Contribution à l'agilité des systèmes d'information

Jannik Laval

▶ **To cite this version:**

Jannik Laval. Contribution à l'agilité des systèmes d'information. Génie logiciel [cs.SE]. Université Lyon 2 Lumière, 2020. tel-02928813

HAL Id: tel-02928813
https://hal.science/tel-02928813

Submitted on 2 Sep 2020

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.

École Doctorale : Informatique et Mathématiques de Lyon (InfoMaths)

Contribution à l'agilité des systèmes d'information

Habilitation à Diriger des Recherches

présentée et soutenue publiquement le 28-08-2020

par

Jannik LAVAL

Composition du jury

Rapporteurs :	Pr. Mireille Blay-Fornarino	Université Nice Sophia Antipolis
	Pr. Salah Sadou	Université de Bretagne Sud
	Pr. Loïc Lagadec	ENSTA Bretagne
Examinatrices :	Pr. Marianne Huchard	Université de Montpellier
	Mcf, Hdr Christelle Urtado	IMT Mines Alès
Garant :	Pr. Yacine Ouzrout	Université Lumière Lyon 2

DISP Lab — EA 4570
Laboratoire Décision et Information pour les Systèmes de Production

Remerciements

Ce dossier d'Habilitation à Diriger des Recherches rend compte du travail scientifique d'une première étape de ma carrière d'enseignant-chercheur. Cette synthèse est le résultat d'une dynamique professionnelle et personnelle. Aussi, je souhaite remercier ici toutes les personnes qui ont rendu ce travail possible et m'ont aidé à le finaliser.

Tout d'abord, je souhaite adresser mes remerciements aux membres du jury, pour avoir alloué de leur précieux temps à l'évaluation de ce manuscrit : Pr. Mireille Blay-Fornarino, Université Nice Sophia Antipolis, Pr. Salah Sadou, Université de Bretagne Sud, Pr. Loïc Lagadec, ENSTA Bretagne pour avoir accepté d'assurer le rôle de rapporteurs ; Pr. Marianne Huchard, Université de Montpellier, Dr. Christelle Urtado, IMT Mines Alès pour avoir accepté d'être examinatrices.

Je tiens à remercier Yacine Ouzrout, qui a accepté d'être le garant de ce travail. Il m'a accompagné et accordé le temps nécessaire à mes questions. Merci également à Vincent Cheutet et Néjib Moalla de m'avoir apporté leurs réflexions pour m'aider à faire aboutir ce projet.

L'environnement dans lequel je m'épanouis m'a permis d'envisager sereinement la réalisation de ce projet. Aussi, je tiens à remercier toute l'équipe du laboratoire DISP, et particulièrement les personnes qui me font confiance pour les accompagner dans les différents projets ou supervisions d'étudiants : Yacine Ouzrout, Néjib Moalla, Vincent Cheutet, Valérie Botta-Genoulaz, Aicha Sekhari mais également tous les enseignants-chercheurs du laboratoire avec qui nous partageons nos différentes idées. Merci à Nawel Amokrane, de l'entreprise Berger Levrault pour tout le parcours effectué grâce à notre collaboration, dont les résultats sont présentés dans ce manuscrit.

Avant d'être affecté à l'Université Lumière Lyon 2, j'ai été en postdoc au Laboratoire Bordelais de Recherche en Informatique (LABRI), et ai effectué 3 ans à Mines Douai. Je tiens à remercier mes plus proches collègues. À Bordeaux, merci à Xavier Blanc de m'avoir fait confiance et de m'avoir soutenu dans mes démarches de recherches. À Douai, merci à Noury Bouraqadi et Luc Fabresse de m'avoir intégré dans leur équipe ; merci à Anthony Fleury pour tous les bons moments passés.

Un remerciement spécial à Mathieu Vermeulen pour son énergie, ses idées, ses passions, son partage. Les projets que nous menons, professionnels comme personnels, sont toujours un plaisir, une réussite et une avancée dans nos travaux de recherche.

Ce travail n'aurait jamais été possible sans l'ensemble des doctorants, postdocs et étudiants de master. Je leur adresse mes remerciements chaleureux. Nous avons réalisé un beau travail d'équipe.

J'ai commencé la recherche à la suite d'une rencontre avec un de mes enseignants, passionné d'informatique, ayant une volonté de transmettre et de partager sans commune mesure. Sa capacité à mobiliser et à mener ses travaux de recherche m'a beaucoup appris durant mes années de thèse. Un grand merci à Stéphane Ducasse, Directeur de Recherche INRIA, qui m'a fait découvrir la recherche et qui a cru en mes compétences il y a un peu plus de 10 ans.

Merci à la communauté ESUG (European Smalltalk User Group), avec qui je partage depuis toutes ces années de nombreuses idées, et cette passion commune. Merci aux membres de la communauté, notamment Loïc Lagadec, Anne Etien, Cédrick Béler, Philippe Cellé de m'avoir donné le dernier déclic pour réaliser ce travail d'Habilitation à Diriger des Recherches.

Merci à ma famille, à tous leurs encouragements durant ces années, à leur confiance dans mes réalisations, à tous ces moments passés à parler de mes travaux. Un grand merci à Babette Perrin pour sa relecture attentionnée.

Enfin, un immense merci à Audrey, pour avoir accepté et supporté mon investissement dans mes travaux, pour ses conseils, sa bienveillance à tout instant. Merci d'être présente dans tous les moments de la vie.

Un grand merci à tous ceux qui m'ont permis d'aboutir à ce résultat.

À Éléa, Lilio, Cléophée.

À Audrey.

Table des matières

1 Introduction **1**

1.1 Contexte . . . 2

1.2 L'agilité du système d'information... . . . 2

1.3 ... et sa structure logicielle . . . 3

1.4 Les systèmes de systèmes . . . 4

1.5 Positionnement . . . 5

1.6 Problématique et Contributions . . . 7

1.7 Organisation du Manuscript . . . 9

2 Axe 1 : Contribution à la construction d'architectures de systèmes d'information interopérables **11**

2.1 Contexte . . . 12

2.2 Profil de capacité d'architecture d'entreprise pour la réutilisation de logiciels . . . 14

2.2.1 Objectifs de recherche . . . 14

2.2.2 Le framework EACP . . . 15

2.2.3 Métamodèle de qualification et d'exploitation des logiciels pour réutilisation 15

2.2.4 Processus 1 : qualification . . . 17

2.2.5 Processus 2 : référentiel EAKR . . . 18

2.2.6 Processus 3 : exploitation . . . 18

2.2.7 Expérimentation . . . 19

2.2.8 Synthèse . . . 20

2.3 Approche centrée donnée pour une collaboration des systèmes d'information . . . 22

2.3.1 Objectifs de recherche . . . 22

2.3.2 Contribution . . . 22

2.3.3 Expérimentation . . . 26

2.3.4 Synthèse . . . 26

2.4 Bilan de l'axe et perspectives . . . 27

2.5 Synthèse des contributions . . . 28

3 Axe 2 : Contribution à la supervision des systèmes d'information distribués **29**

3.1 Contexte . . . 30

3.2 Supervision des échanges de données d'un système d'information distribué 32
3.2.1 Objectifs de recherche 32
3.2.2 Termes et vocabulaire 33
3.2.3 Le framework Pulse 34
3.2.4 Métamodèle Pulse 36
3.2.5 Requêtes et indicateurs de supervision du système 38
3.2.6 Application à l'analyse de l'interopérabilité 39
3.2.7 Mise en oeuvre du framework 41
3.2.8 Synthèse 42
3.3 Gestion des vulnérabilités de la cyber-sécurité à l'aide de réseaux de neurones 44
3.3.1 Objectifs de recherche 44
3.3.2 Le framework 46
3.3.3 Synthèse 52
3.4 Bilan de l'axe et perspectives 53
3.5 Synthèse des contributions 53

4 Axe 3 : Contribution à l'outillage pour l'évolution des systèmes cyberphysiques 55
4.1 Contexte 56
4.2 Plateforme de simulation de CPS : le cas d'une architecture multirobot 58
4.2.1 Objectifs de recherche 58
4.2.2 L'architecture utilisée 59
4.2.3 La configuration d'une simulation 60
4.2.4 Structure d'un robot 60
4.2.5 Discussion 61
4.2.6 Synthèse 63
4.3 Méthodologie de tests pour les CPS : application aux robots mobiles 64
4.3.1 Objectifs de recherche 64
4.3.2 Framework de tests 64
4.3.3 Expérimentation 67
4.3.4 Synthèse 69
4.4 Méthodologie de co-conception logiciel / FPGA 71
4.4.1 Objectifs de recherche 71
4.4.2 Le Métamodèle OoRC 72

4.4.3 DSL simplifiée . . . 73
4.4.4 Réutilisation du code VHDL hérité . . . 73
4.4.5 Environnement de co-conception . . . 74
4.4.6 Expérimentation . . . 75
4.4.7 Synthèse . . . 75
4.5 Bilan de l'axe et perspectives . . . 76
4.6 Synthèse des contributions . . . 77

5 Bilan et Perspectives **79**
5.1 Bilan . . . 80
5.2 Transfert . . . 83
5.2.1 Transfert aux étudiants . . . 83
5.2.2 Transfert aux entreprises . . . 83
5.2.3 Projets nationaux et internationaux . . . 84
5.3 Perspectives . . . 85
5.3.1 Perspectives 1 : vers l'auto-construction des systèmes d'information interopérables . . . 85
5.3.2 Perspective 2 : vers une surveillance de la dynamique des systèmes cyberphysiques . . . 87
5.3.3 Perspective Axe 3 : vers l'aide à l'évolution des systèmes d'information flexibles et ouverts . . . 88

Production scientifique **95**

Bibliographie **103**

CHAPITRE 1

Introduction

Contents

1.1 Contexte **2**
1.2 L'agilité du système d'information... . . . . . **2**
1.3 ... et sa structure logicielle **3**
1.4 Les systèmes de systèmes **4**
1.5 Positionnement **5**
1.6 Problématique et Contributions **7**
1.7 Organisation du Manuscript **9**

Grimper est secondaire, ce qui est important c'est l'homme qui naît au cours de l'ascension.

