications, architecture and design*, CRC press, 2016.
- [Ishikawa et al. 2005] F. ISHIKAWA, N. YOSHIOKA et S. HONIDEN, « Mobile agent system for Web service integration in pervasive network », *Systems and Computers in Japan*, vol. 36, n°11, 2005, p. 34–48.
- [ISTAG 2001] ISTAG, *Scenarios for Ambient Intelligence in 2010*, Research report, European Commission - Community Research, Seville, février 2001.
- [ISTAG 2003] ISTAG, *Ambient Intelligence : from vision to reality*, Research report, European Commission - Community Research, 2003.
- [Jeronimo et Weast 2003] M. JERONIMO et J. WEAST, *UPnP Design by Example : A Software Developer's Guide to Universal Plug and Play*, Intel Press, 2003.
- [JRuby] JRUBY, <http://jruby.org>.
- [Juric 2006] M. B. JURIC, *Business Process Execution Language for Web Services - BPEL and BPEL4WS 2/e*, Packt Publishing, 2006.
- [Jython] JYTHON, <http://www.jython.org>.
- [Kamp et Watson 2000] P.-H. KAMP et R. N. M. WATSON, *Jails : Confining the omnipotent root*, Technical report, The FreeBSD Project, 2000.
- [Karypis et Kumar 1998] G. KARYPIS et V. KUMAR, « A fast and high quality multilevel scheme for partitioning irregular graphs », *SIAM Journal on scientific Computing*, vol. 20, n°1, 1998, p. 359–392.
- [Katasonov et al. 2008] A. KATASONOV, O. KAYKOVA, O. KHRIYENKO, S. NIKITIN et V. TERZIYAN, « Smart Semantic Middleware for the Internet of Things », *Proceedings of the 5th International Conference on Informatics in Control, Automation and Robotics*, p. 11–15, 2008.
- [Kay 2004] A. KAY, « A Conversation with Alan Kay », *ACM Queue*, vol. 2, n°9, décembre 2004.
- [Kephart et Chess 2003] J. O. KEPHART et D. M. CHESS, « The Vision of Autonomic Computing », *Computer*, vol. 36, n°1, janvier 2003, p. 41–50.
- [Kiczales et al. 2001] G. KICZALES, E. HILSDALE, J. HUGUNIN, M. KERSTEN, J. PALM et W. GRISWOLD, « Getting Started with ASPECTJ », *Communications of the ACM*, vol. 44, n°10, octobre 2001, p. 59–65, <http://www.eclipse.org/aspectj/>.