[Gaston Rébuffat, La montagne est mon domaine]

En quelques mots...

Ce chapitre introduit le contexte de cette Habilitation à Diriger des Recherches. Après une définition de l'agilité des systèmes d'information, j'aborde sa structure logicielle et notamment le concept de Système de Systèmes. Je termine ce chapitre avec un positionnement de mes travaux actuels.

1.1 Contexte

Les systèmes d'information révolutionnent et transforment les activités des organisations. Ils sont essentiels à la mise en œuvre de la stratégie de toute entreprise pour atteindre ses objectifs. O'Brien et Marakas [O'Brien 2006] définissent un système d'information comme un ensemble organisé de ressources de divers types. C'est une combinaison de personnes, de logiciels, de réseaux de communication, de matériel, de données, de politiques et de procédures qui stocke, récupère, transforme et diffuse diverses informations dans une organisation [Al-Mamary 2014].

Dans le monde d'aujourd'hui, l'entreprise a besoin d'un système d'information pour suivre toutes ses activités. Ces dernières, représentées par des processus métiers, peuvent être assistées par des outils logiciels et des applications d'entreprise dont l'objectif est l'automatisation des échanges. Elle peut être une source de force concurrentielle si elle permet à l'entreprise d'innover ou de réaliser des tâches plus rapidement que les entreprises concurrentes. Shackelford et al. [Shackelford 2006] affirment que "les spécialistes des systèmes d'information se concentrent sur l'intégration de solutions informatiques et de processus métiers, pour répondre aux besoins d'informations des entreprises, leur permettant d'atteindre leurs objectifs de manière efficace et efficiente".

Pour autant, le système d'information n'est souvent pas en parfait alignement avec les processus de l'entreprise. D'une part, les besoins de l'entreprise évoluent continuellement et le système d'information doit s'adapter à ses exigences. D'autre part, l'écart entre les spécifications et la réalisation représente l'une des causes principales d'échecs des projets informatiques [Int'l 2015].

De manière à répondre à ces écarts, l'agilité est considérée comme une caractéristique clé des systèmes d'information [Rasouli 2017, Tarafdar 2017]. En même temps, le système d'information dépend d'applications logicielles de plus en plus nombreuses et interconnectées, par conséquent plus complexes à transformer.

1.2 L'agilité du système d'information...

L'agilité suit une philosophie progressive et pragmatique. Le mot agile est dérivé du latin "agere" qui signifie "agir". Selon le Larousse, il est défini comme "Qui a de l'aisance et de la promptitude dans les mouvements du corps ; souple, alerte".

Même si la définition de l'agilité est ambivalente dans la littérature scientifique et l'industrie, elle peut être définie comme : "la disponibilité continue d'une entité à adopter rapidement ou intrinsèquement, de manière proactive ou réactive, le changement, grâce à des composants de haute qualité, simplistes et économiques et les relations avec son environnement" [Conboy 2004].

Dans le contexte des systèmes d'information, l'agilité traite généralement les problèmes de technologie de l'information, c'est-à-dire le développement de systèmes d'information agile où l'informatique est présente pour détecter, diagnostiquer et répondre en temps réel aux changements [Baskerville 2011, Lowry 2016].

Dans notre article [2], nous référençons plusieurs définitions de l'agilité, partiellement appliquées aux systèmes d'information. Ces définitions se concentrent soit sur l'agilité des processus métier, soit sur l'agilité de développement des systèmes d'information. Ces définitions proposées de l'agilité ne sont pas complètes, mais plutôt complémentaires.

Sur la base de ces définitions, nous pouvons identifier deux visions de l'agilité du système d'information. La première vision est l'agilité basée sur la capacité intrinsèque du système d'information à répondre à l'entreprise. C'est la capacité du système d'information à répondre aux besoins de l'entreprise même s'il y a des changements dans l'environnement. La deuxième vision est la capacité du système d'information à évoluer afin d'intégrer de nouveaux besoins et de nouvelles technologies pour rester compétitif. Nous l'appelons l'agilité dans le processus d'amélioration.

En incluant les deux visions, nous proposons la définition suivante de l'agilité du système d'information : **"Un système d'information est agile à partir du moment où il s'approprie un ensemble de processus et de stratégies qui implique progressivement l'utilisateur, afin d'acquérir la capacité d'adaptation de manière flexible aux changements de l'activité et continue à atteindre ses objectifs, en diminuant le temps de réponse, malgré la pression et les turbulences dans un environnement imprévisible"** [2].

1.3 ... et sa structure logicielle

Le système d'information est porté par des applications informatiques interconnectées. Ses constituants sont capables de partager et d'échanger des informations sans dépendre d'un acteur particulier et de fonctionner indépendamment les uns des autres, nous les qualifions de systèmes interopérables [Chen 2008]. L'interopérabilité est une propriété qui définit la capacité de deux systèmes ou composants ou plus à échanger des informations et à utiliser les informations échangées [Geraci 1991]. Lorsqu'il est établi entre des systèmes d'information communicants, il garantit une productivité et une efficacité croissantes des processus inter et intra-entreprise. Ceci est techniquement possible par l'automatisation des échanges d'informations basée sur l'utilisation de formats d'échange partagés et de protocoles de communication appropriés. Pour réaliser l'interopérabilité des données [Chen 2006], les entreprises construisent des architectures flexibles, évolutives et à couplage lâche, telles que des architectures orientées service et événementielles (SOA, EDA).

L'architecture orientée service (SOA) est une de ces approches utilisées pour l'intégration de plates-formes, de protocoles et de systèmes hérités [Da Xu 2011]. Elle se caractérise par la simplicité, la flexibilité et l'adaptabilité. Les chercheurs considèrent que les fonctionnalités SOA apportent des avantages significatifs aux organisations, car elles leur permettent de concevoir de nouvelles applications de manière dynamique et de répondre aux besoins changeants de l'entreprise. En tant qu'approche architecturale, elle décompose les applications métiers en processus individuels et en services. Ces derniers représentent les fonctionnalités de l'entreprise [Iacob 2009]. Ils peuvent être recomposés pour créer des applications alternatives. Ils peuvent également être exposés à d'autres systèmes permettant à différentes applications de réutiliser des parties communes.

Les nouvelles évolutions technologiques qui arrivent régulièrement pour répondre à différents besoins métiers, techniques (architecture plus fiable, cloud, CPS...) ou qualitatifs (intégration sémantique, sécurité...) doivent être considérées par les entreprises [Panetto 2016,Li 2015]. Actuellement, les paradigmes tels que l'intelligence artificielle, la Blockchain, l'Internet des Objets (IoT) encouragent la transformation des systèmes d'information vers des infrastructures ouvertes. Au-delà d'une "simple" évolution logicielle, cette transformation implique des évolutions architecturales, matérielles, stratégiques, et aussi des changements de processus et de sécurité [Kamble 2019]. Malgré ces évolutions des applications logicielles, la complexité continue d'augmenter notamment en déplaçant le problème d'architecture vers des problèmes de déploiement [Avritzer 2018].

Non seulement les entreprises doivent faire face à l'arrivée de ces technologies et apprendre à les utiliser, mais également elles doivent les intégrer dans leur patrimoine logiciel. Ce patrimoine est constitué de nombreuses applications utilisées, fiables et stables. Certaines sont en phase de maturité, d'autres sur le déclin. Cela implique le développement d'une coexistence entre ces applications et les nouvelles applications développées à partir des nouvelles technologies [Zijden 2019]. Ainsi, en plus de l'adoption de nouvelles technologies, les entreprises font face au besoin d'intégrer voire de transformer leurs anciennes applications [Mateos 2004, Hussain 2017].

Dans cet écosystème logiciel, il est important de considérer les lois de l'évolution logicielle [Lehman 1996]. Parmi les huit lois énoncées, quatre d'entre elles sont particulièrement intéressantes dans le cadre de l'évolution du logiciel :

— Croissance continue : un logiciel doit augmenter son nombre de fonctionnalités pour intéresser les utilisateurs.

— Changement continu : un logiciel doit être adapté continuellement pour répondre aux nouveaux besoins des utilisateurs.

— Complexité croissante : la complexité augmente avec l'évolution du logiciel.

— Qualité décroissante : la qualité d'un logiciel décroît avec son évolution.

Ces lois montrent que le logiciel est dépendant de son environnement. Les deux premiers points montrent que pour maintenir un logiciel en vie, la satisfaction et l'intérêt de l'utilisateur sont centraux. Les deux points suivants montrent que la maintenance du logiciel se complexifie avec le temps. Il est ainsi nécessaire d'allouer du temps pour réduire la complexité et maintenir la qualité. Ces lois montrent également qu'une application logicielle est définie comme un système vivant [Retaillé 2011].