- [Knuth 1974] D. E. KNUTH, « Computer Programming as an Art », *Communications of the ACM*, vol. 17, n°12, décembre 1974, p. 667–673ds.
- [Kosmyna et al. 2016] N. KOSMYNA, F. TARPIN-BERNARD, N. BONNEFOND et B. RIVET, « Feasibility of BCI Control in a Realistic Smart Home Environment », *Frontiers in Human Neuroscience*, vol. 10, 2016, p. 416.
- [Krajzewicz et al. 2012] D. KRAJZEWICZ, J. ERDMANN, M. BEHRISCH et L. BIEKER, « Recent development and applications of SUMO—simulation of urban mobility », *International Journal On Advances in Systems and Measurements*, vol. 5, n°3&4, 2012.
- [Kumar 2015] S. KUMAR, *Fundamental Limits to Moore's Law*, Research Report n°arXiv :1511.05956, Stanford University, Stanford, CA 94305, USA., novembre 2015.
- [Lambert et al. 2013] J. H. LAMBERT, A. I. PARLAK, Q. ZHOU, J. S. MILLER, M. D. FONTAINE, T. M. GUTERBOCK, J. L. CLEMENTS et S. A. THEKDI, « Understanding and managing disaster evacuation on a transportation network », *Accident Analysis & Prevention*, vol. 50, 2013, p. 645–658.
- [Laurent et al. 2001] S. ST. LAURENT, E. DUMBILL, J. JOHNSTON et D. WINER, *Programming Web Services with XML-RPC*, O'Reilly & Associates, Inc., Sebastopol, CA, USA, 2001.
- [Lee et Seligman 2000] R. LEE et S. SELIGMAN, *The JNDI API Tutorial and Reference : Building Directory-Enabled Java Applications*, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2000.
- [Li et al. 2012] L. LI, Z. JIN, G. LI, L. ZHENG et Q. WEI, « Modeling and analyzing the reliability and cost of service composition in the IoT : A probabilistic approach », *Proceedings of 2012 IEEE 19th International Conference on Web Services (ICWS)*, IEEE, p. 584–591, 2012.
- [Luettel et al. 2012] T. LUETTEL, M. HIMMELSBACH et H.-J. WUENSCH, « Autonomous ground vehicles - Concepts and a path to the future », *Proceedings of the IEEE*, vol. 100, n°Special Centennial Issue, 2012, p. 1831–1839.
- [Martin et al. 2004] D. MARTIN, M. BURSTEIN, E. HOBBS, O. LASSILA, D. MCDERMOTT, S. MCILRAITH, S. NARAYANAN, B. PARSIA, T. PAYNE, E. SIRIN, N. SRINIVASAN et K. SYCARA, *OWL-S : Semantic Markup for Web Services*, W3c member submission, novembre 2004.
- [Minsky 1970] M. MINSKY, « Form and Content in Computer Science (1970 ACM Turing Lecture) », *Journal of the ACM*, vol. 17, n°2, avril 1970, p. 197–215.
- [Miranda et al. 2015] J. MIRANDA, N. MÄKITALO, J. GARCIA-ALONSO, J. BERROCAL, T. MIKKONEN, C. CANAL et J. M. MURILLO, « From the Internet of Things to the Internet of People », *IEEE Internet Computing*, vol. 19, n°2, 2015, p. 40–47.
- [Mockford 2004] K. MOCKFORD, « Web services architecture », *BT Technology Journal*, vol. 22, n°1, 2004, p. 19–26.

- [Moore 1965] G. E. MOORE, « Cramming More Components onto Integrated Circuits », *Electronics*, vol. 38, n°8, avril 1965, p. 114–117.
- [Motorola 2016] MOTOROLA, *Intelligent Middleware - The Invisible Glue to Enable Truly Integrated Communications*, White Paper, 2016.
- [MT et Wolshon 2013] P. MURRAY-TUITE et B. WOLSHON, « Evacuation transportation modeling : An overview of research, development, and practice », *Transportation Research Part C : Emerging Technologies*, vol. 27, 2013, p. 25–45.
- [Naur et Randell 1968] P. NAUR et B. RANDELL, *Software Engineering*, Research report, NATO Software Engineering Conference, Garmisch, Germany, octobre 1968, (eds).
- [Newborn 2012] M. NEWBORN, *Kasparov versus Deep Blue : Computer chess comes of age*, Springer Science & Business Media, 2012.
- [Newcomer et Robinson 2009] E. NEWCOMER et I. ROBINSON, *Web Services Coordination (WS-Coordination)*, OASIS Standard n°version 1.2, OASIS, février 2009, (chairs).
- [NIS 2001] NIST - National Institute of Standards and Technology, *Pervasive Computing Conference*, Gaithersburg, Maryland, mai 2001.
- [Odersky et Zenger 2005] M. ODERSKY et M. ZENGER, « Independently Extensible Solutions to the Expression Problem », *Proceedings of The Twelfth International Workshop on Foundations of Object-Oriented Languages (FOOL '12)*, ACM SIGPLAN, Long Beach, California, USA, janvier 2005.
- [O'Grady et al. 2016] M. J. O'GRADY, C. MULDOON, D. CARR, J. WAN, B. KROON et G. M. P. O'HARE, « Intelligent Sensing for Citizen Science », *Mobile Networks and Applications*, vol. 21, n°2, 2016, p. 375–385.
- [o] 2013] GOVERNMENT OF JAPAN, *Working Group for Planning Against Tokai Earthquake under Cabinet Office*, Reports on damages to lifelines and transport facilities, 2013.
- [OMG 1992] OMG, *The Common Object Request Broker : Architecture and Specification*, Technical Report n°91.12.1 rev 1.1, Object Management Group, 1992.
- [Oracle] ORACLE, *Java 7*, <http://docs.oracle.com/javase/7/docs/api/>.
- [Paolucci et al. 2002] M. PAOLUCCI, T. KAWAMURA, T. R. PAYNE et K. SYCARA, « Semantic matching of web services capabilities », *Proceedings of the International Semantic Web Conference*, Springer, p. 333–347, 2002.
- [Parrend et al. 2006] P. PARREND, Y. ROYON et N. IBRAHIM, « Service-Oriented Distributed Communities in Residential Environments », *Proceedings of the 1st IEEE International Workshop on Services Integration in Pervasive Environments (SIPE'2006) in conjunction with the IEEE International Conference on Pervasive Services (ICPS'2006)*, Lyon, France, juin 2006.