Le cycle de vie d'une application logicielle est maintenant bien établi [ISO Central Secretary 2015]. Dans ce manuscrit, il sera simplifié en trois phases : la conception, l'exploitation, la maintenance, représentant les trois axes de recherche développés. La conception correspond à la création de l'application jusqu'à sa première mise à disposition auprès des utilisateurs. L'exploitation correspond à la phase d'utilisation par les utilisateurs, et d'amélioration (la correction de bugs et l'ajout de fonctionnalités) permettant de répondre au mieux aux besoins métiers. La maintenance correspondant à une période où l'amélioration devient plus lourde pour différentes raisons. Ces raisons peuvent être la perte de la connaissance, le changement de technologie, la complexité importante du logiciel. Lorsque la maintenance n'est plus viable pour l'entreprise, la migration vers un nouveau logiciel est étudiée.

1.4 Les systèmes de systèmes

Une approche Système de Systèmes permet d'appréhender ces problématiques tout en répondant à la contrainte de l'agilité des systèmes d'information.

On appelle système de systèmes (SdS) le résultat de l'intégration de plusieurs systèmes indépendants, interopérables, et interconnectés dans le but de faire émerger de nouvelles fonctionnalités [Maier 1998]. Ces systèmes indépendants sont appelés des systèmes constituants du système de systèmes. Ils peuvent être des systèmes hérités que l'on encapsule comme des boîtes noires, ou des technologies destinées à être utilisées de manière connectée, par exemple les systèmes cyberphysiques (CPS) (Figure 1.1).

Les systèmes de systèmes sont ainsi composés d'artefacts concurrents et distribués. Cela engendre de nouveaux défis sur tout le cycle de vie du système : de la conception en passant par l'exploitation

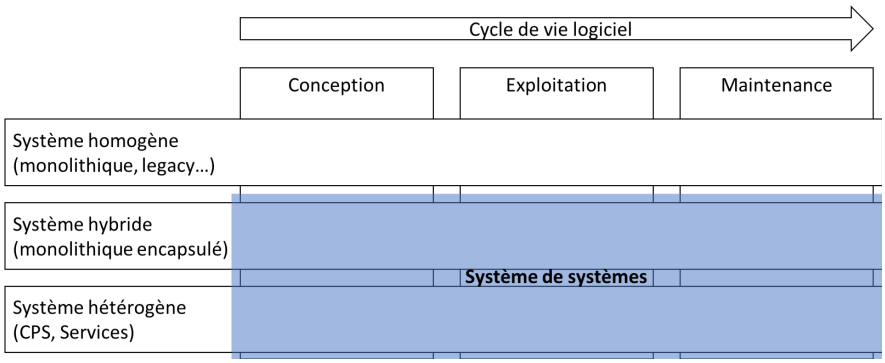

FIGURE 1.1 – Positionnement d'un système de systèmes

des systèmes, et jusqu'à leurs maintenances [Dahmann 2015].

La conception de système de systèmes présente des différences importantes par rapport à la conception de systèmes classiques [Laboratoire LIUPPA 2014]. Chaque système constituant a une indépendance opérationnelle, et peut être distribué géographiquement. Par ailleurs, les exigences du SdS sont parfois en contradiction avec les exigences des systèmes constituants en termes de performances, sécurité ou tolérance aux pannes.

Lors de son exploitation, le SdS n'est pas figé. À tout moment, de nouveaux systèmes peuvent être ajoutés ou supprimés. De là, il peut apparaître des comportements émergents. Ajoutons à cela que chaque système constituant a une indépendance managériale, c'est-à-dire qu'il peut subir des transformations ayant une incidence sur l'ensemble du SdS. La dynamicité d'un SdS fait qu'il est difficile d'anticiper son comportement global au moment de sa conception. Ainsi ses propriétés non fonctionnelles (notamment la qualité ou la sécurité) sont difficiles à prévoir [Ceccarelli 2016].

Lors de leur maintenance, les SdS doivent être capables de s'adapter dynamiquement pour continuer d'assurer leur mission malgré les évolutions internes ou externes qu'ils subissent. Cette dynamicité rend le système difficile à maintenir, car chaque constituant est maintenu indépendamment des autres [Ceccarelli 2016].

Ainsi, les systèmes d'information et leurs interconnexions peuvent être considérés comme des systèmes de systèmes. Dans la suite du manuscrit, j'utilise les termes "système d'information", "système distribué" ou "CPS". Tous ces systèmes se retrouvent dans le cadre de système de systèmes soit en tant que système hybride (notamment pour certains systèmes d'information), soit en tant que système hétérogène (Figure 1.1).

1.5 Positionnement

Dans ce contexte, mes travaux visent à proposer des modèles et environnements informatiques pour favoriser l'agilité des systèmes d'information et comprendre la complexité des systèmes de systèmes.

Ma mobilité géographique a eu des répercussions sur mes thématiques scientifiques. J'illustre, en Figure 1.2, l'évolution chronologique de mes thématiques. Après ma thèse en maintenance logicielle, mes travaux se sont diversifiés dans le domaine des systèmes d'information distribués. D'une part,

entre 2012 et 2015 en abordant des systèmes multirobots, considérés comme des systèmes de systèmes [Cadavid 2020b]. D'autre part, à partir de 2015, mes travaux se sont élargis à l'ensemble du cycle de vie des systèmes d'information distribués.

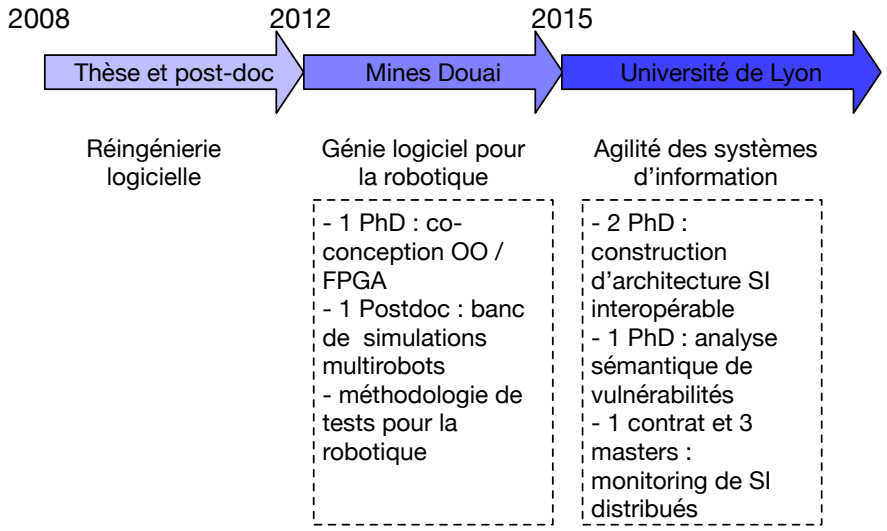

FIGURE 1.2 – Évolution chronologique de mes travaux de recherche

2008-2012 - Thèse et Postdoctorat : J'ai rédigé ma thèse dans le domaine de la réingénierie logicielle, mes travaux ont été fortement influencés par les lois de l'évolution logicielle. La maintenance représente plus de 80% du coût de développement d'un logiciel [Davis 1995, Erlikh 2000]. De plus, maintenir un logiciel nécessite bien souvent de modifier une partie du code source d'un ensemble (le logiciel) trop complexe pour être compris dans sa totalité. Purushothaman et Perry [Purushothaman 2005] montrent que 40% des bogues sont introduits en corrigeant d'autres bogues, faute de pouvoir connaître l'ensemble de leurs implications. Ces études montrent que la phase de maintenance est une période critique qui mérite une prise en charge importante. Améliorer les conditions de maintenance est possible grâce à une architecture modulaire. Cela permet de limiter l'impact d'une modification sur les autres éléments. Je propose dans mes travaux de thèse une approche pour aider les développeurs à rendre modulaires leurs architectures logicielles monolithiques.

2012-2015 - Parcours en Génie Logiciel pour la Robotique : Alors que durant ma thèse et mon postdoctorat, j'ai étudié les systèmes logiciels, la suite de mes travaux consiste à étudier les systèmes d'information distribués dans le domaine de la robotique mobile.

Les robots mobiles intègrent des capacités de perception, de communication et d'adaptation à de nombreuses situations. Ils représentent une forme de CPS (Cyber Phisical System). Ils répondent à des exigences importantes en termes de robustesse, d'ergonomie et de sécurité. Les tâches sont exécutées avec toujours plus d'autonomie, mais aussi en coordination avec d'autres robots autonomes ou des systèmes cyberphysiques. Il en résulte une forte croissance de l'architecture logicielle et de sa complexité.

Développer des architectures robotiques nécessite de combiner le développement du matériel (assemblage et synchronisation des différents éléments du robot) et le développement logiciel. Le logiciel

doit prendre en compte les contraintes temps réel, un nombre de capteurs et d'actionneurs toujours plus important, et l'interaction avec l'environnement. Le cycle de développement d'une application robotique est contraignant : il s'effectue sur une machine de développement (un PC). Puis, l'application est déployée et testée sur le robot cible. Les corrections nécessitent de revenir à la machine de développement et de refaire un cycle de déploiement. Les pratiques actuelles rendent le processus de développement d'applications robotiques plus rigide et plus long que pour une application informatique "standard". Ainsi, le développement d'architectures logicielles correctes et fiables est un enjeu majeur. J'ai donc abordé des problématiques liées à l'aide au développement en utilisant une méthodologie de test et en développant la simulation robotique. Cela permet d'accompagner notamment le déploiement de code sur des architectures différentes, notamment en réalisant de la co-conception que nous avons mise en oeuvre entre l'architecture orientée objet et une architecture FPGA.

2015-... - Parcours en Agilité des Systèmes d'Information : Depuis 2015, au laboratoire DISP, je travaille dans le domaine de l'agilité des systèmes d'information, avec une considération du métier et de la sémantique des flux d'informations.

Ici, le travail est appliqué sur des systèmes d'information d'entreprise pour lesquels les besoins en matière de partage, d'échanges et de valorisation des informations sont en constante augmentation et représentent désormais un enjeu majeur. Il est, par conséquent, essentiel de concevoir des « plateformes » capables d'apporter des réponses à la rationalisation et à la simplification des échanges de données entre applications logicielles et avec le monde extérieur.

Les architectures orientées service et les architectures événementielles (SOA, EDA) sont matures et largement utilisées pour répondre à ces besoins. Ces architectures se caractérisent par la flexibilité et le couplage lâche des sous-systèmes qui les composent (c.-à-d. services, applications, SI...) et reposent sur plusieurs moyens [Hohpe 2004] pour acheminer les données au sein de ce réseau de systèmes communicants hétérogènes [Moalla 2012]. À ce stade de maturité, nous observons que ces échanges de données sont opérationnels et répondent aux exigences d'interopérabilité entre systèmes hétérogènes [Leal 2018]. Pour autant, l'évolution des différentes technologies impacte de façon quasi systématique les échanges de données mis en place pour assurer l'interopérabilité. D'où la nécessité de construire des architectures d'échange flexibles et évolutives et de suivre l'évolution de ces échanges.

Le travail de recherche que je mène actuellement prend ces considérations en compte selon 2 grands axes : l'intégration de nouvelles fonctionnalités dans des applications d'entreprise provenant d'applications open source au moment de la conception, et le monitoring de systèmes d'information pour en assurer l'exploitation et l'évolution.

1.6 Problématique et Contributions

L'Habilitation à Diriger des Recherches permet de faire une synthèse des travaux effectués en début de carrière et de positionner ses futures recherches. Les travaux présentés dans ce manuscrit, abordés en collaboration avec des scientifiques de différents domaines (génie logiciel, robotique, système d'information d'entreprise...) ont tous un point commun : ma vision est orientée génie logiciel, et influencée par une approche cycle de vie logiciel.

Alors que les lois de l'évolution logicielle portent sur la vie du code source, l'hypothèse sur laquelle reposent mes travaux est que ces lois ne sont pas limitées au génie logiciel, mais influencent

la vie du système d'information. Ainsi, j'aborde la problématique du cycle de vie des systèmes de systèmes avec pour objectif l'agilité des systèmes d'information distribués. Mes travaux de recherche traitent la problématique de la conception, de l'exploitation et de la maintenance de ces systèmes.

Pour répondre à ce défi, j'ai organisé mes travaux selon ces trois axes (Figure 1.3). Dans la phase de conception, nous considérons la création d'un système d'information à partir d'applications existantes depuis les besoins des utilisateurs jusqu'à son implémentation. Dans la phase d'exploitation, nous considérons l'utilisation du système, c'est-à-dire comment nous nous assurons qu'il reste utilisable. Dans cette phase, nous travaillons sur la supervision du système d'information distribué. Dans la phase de maintenance, nous considérons l'outillage permettant de maintenir un système stable, c'est-à-dire de réduire les fautes.

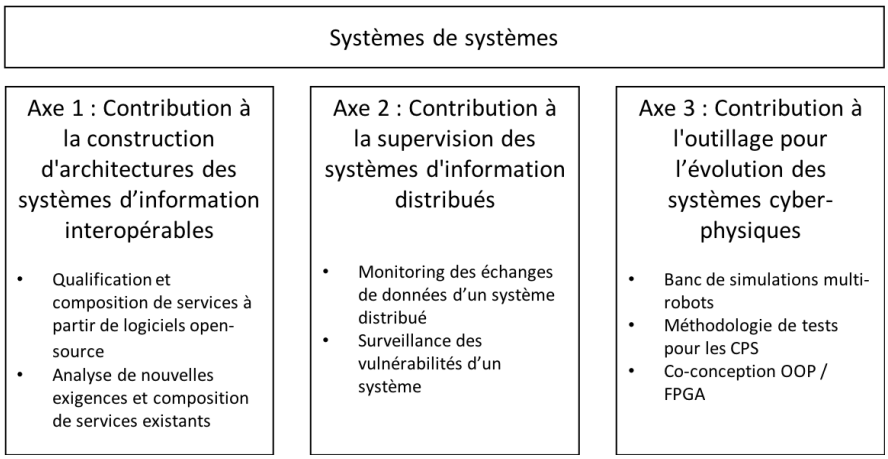

FIGURE 1.3 – Structure du manuscrit

1. Axe 1 : Contribution à la construction d'architectures de systèmes d'information interopérables. Nous considérons la création d'un système d'information distribué à partir d'applications existantes. Ici je développe le besoin des entreprises de faire de la réutilisation d'applications existantes.

   — La thèse d'Abdeladhi Belfadel aborde l'utilisation et la composition de fonctionnalités de logiciels open source de manière à réduire le temps de développement tout en assurant la viabilité du système final. Le framework proposé permet de spécifier le besoin, de lister les fonctionnalités existantes et d'associer les deux pour proposer un système opérationnel.

   — La thèse de Rami Thorkani propose la création de Robotic Process Automation (RPA) à partir des données entrantes et des services existants. Le framework cherche des services permettant de traiter les informations et propose de les assembler.

   Perspectives : vers l'autoconstruction des systèmes d'information interopérables. Pour cela, nous devons aborder l'analyse sémantique des logiciels open source, l'association sémantique avec les besoins, l'auto-orchestration des microservices, l'évolution dynamique des SI par rapport aux besoins.

2. Axe 2 : Contribution à la supervision des systèmes d'information distribués : ici, nous considérons la dynamique d'un système d'information distribué. Le système d'information peut devenir instable ou fragile, et perdre sa fiabilité. Mes travaux ici se divisent en deux parties :

— Le monitoring des systèmes d'information distribués. Le framework proposé récolte les données du système. Les données sont structurées grâce à l'Ingénierie Dirigée par les Modèles. Cela nous permet d'analyser et de proposer des éléments de remédiations. Notre contribution est issue d'une collaboration avec l'entreprise Berger-Levrault.

— La thèse d'Houssem Gasmi propose l'analyse sémantique d'informations provenant de sources externes pour pouvoir améliorer et sécuriser le système d'information. Nous avons choisi d'étudier les vulnérabilités des systèmes en utilisant une approche d'analyse sémantique des bases de données de cybersécurité. Ces analyses permettent la levée d'alertes lorsque le système d'information est mis en danger.

Perspectives : vers une surveillance de la dynamique des systèmes cyberphysiques. Les problématiques à aborder sont la surveillance contextualisée et intelligente et la remédiation automatique, l'exploration automatique de microservices similaires, l'adaptation dynamique de l'architecture, la découverte de fonctionnalités émergentes à partir du monitoring, la gestion de la variabilité et de la résilience du système.

3. Axe 3 : Contribution à l'outillage pour l'évolution des systèmes cyberphysiques. Dans cet axe, j'aborde l'outillage logiciel permettant d'aider à transformer une architecture distribuée. Ces systèmes ont un comportement incertain par définition. Mes travaux adaptent des outils du génie logiciel selon 3 axes :

   — Le suivi du postdoctorat de Zhi Yan propose la réalisation de plateforme de simulation de systèmes multirobots.

   — La mise en place d'une méthodologie de tests pour les systèmes cyberphysiques (robotiques dans mon cadre d'expérimentation).

   — La thèse de Xuan Sang Le propose la co-conception de code source orienté objet avec une architecture FPGA.

   Perspectives : vers l'aide à l'évolution des systèmes d'information flexibles et ouverts. Les perspectives sont d'assurer l'évolution des CPS sans rupture de service, la validation dynamique pour les environnements flexibles (la simulation et les tests dynamiques), rendre l'évolution des SdS explicable et fiable, notamment grâce à l'historisation des comportements.

1.7 Organisation du Manuscript

Le manuscrit est organisé de la façon suivante : la partie 2 (page 11) aborde l'axe 1, la partie 3 (page 29) aborde l'axe 2. L'axe 3 est traité dans la partie 4 (page 55). La partie 5 (page 79) développe les perspectives.

Dans chaque axe, après la présentation des enjeux scientifiques, les différentes approches proposées seront exposées. Chaque axe sera composé d'une partie "contexte", d'une partie "contribution" avec pour chaque sous-partie, un résumé d'un travail de recherche, et d'une partie "conclusion/perspectives" représentant les travaux à venir.

CHAPITRE 2

Axe 1 : Contribution à la construction d'architectures de systèmes d'information interopérables

Contents

2.1 Contexte 12
2.2 Profil de capacité d'architecture d'entreprise pour la réutilisation de logiciels . . . 14
2.3 Approche centrée donnée pour une collaboration des systèmes d'information . . . 22
2.4 Bilan de l'axe et perspectives 27
2.5 Synthèse des contributions 28

La technique résout les problèmes et apporte des satisfactions, mais elle n'est qu'un moyen et reste pauvre si on la sépare de l'esprit qui la guide.