- [Paulin et al. 2006] P. G. PAULIN, C. PILKINGTON, M. LANGEVIN, E. BENSOUUDANE, O. BENNY, D. LYONNARD, B. LAVIGUEUR et D. LO, « Distributed Object Models for Multi-processor SoC's, with Application to Low-power Multimedia Wireless Systems », *Proceedings of the Conference on Design, Automation and Test in Europe*, p. 482–487, European Design and Automation Association, 3001 Leuven, Belgium, Belgium, 2006.
- [Pease et al. 2016] S. G. PEASE, I. W. PHILLIPS et L. GUAN, « Adaptive Intelligent Middleware Architecture for Mobile Real-Time Communications », *IEEE Transactions on Mobile Computing*, vol. 15, n°3, 2016, p. 572–585.
- [Peeta et Hsu 2009] S. PEETA et Y.-T. HSU, *Integrating supply and demand aspects of transportation for mass evacuation under disasters*, Technical report, NEXTRANS, 2009.
- [Perlis 1982] A. J. PERLIS, « Special Feature : Epigrams on Programming », *SIGPLAN Notices*, vol. 17, n°9, septembre 1982, p. 7–13.
- [Perry 2002] J. S. PERRY, *Java Management Extensions*, O'Reilly & Associates, Inc., Sebastopol, CA, USA, 2002, 1st édition.
- [Ponge 2011] J. PONGE, « Fork and Join : Java Can Excel at Painless Parallel Programming Too! », *Oracle Technology Network*, juillet 2011.
- [Popovici et al. 2010] K. POPOVICI, F. ROUSSEAU, A. A. JERRAYA et M. WOLF, *Embedded Software Design and Programming of Multiprocessor System-on-Chip : Simulink and System C Case Studies*, Springer Publishing Company, Incorporated, 2010, 1st édition.
- [Pozo et Miller 2000] R. POZO et B MILLER, *SCImark 2.0*, 2000, <http://math.nist.gov/scimark2/>.
- [Puvvala et Pota 2003] PUVVALA et A. POTA, *.NET for Java Developers : Migrating to C#*, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2003.
- [Qu et al. 2013] X. QU, P. JJ ALVAREZ et Q. LI, « Applications of nanotechnology in water and wastewater treatment », *Water research*, vol. 47, n°12, 2013, p. 3931–3946.
- [Reijers et al. 2013] N. REIJERS, K.-J. LIN, Y.-C. WANG, C.-S. SHIH et J. Y.-J. HSU, « Design of an Intelligent Middleware for Flexible Sensor Configuration in M2M Systems », *SENSORNETS 2013 - Proceedings of the 2nd International Conference on Sensor Networks*, p. 41–46, SciTePress, Barcelona, Spain, février 2013.
- [Rhino Javascript] RHINO JAVASCRIPT, <https://developer.mozilla.org/en-US/docs/Rhino>.
- [Riahi et al. 2013] A. RIAHI, Y. CHALLAL, E. NATALIZIO, Z. CHTOUROU et A. BOUABDALLAH, « A systemic approach for IoT security », *Proceedings of 2013 IEEE International Conference on Distributed Computing in Sensor Systems*, IEEE, p. 351–355, 2013.
- [Richardson et al. 2013] L. RICHARDSON, M. AMUNDSEN et S. RUBY, *RESTful Web APIs*, O'Reilly Media, Inc., 2013.