[Gaston Rébuffat]

En quelques mots...

Ce chapitre introduit mes contributions à la construction d'architectures de systèmes d'information interopérables. Partant des exigences jusqu'à l'instanciation du système d'information, nous avons mis en oeuvre des frameworks permettant de modéliser, analyser et instancier des systèmes d'information en partant des besoins. Ces travaux sont supportés par deux thèses. Ils ont été présentés dans les publications [17, 18, 20, 21, 23, 24, 89–91].

2.1 Contexte

Dans les travaux de cet axe, nous mettons en avant la réutilisation de logiciels existants pour construire ou améliorer un système d'information. D'une part, réutiliser les fonctionnalités de logiciels existants au sein du système d'information permet de valoriser les développements antérieurs. Par effet de bord, l'amélioration de ces applications existantes est un moyen efficace de garantir la maintenance, la capitalisation du savoir-faire acquis au fil des ans, la réduction des coûts de développement, et la rationalisation de la transformation numérique. D'autre part, intégrer les fonctionnalités d'applications externes contribue à faciliter le développement des systèmes d'information.

Pour identifier les composants les plus adaptés à réutiliser, Paschali et al. [Paschali 2017] notent qu'aucun processus standardisé n'a été défini et les praticiens utilisent toujours des méthodes ad hoc. En effet, la complexité de l'écosystème logiciel conduit à des difficultés pour rechercher, évaluer, récupérer et connecter des composants logiciels. Hauge et al. [Hauge 2010] identifient la difficulté à sélectionner le bon logiciel avec des caractéristiques ou fonctionnalités spécifiques, en raison d'une évaluation longue et complexe des solutions. Le manque de documentation et l'incertitude sur la qualité des composants logiciels sont les facteurs les plus importants empêchant la réutilisation [Kakarontzas 2010]. L'effort d'adaptation nécessaire pour que la solution sélectionnée s'adapte au contexte dans lequel elle sera utilisée est difficilement estimable et peut entraîner un surcoût.

Dans le manuscrit de thèse d'Abdelhadi Belfadel [Belfadel 2019], nous avons analysé les travaux de recherches existants, par une analyse systématique. Nous avons classé les travaux selon différents domaines de recherche. Le premier domaine de recherche est la description des capacités logicielles pour la réutilisation des logiciels. Nous avons analysé différentes dimensions de la réutilisation de logiciels tels que la réutilisation de logiciels open source, les profils de capacités logicielles et la description des services. Le deuxième domaine de recherche concerne l'impact de l'architecture d'entreprise dans le raffinement et la consolidation des exigences, et le positionnement du cycle de vie du développement logiciel (SDL) dans un processus d'architecture d'entreprise. Nous avons étudié l'importance d'aligner les exigences et l'architecture pour sélectionner les blocs logiciels appropriés à réutiliser pour le développement de nouveaux systèmes d'information d'entreprise. Le troisième domaine de recherche est la gestion des connaissances et les référentiels pour la réutilisation des logiciels. Ce domaine est important pour prendre en compte les différents niveaux de description tels que le niveau organisationnel, le niveau technique ou technologique et la façon dont ces niveaux peuvent être structurés. De cette analyse, nous remarquons qu'actuellement des méthodes ad hoc sont utilisées pour identifier les logiciels ou artefacts les plus appropriés à réutiliser.

Pour pouvoir interopérer des solutions, nous devons analyser le processus métier de chaque système impliqué. Cette analyse doit inclure la qualification des processus afin de pouvoir déterminer l'écart fonctionnel [Ostadzadeh 2015]. Par conséquent, nos travaux s'attachent à améliorer la description des capacités des logiciels selon différents niveaux de description, ainsi que leur exploitation. Notre objectif est de maximiser la réutilisation des logiciels internes et externes d'une organisation en améliorant la découverte et la durabilité des solutions existantes, et en recherchant la compatibilité fonctionnelle et non fonctionnelle la plus élevée.

Parmi les approches visant à identifier les capacités des logiciels, l'ISO 16100, intitulé “”Manufacturing software capability profiling for interoperability” [ISO Central Secretary 2018] propose un profil des capacités logicielles que nous utilisons dans nos travaux. La série ISO 16100 permet l'intégration d'applications en fournissant une spécification d'interface commune. Elle permet de spécifier

une interface qui affiche les capacités des applications et facilite la compréhension des processus. Elle est composée de différents artefacts :

— Action : elle décrit chacune des fonctions pouvant être exécutée.

— Ressource : elle décrit les ressources nécessaires à la réussite de l'exécution d'une action.

— Relation : elle décrit les relations entre les applications impliquées dans l'action.

— Échange : cette partie détaille les éléments nécessaires pour remplir complètement l'interconnexion entre les applications concernées.

Dans ce chapitre, nous abordons deux visions de la problématique illustrées par deux thèses.

Par la qualification des exigences : La section 2.2 décrit nos travaux dans la réutilisation d'applications open source à partir de la description du besoin et de la description des applications logicielles. Ce travail est illustré par la thèse d'Abdeladhi Belfadel [Belfadel 2019], soutenue en décembre 2019. Dans ce travail, nous abordons l'utilisation et la composition de fonctionnalités de logiciels open source de manière à réduire le temps de développement tout en assurant la viabilité du système final. Le framework proposé permet de spécifier les exigences, d'identifier les fonctionnalités existantes et d'associer les deux pour proposer un système opérationnel.

Par l'exploitation des données : les systèmes d'information doivent également gérer les nouveaux flux de données. Les applications doivent pouvoir consommer des données qui n'ont pas la même représentation que la leur. Depuis les années 2000, plusieurs initiatives de recherche ont été lancées pour proposer des approches de résolution des conflits de données. Les approches les plus connues ont été répertoriées dans les ouvrages suivants [Dong 2009, Harman 2007]. La section 2.3 décrit nos travaux dans la réutilisation de composants logiciels dans le cadre de Robotic Process Automation (RPA). Ce travail est illustré par la thèse de Rami Thorkani, dont la soutenance est prévue en 2020. Dans ce travail, nous proposons la création de processus automatisés entre différents systèmes à partir des données entrantes et des services existants. Le framework proposé cherche les services permettant de traiter les informations et propose de les assembler.

2.2 Profil de capacité d'architecture d'entreprise pour la réutilisation de logiciels

2.2.1 Objectifs de recherche

Dans cette section, je décris nos travaux de conception d'un système d'information à partir de la réutilisation de solutions qualifiées, en recherchant la compatibilité fonctionnelle et non fonctionnelle la plus élevée. Ces travaux proposent une approche générique applicable à toutes les solutions logicielles d'une entreprise, mais ont été appliqués sur des solutions orientées service, permettant ainsi de nous concentrer sur les problématiques de qualification et de composition plutôt que d'analyse du code source.

Pour maximiser les opportunités de réutilisation, la première chose à faire est de comprendre et de décrire de manière précise les exigences à satisfaire. Les exigences d'un système sont les descriptions de ce que le système doit faire, les services qu'il doit fournir et les contraintes ou la qualité qu'il doit avoir pour le rendre acceptable et attrayant pour l'utilisateur [Robertson 2012]. La deuxième chose à considérer est le recensement des services existants et réutilisables. Pour découvrir les opportunités de réutilisation et répondre à une préoccupation spécifique, un modèle fonctionnel conforme au modèle d'exigence doit être envisagé. L'ingénierie des exigences [Sommerville 2011] permet de répondre à ce problème.

Nous considérons qu'il est nécessaire d'avoir une représentation de haut niveau des systèmes informatiques et des processus métiers de l'entreprise consommant les services. Ainsi, nous devons considérer une qualification tenant compte des vues métiers, opérationnelles et techniques des logiciels et de leurs services associés. Pour promouvoir une description de la capacité logicielle permettant l'utilisation d'un service dans différents contextes, l'impact de l'architecture sur les exigences doit être explicite et doit pouvoir s'adapter lors de la sélection des logiciels et des services.

En considérant une approche basée sur l'architecture d'entreprise, il est possible d'organiser les différents artefacts de façon à analyser les possibilités de réutilisation pour une organisation et d'assurer la faisabilité d'un système cible. L'architecture d'entreprise est la définition et la représentation d'une vue de haut niveau des systèmes informatiques et des processus métiers des entreprises. Elle offre une abstraction du système pour gérer la complexité des systèmes logiciels [Alebrahim 2017]. Elle permet de déterminer les besoins et les priorités de changement d'un point de vue métier, et d'autre part, d'organiser les différents composants et artefacts techniques et d'évaluer comment une organisation peut en émerger.

Elle peut également être utilisée pour évaluer la transition d'une situation à sa version suivante. Les méthodes et les lignes directrices proposées dans les frameworks d'architecture d'entreprise servent à guider les architectes dans la conception des processus métier et les développeurs dans la création d'applications d'une manière conforme aux objectifs et politiques métier [Lankhorst 2009].

Dans ce contexte, le principal problème de recherche abordé est la gestion de la complexité de l'alignement des exigences avec les capacités logicielles, pour sélectionner les meilleurs composants candidats et construire un nouveau système. Les objectifs de recherche que nous avons abordés sont les suivants :

— Définition d'un profil de capacités logicielles qui décrit les capacités des composants logiciels selon plusieurs angles, dont les points de vue organisationnels, fonctionnels, techniques et tech-

nologiques ainsi que la qualité et les contraintes associées. Pour cela nous avons proposé un profil de capacité basé sur l'ISO 16100 [ISO Central Secretary 2018] et le framework TOGAF [The Open Group 2018].

— Proposition de réutilisation de solutions en envisageant un processus qui inclut l'ingénierie des exigences pour formaliser les exigences et les contraintes du système cible. Nous avons proposé un framework permettant l'évaluation, la découverte et la réutilisation des composants pour servir de blocs dans le nouveau système.

— Définition d'un modèle d'exploitation des profils de capacités logicielles. Nous avons proposé un plan d'exploitation inscrit dans un cycle d'ingénierie structuré en plusieurs phases, inspiré du framework TOGAF. Il permet de produire une architecture à partir des exigences définies et des profils de capacités logicielles sélectionnés.

— Réalisation d'une base de connaissances pour gérer et partager les profils de capacités. Nous avons proposé un référentiel de connaissances de l'architecture d'entreprise. Il s'agit d'une approche sémantique qui fournit une ontologie permettant d'alimenter le framework proposé.

2.2.2 Le framework EACP

Pour répondre à ces objectifs de recherche, nous avons proposé le framework EACP (Enterprise Architecture Capability Profile). Il étend le framework TOGAF [The Open Group 2018] et sa méthode de développement d'architecture (Architecture Development Method - ADM), en l'améliorant avec une représentation fonctionnelle. Il couvre la qualification et la découverte de logiciels orientés service. La Figure 2.1 montre le framework dans sa globalité. Ce dernier se compose de trois processus : le processus de qualification des applications, le processus d'enregistrement dans une base de connaissance, que nous appelons référentiel EAKR et le processus d'exploitation.

2.2.3 Métamodèle de qualification et d'exploitation des logiciels pour réutilisation

Pour pouvoir exploiter ces trois processus, nous avons défini un Métamodèle permettant de structurer les informations d'un service logiciel dans ce que nous avons appelé un profil de capacités. Le Métamodèle proposé est illustré dans la Figure 2.2. Il comporte les spécifications fonctionnelles et non fonctionnelles, la description de l'impact organisationnel d'une fonctionnalité, et il relie les fonctionnalités à leurs composants associés.

Le Métamodèle est composé de 6 packages :

— Package "Organization" : Il décrit l'unité organisationnelle, avec ses buts et objectifs métiers associés qui guident le développement des logiciels existants.

— Package "Architecture Building Blocks" (ABB) : Ce package est construit à partir de la méthode ADM. Les ABB sont divisés en deux niveaux. Le premier niveau décrit le problème métier pour lequel le composant a été développé, ses spécifications d'implémentation, les normes utilisées et les parties prenantes. Le deuxième niveau contient plus de détails et concerne la vision opérationnelle du composant. Il définit la fonction métier, ses attributs et contraintes, les exigences d'interopérabilité des données et des applications et les attributs non fonctionnels connus lors de la conception.

— Package "Solution Building Blocks" (SBB) : Ce package représente la solution physique d'un ABB. Il décrit le composant exposé par le logiciel. Il définit les attributs qualitatifs de l'exé-

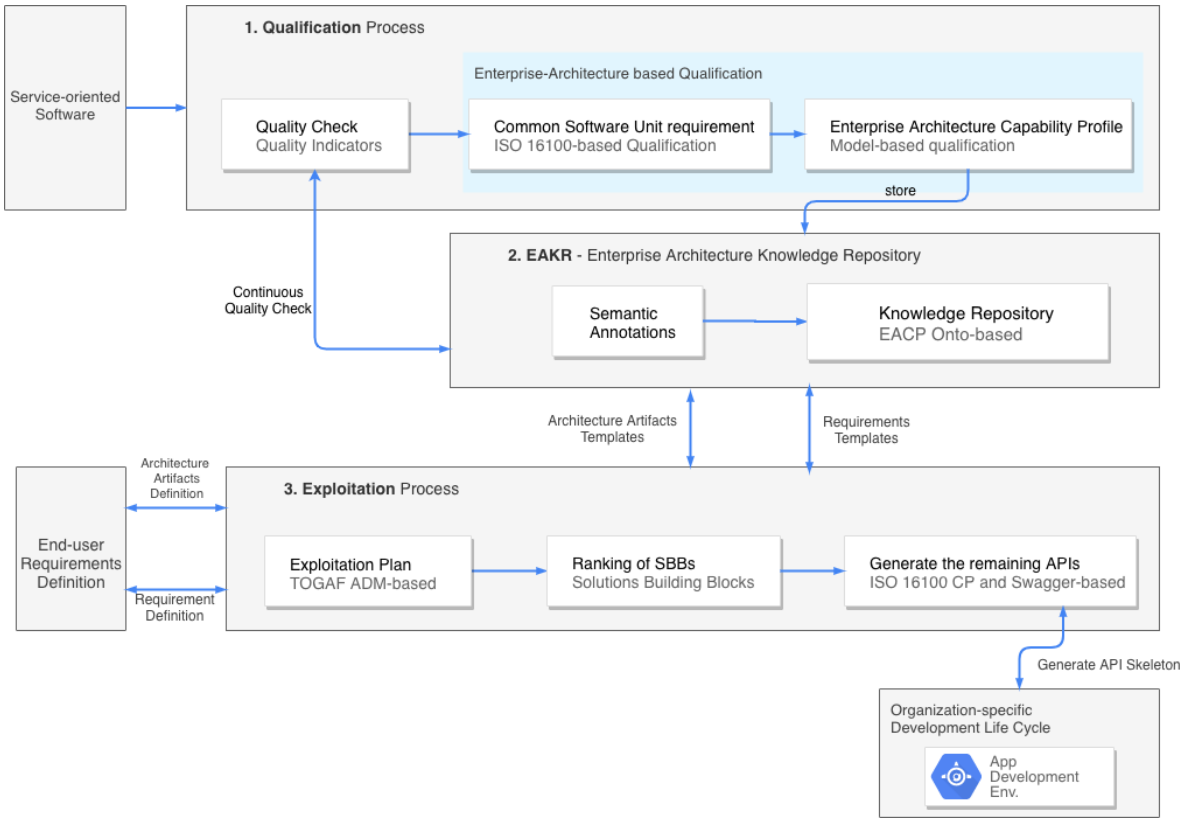

FIGURE 2.1 – Le framework EACP

cution (par exemple, les performances et la disponibilité), et non fonctionnels (par exemple, la sécurité et les paramètres d'authentification ou d'autorisation).

— Package "Application" : Ce package décrit les exigences techniques du logiciel. Il décrit les environnements d'exécution sur lesquels l'application s'exécute et l'API exposée.

— Package "Business process" : Ce package est utilisé en phase d'exploitation et représente l'application métier «à venir» à réaliser. Il est composé des activités qui composent la nouvelle application métier, des rôles et des acteurs concernés par les activités et l'événement déclenchant le processus.

— Package "Requirements" : Ce package est utilisé dans la phase d'exploitation et représente une formalisation des exigences, aidant le développeur ou l'architecte pendant la conception du système d'information. Il est construit à partir de la méthode ADM. Il permet la définition du pilote de projet, la définition des cas d'utilisation et des exigences à différents niveaux (fonctionnel et non fonctionnel, technique, les exigences et contraintes d'interopérabilité).

Ce Métamodèle permet d'instancier des modèles issus de chaque package lors du processus de qualification. Il est utilisé au cours du processus d'exploitation, permettant de trouver le lien entre les nouveaux besoins métiers et les ABB et SBB existants. Les instances de profils de capacités sont enregistrées dans le référentiel de connaissances (référentiel EAKR), pour les réutiliser lors du développement de nouvelles applications métiers.