- [Satyanarayanan 2001] M. SATYANARAYANAN, « Pervasive Computing : Vision and Challenges », *IEEE Personal Communications*, vol. 8, 2001, p. 10–17.
- [Schmidhuber 2015] J. SCHMIDHUBER, « Deep learning in neural networks : An overview », *Neural Networks*, vol. 61, 2015, p. 85–117.
- [Schonwalder et al. 2008] J. SCHONWALDER, V. MARINOV et M. BURGESS, « Integrating cfengine and scli : Managing network devices like host systems », *Proceedings of IEEE Network Operations and Management Symposium (NOMS '2008)*, 2008.
- [Seyedi et al. 2013] M. SEYEDI, B. KIBRET, D. T. H. LAI et M. FAULKNER, « A Survey on Intrabody Communications for Body Area Network Applications », *IEEE Transactions on Biomedical Engineering*, vol. 60, n°8, Aug 2013, p. 2067–2079.
- [Shoch 1979] J. F. SHOCH, « An Overview of the Programming Language Smalltalk-72 », *SIGPLAN Notices*, vol. 14, n°9, septembre 1979, p. 64–73.
- [Silver et al. 2016] D. SILVER, A. HUANG, C. J. MADDISON, A. GUEZ, L. SIFRE, G. VAN DEN DRIESSCHE, J. SCHRITTWIESER, I. ANTONOGLOU, V. PANNEERSHELVA, M. LANCTOT, S. DIELEMAN, D. GREWE, J. NHAM, N. KALCHBRENNER, I. SUTSKEVER, T. LILICRAP, M. LEACH, K. KAVUKCUOGLU, T. GRAEPEL et D. HASSABIS, « Mastering the game of Go with deep neural networks and tree search », *Nature*, vol. 529, n°7587, janvier 2016, p. 484–489.
- [Smith et al. 2014] D. SMITH, S. DJAHEL et J. MURPHY, « A SUMO based evaluation of road incidents' impact on traffic congestion level in smart cities », *Proceedings of the 2014 IEEE 39th Conference on Local Computer Networks Workshops (LCN Workshops)*, IEEE, p. 702–710, 2014.
- [Spiekermann 2012] S. SPIEKERMANN, « The challenges of privacy by design », *Communications of the ACM*, vol. 55, n°7, 2012, p. 38–40.
- [Stolpe 2016] M. STOLPE, « The Internet of Things : Opportunities and Challenges for Distributed Data Analysis », *SIGKDD Explorations Newsletter*, vol. 18, n°1, août 2016, p. 15–34.
- [Tanenbaum 2002] A. TANENBAUM, *Computer Networks*, Prentice Hall Professional Technical Reference, 2002, 4th édition.
- [Townsend et al. 2004] J. J. TOWNSEND, D. RIZ et D. SCHAFFER, *Building Portals, Intranets, and Corporate Web Sites Using Microsoft Servers, 1/e*, Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA, 2004.
- [Townsend 2008] E. TOWNSEND, *The 25-Year History of Service-Oriented Architecture*, juillet 2008, http://www.eriktownsend.com/look/doc_download/12-1-the-25-year-history-of-service-oriented-architecture.
- [Uppoor et Fiore 2011] S. UPPOOR et M. FIORE, « Large-scale urban vehicular mobility for networking research », *Proceedings of the Vehicular Networking Conference (VNC), 2011 IEEE*, IEEE, p. 62–69, 2011.