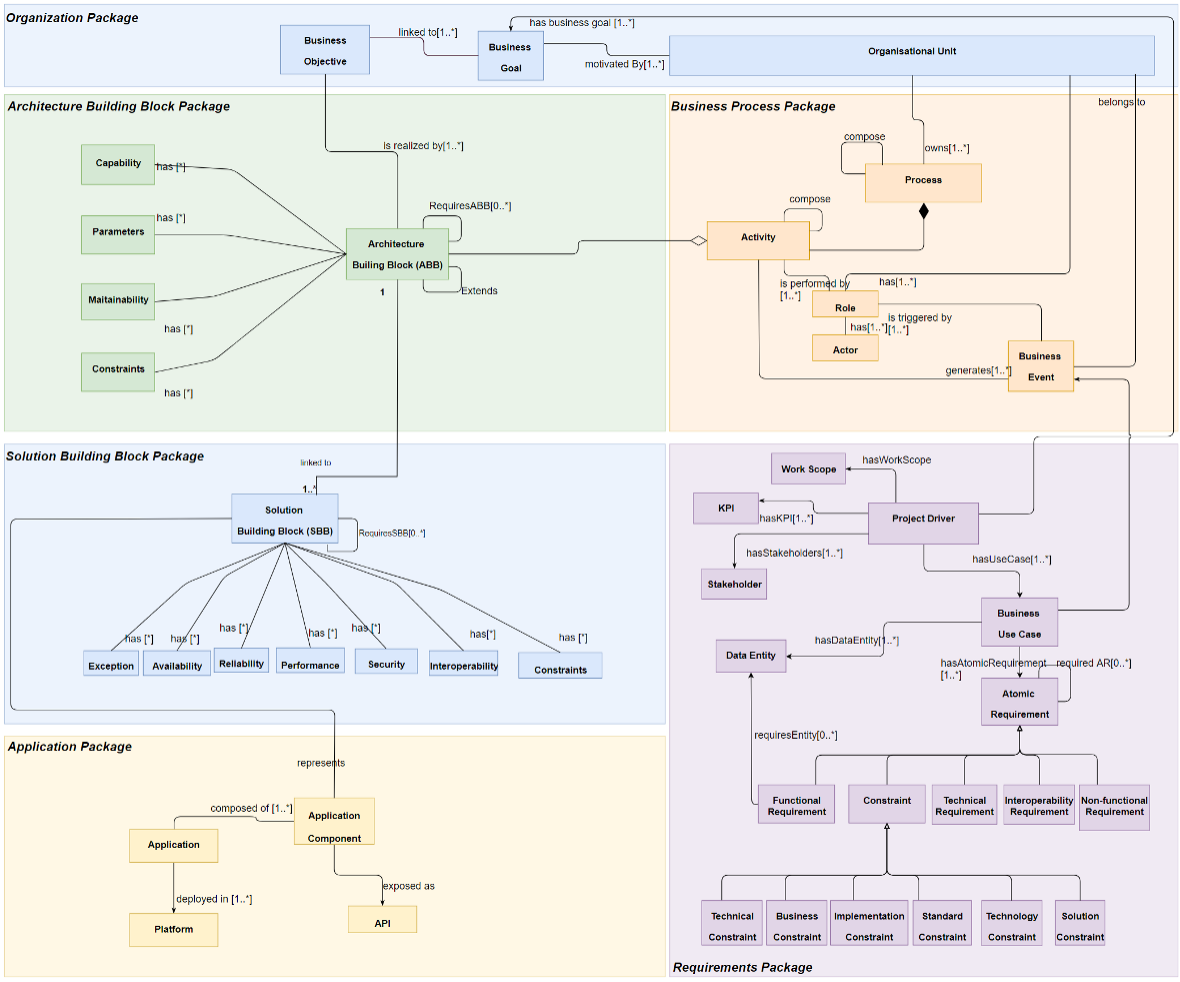

FIGURE 2.2 – Le Métamodèle EACP

2.2.4 Processus 1 : qualification

Ce premier processus permet de qualifier un service exposé d'une solution logicielle. Il prend en entrée un logiciel et ses services associés. La qualification est réalisée sur chaque service et crée un lien entre une fonctionnalité (d'un point de vue métier) et le composant logiciel correspondant (dans ce cas, le service).

Ce processus suit une approche bottom-up, en partant de la définition de la solution (SBB), puis en définissant l'ABB et les buts et objectifs métiers.

Il commence par un contrôle de la qualité du logiciel et de ses services associés. Dans le cas de services ou d'APIs web, les paramètres de qualité de service tels que la disponibilité, le temps de réponse, la fiabilité, le débit, la documentation sont parmi les plus importants [Al-Masri 2007]. Cependant, la conception des services est influencée par l'environnement, le contexte et d'autres décisions prises par les concepteurs de services, ce qui peut entraîner des violations des principes de qualité de service [Ouni 2015].

Nous évaluons les garanties offertes par chaque service qui compose l'application à partir de son analyse et de sa documentation existante (site web, référentiel de code, documents d'installation et de guide d'utilisation ...). Nous avons instancié des indicateurs techniques recommandés par l'ISO 25010 [ISO Central Secretary 2011], étendus avec les meilleures pratiques du Guide d'architecture d'application Microsoft® [Patterns 2009] et du framework TOGAF [The Open Group 2018].

Après ce contrôle qualité, le processus permet la construction du profil de capacité. Ce profil spécifie les propriétés fonctionnelles, le nom de l'activité de la fonctionnalité, les paramètres et les dépendances fonctionnelles. Ce profil décrit l'action effectuée par le service exposé et offre une spécification fonctionnelle en décrivant les entrées et les sorties, les types de données, le nom de classe, le nom de méthode ou de fonction. Il décrit les exigences techniques de l'application en général.

Ce processus génère une qualification du logiciel et des services exposés. Pour chaque service, il en résulte l'impact organisationnel, les détails sur l'environnement d'exécution, les propriétés non fonctionnelles, les attributs de qualité de service et les contraintes correspondantes. Le résultat de ce processus est une instance du Métamodèle présenté dans la section 2.2.3.

2.2.5 Processus 2 : référentiel EAKR

La sortie du processus de qualification est considérée comme une entrée du processus EAKR (Enterprise Architecture Knowledge Repository). Chaque profil EACP est annoté en utilisant des annotations sémantiques liées à une ontologie spécifique. Cette dernière offre un vocabulaire standard pour travailler sur les profils de capacité. Chaque profil est stocké dans le référentiel EAKR, visible dans la Figure 2.1. Un contrôle qualité peut être appliqué sur les solutions déjà qualifiées pour actualiser les indicateurs qualité.

Pour manipuler ce référentiel, nous avons développé une ontologie conçue pour être réutilisée comme modèle de qualification pour la description des composants techniques de l'application. Nous l'avons créée de manière à pouvoir exploiter l'ensemble de notre Métamodèle. Elle a donc été créée à partir de l'ontologie Basic Formal Ontology (BFO) et quatre ontologies spécifiques : l'ontologie TOGAF [Gerber 2010], l'ontologie OWL-S [Martin 2004], l'ontologie BPMN [Natschläger 2011] et l'ontologie IAO [Ceusters 2012].

L'ontologie TOGAF explicite la signification des artefacts d'architecture d'entreprise, y compris les artefacts d'exigence du Métamodèle TOGAF. L'ontologie OWL-S vise à appliquer des capacités de raisonnement au comportement et à l'exécution des services web. L'ontologie BPMN permet la formalisation sémantique de tous les éléments du Métamodèle BPMN 2.0, ainsi que les attributs et conditions définissant comment les éléments peuvent être combinés dans un processus métier. L'ontologie IAO (Information Artifact Ontology) décrit les entités d'information de manière indépendante du domaine.

Ces ontologies ont été intégrées dans l'ontologie EACP à l'aide de l'éditeur d'ontologie Protégé V4.3. La description complète de cette ontologie est disponible dans la thèse d'Abdelhadi Belfadel [Belfadel 2019].

2.2.6 Processus 3 : exploitation

Pour réaliser la conception d'une nouvelle application métier en utilisant les fonctionnalités qualifiées présentes dans le référentiel EAKR, l'architecte passe par un processus de définition des exi-

gences. Il s'inscrit dans un cycle d'ingénierie structuré en plusieurs phases inspirées du framework TOGAF et de la méthode associée ADM (Figure 2.3). Il s'agit de quatre phases commençant par la définition de la vision architecturale et d'exigences, suivie par la phase d'architecture métier, la phase d'architecture du système d'information (données, applications...) et la phase d'architecture technologique. Pour la première phase, les spécifications des exigences utilisent Volere [Robertson 2012] qui est le résultat de nombreuses années de pratique, de conseil et de recherche en ingénierie des exigences et en analyse métier.

Le résultat de ce plan d'exploitation est un ensemble de SBB qui correspond aux besoins exprimés. Ces SBB sont classés en fonction de la qualité de service. Les mesures réalisées au cours du processus de qualification sont utilisées pour le classement. Le classement obtenu permet de sélectionner le meilleur SBB pour chaque exigence, conduisant à la génération d'un BPMN qui consomme les services associés.

Pour les fonctions métier qui ne correspondent à aucun SBB existant, nous proposons un outil qui aide à la génération du squelette de code source pour développer l'API requise et faciliter le développement processus.

2.2.7 Expérimentation

L'expérimentation du framework a été réalisée dans le cadre d'un projet financé par le programme-cadre H2020 de la Commission européenne. Le projet vf-OS (Virtual Factory Open Operating System) visait à développer un système d'exploitation ouvert composé d'un noyau, d'une interface de programmation d'application et d'un middleware spécialement conçus pour l'usine du futur.

Nous avons développé un scénario concernant deux sociétés participant au projet vf-OS. La première entreprise est spécialisée dans la transformation de plastique et la seconde dans la transformation de métaux. Leur principal problème métier est de détecter les opportunités de collaboration métier les plus appropriées parmi les projets clients. À partir d'une description CAO des projets clients, les deux entreprises doivent identifier rapidement l'ensemble des projets qu'elles sont capables de produire.

Le cas d'utilisation proposé vise à accélérer et à maintenir un canal de collaboration entre leurs deux domaines métiers. Les objectifs sont de réduire les coûts et le délai de traitement des devis clients. Actuellement, les clients envoient des demandes de produits à l'une des sociétés sous forme de fichiers CAD / PDF. Ensuite, l'entreprise décompose toutes les caractéristiques du projet (pièces, dimensions, type de surface, type de matière première, etc.) pour comprendre les besoins du client, vérifier la faisabilité du produit et déterminer la pertinence de l'opportunité métier en termes de ROI. Après la décomposition, les deux sociétés mènent une succession de négociations, explorent plusieurs voies pour répondre aux exigences du client et lui présenter la meilleure offre.

Nous avons validé l'approche EACP avec l'implémentation du cas d'utilisation. Nous avons décrit, via notre framework, le processus de description des exigences pour découvrir les services existants et développer l'application métier requise. Une vidéo de ce scénario de cas d'utilisation est accessible [1].