- [Verbeek 2011] P.-P. VERBEEK, *Moralizing technology : Understanding and designing the morality of things*, University of Chicago Press, 2011.
- [Waldo 1999] J. WALDO, « The Jini Architecture for Network-centric Computing », *Communications of the ACM*, vol. 42, n°7, juillet 1999, p. 76–82.
- [Walsh 2002] A. E. WALSH (éd.), *Uddi, Soap, and Wsdl : The Web Services Specification Reference*, Prentice Hall Professional Technical Reference, 2002.
- [Wein et al. 2010] L. M WEIN, Y. CHOI et S. DENUIT, « Analyzing Evacuation Versus Shelter-in-Place Strategies After a Terrorist Nuclear Detonation », *Risk Analysis*, vol. 30, n°9, 2010, p. 1315–1327.
- [Weiser 1991] M. WEISER, « The Computer for the 21st Century », *Scientific American*, vol. 265, n°3, January 1991, p. 66–75.
- [White 1975] J. E. WHITE, *A High-Level Framework for Network-Based Resource Sharing*, Internet RFC 707, December 1975.
- [Wikipedia 2016a] WIKIPEDIA, *Conjectures de Moore*, octobre 2016, https://fr.wikipedia.org/wiki/Loi_de_Moore.
- [Wikipedia 2016b] WIKIPEDIA, *Logiciel*, octobre 2016, <https://fr.wikipedia.org/wiki/Logiciel>.
- [Wikipedia 2016c] WIKIPEDIA, *Réseau informatique*, octobre 2016, https://fr.wikipedia.org/wiki/Réseau_informatique.
- [Wisetjindawat et al. 2015] W. WISETJINDAWAT, H. ITO et M FUJITA, « Integrating a Stochastic Failure of the Road Network and a Road Recovery Strategy into the Planning of Goods Distribution in the Aftermath of a Large-Scale Earthquake », 2015.
- [Wolf 2004] W. WOLF, « The future of multiprocessor systems-on-chips », *Proceedings of the 41st Design Automation Conference*, IEEE, p. 681–685, 2004.
- [Wolshon et McArdle 2009] B. WOLSHON et B. MCARDLE, « Temporo-spatial analysis of Hurricane Katrina regional evacuation traffic patterns », *Journal of Infrastructure Systems*, vol. 15, n°1, 2009, p. 12–20.
- [WSAWG 2002] WEB SERVICES ARCHITECTURAL WORKING GROUP WSAWG, *Web Services Glossary*, W3C, 2002.
- [Zhao et Wu 2011] Y. ZHAO et J. WU, « Towards approximate event processing in a large-scale content-based network », *Proceedings of the 31st International Conference on Distributed Computing Systems (ICDCS)*, IEEE, p. 790–799, 2011.
- [Zheng et al. 2011] Y. ZHENG, Y. LIU, J. YUAN et X. XIE, « Urban computing with taxicabs », *Proceedings of the 13th international conference on Ubiquitous computing*, ACM, p. 89–98, 2011.



— “Family”

Extrait de “Ellyn’s Elements of Style”

07/08/2010

COLOPHON

UN HOMMAGE À “THE ELEMENTS OF TYPOGRAPHIC STYLE”

Cette thèse a été typographiée avec $\text{\LaTeX} 2_{\epsilon}$ avec les polices de caractères Hermann Zapf’s *Palatino* et *Euler* (fontes Type 1 PostScript *URW Palladio L* et *FPL*). Les algorithmes sont typographiés avec la police *Bera Mono*, développée à l’origine par Bitstream, Inc. sous le nom “Bitstream Vera” (fonte Type 1 PostScript rendue disponible par Malte Rosenau et Ulrich Dirr.).

Ce style typographique, développé par André Miede, est inspiré du remarquable livre “*The Elements of Typographic Style*” par Robert Bringhurst [[Bringhurst 2008](#)]. Il est disponible pour \LaTeX via CTAN dans le style “`classicthesis`”. Son adaptation pour \LyX par Nick Mariette est disponible dans le style “`classic-thesis-for-lyx`”.

Version finale du 20 février 2018 à 9:12.