Dans le cas développé, le framework EACP trouve peu de résultats car aucun projet précédent avec ses exigences connexes n'a déjà été introduit et capitalisé ; cela dépend du nombre de solutions

1. https://bul.univ-lyon2.fr/index.php/s/KCCk6a9SmkSHJWz

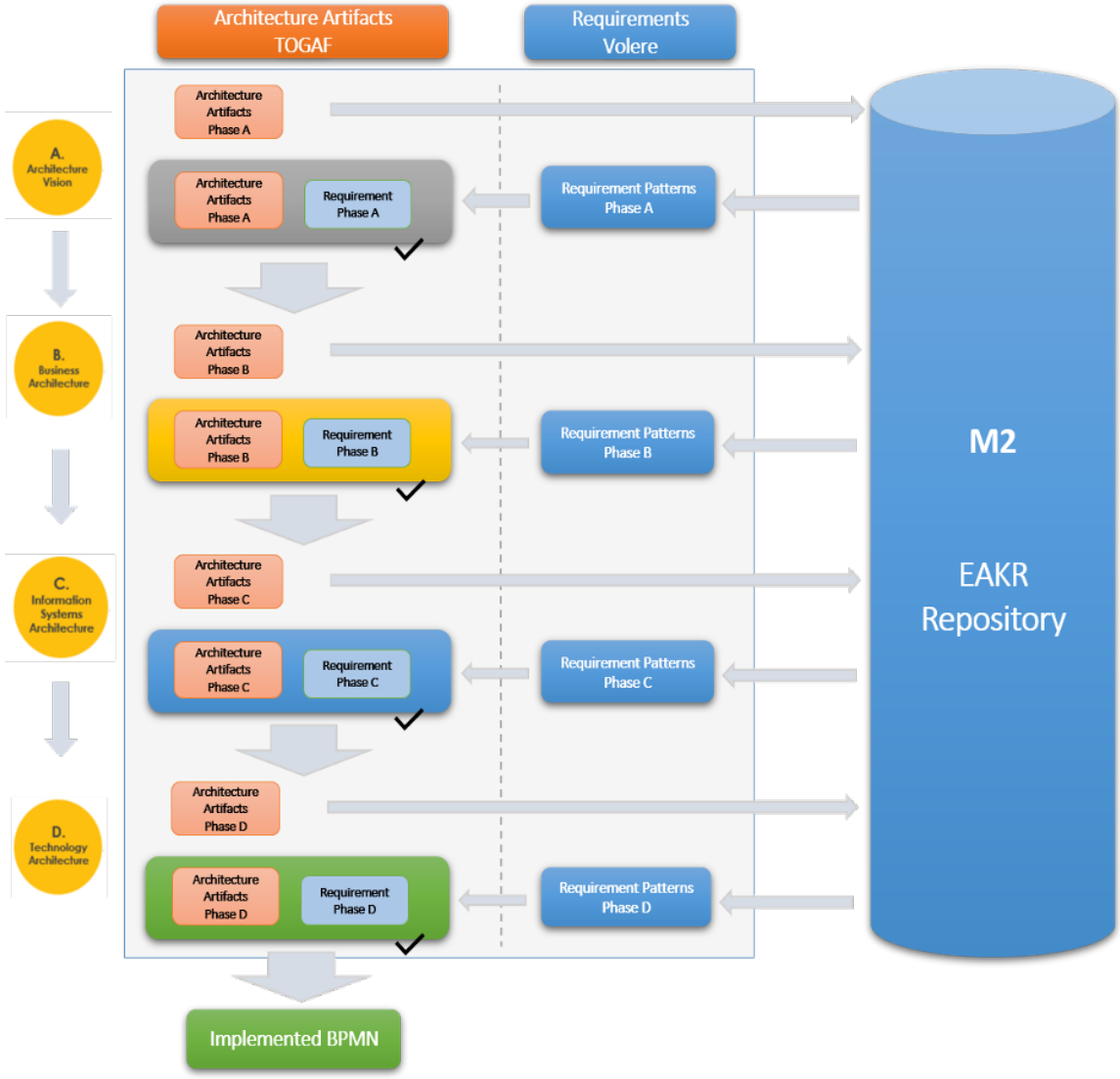

FIGURE 2.3 – Le processus d'exploitation EACP

qualifiées dans le référentiel EAKR. Une qualification continue est nécessaire pour maximiser l'exploitation et doit être réalisée fréquemment pour tirer pleinement parti de la méthodologie. Par ailleurs, l'augmentation du nombre de solutions qualifiées, et la multiplication des exigences poseront à terme la question de la complexité des algorithmes de proposition de services.

2.2.8 Synthèse

Ce travail de recherche fournit un framework de construction et d'exploitation de profil de capacité permettant la qualification, la découverte et la réutilisation pour le développement de nouveaux systèmes d'information.

Ce framework est construit sur un Métamodèle, sur la base du référentiel TOGAF et du standard ISO 16100, permettant la conception et la consolidation d'un référentiel contenant les profils de

capacités. Ces derniers peuvent être directement reliés aux exigences des utilisateurs.

Il offre, d'une part, un processus de qualification des logiciels des entreprises. Il crée un lien entre une fonctionnalité et le composant physique correspondant (dans ce cas, le point de terminaison de service). Il permet d'offrir une qualification qui traite de la perspective métier, contenant les contraintes organisationnelles, fonctionnelles et de la perspective technique, contenant les points de vue techniques et technologiques ainsi que les mesures de qualité associées. D'autre part, un processus d'exploitation proposé pour guider un développeur ou un architecte pour formaliser et consolider l'exigence, et identifier les composants ou artefacts les plus appropriés à réutiliser.

Pour organiser les profils de capacités produits au cours du processus de qualification, nous avons adopté une approche sémantique pour concevoir ce que nous avons appelé Enterprise Architecture Knowledge Repository (EAKR) et utilisant l'ontologie EACP, créée spécifiquement pour intégrer les concepts de notre Métamodèle.

Nous avons validé l'approche proposée sur un cas d'utilisation industrielle couvrant les différents problèmes traités par notre méthodologie. Les différents concepts présentés dans ce travail de recherche ont été mis en œuvre en tant que prototypes open source basés sur NodeJS et Java [2].

2. https://github.com/AbdBelf/EacpFramework

2.3 Approche centrée donnée pour une collaboration des systèmes d'information

2.3.1 Objectifs de recherche

Alors que dans la section précédente, nous avons traité la conception de systèmes d'information à partir de définition de capacités logicielles, dans les travaux présentés ci-dessous, nous traitons de l'interopérabilité de systèmes d'information existants, en fonctionnement, difficilement modifiables, appelés communément systèmes hérités.

L'interopérabilité a toujours été perçue comme la faculté pour deux systèmes informatiques hétérogènes de fonctionner conjointement et de donner accès à leurs ressources de manière réciproque [Chen 2008]. Cependant, plusieurs contraintes peuvent restreindre la collaboration des systèmes. Parmi ces contraintes, les systèmes hérités ont plusieurs problèmes : manque de documentation, coûts de maintenance importants, et surtout ils nécessitent une main-d'œuvre qualifiée. Malgré ces limites, les entreprises ne peuvent pas éliminer ou remplacer ces solutions, car elles représentent un potentiel économique important.

L'automatisation robotisée des processus [van der Aalst 2018] (RPA – Robotic Process Automation) permet d'automatiser les processus et le traitement d'information en manipulant les entrées de ces applications, par exemple les interfaces homme-machine. Elle permet de traiter de grandes quantités de données en limitant le développement de pivots d'interopérabilité entre applications. Ces concepts ont permis une certaine ouverture et évolutivité, mais ils ont été critiqués notamment à cause de l'effort de maintenance supplémentaire et de la difficulté de capitaliser les règles métier qui sont embarqués dans ces processus.

Dans la thèse de Rami Torkhani, nous abordons le problème d'amélioration des logiciels hérités en analysant les données et en favorisant la réutilisation des fonctions accessibles. Nous cherchons l'extensibilité du système d'information par la génération des microservices utilisant la RPA. Pour cela, nous analysons les informations issues d'un flux de données semi-structuré et proposons la réutilisation des systèmes d'information existants. L'objectif est de proposer un mécanisme qui pourra générer automatiquement le processus collaboratif entre le flux de données à traiter et le système hérité.

Nous avons appliqué l'approche dans un contexte métier d'assurance. Dans cet environnement, il y a trois acteurs clés : les compagnies d'assurance, les courtiers et les clients. Depuis 2015, de nombreux changements économiques et organisationnels ont eu lieu. Ces changements sont étroitement liés à l'évolution du contexte réglementaire européen, qui présente de nouvelles contraintes et nécessite de faire communiquer les systèmes d'information entre eux.

2.3.2 Contribution

Notre contribution consiste en un framework d'interopérabilité basé sur l'automatisation robotisée des processus. Dans cette étude, nous automatisons la collaboration entre les systèmes d'information en essayant de ne pas impacter la stabilité des logiciels concernés. Notre objectif est de concevoir un outil qui pourra exécuter des tâches récurrentes telles que la gestion de nouveaux contrats ainsi que la diffusion des mises à jour d'information client sur toutes les applications de l'entreprise.

La solution doit être capable d'harmoniser automatiquement un flux de données tout en effectuant

les transformations nécessaires pour pouvoir exploiter et traiter cet ensemble de données par le système d'information cible. Ainsi, la première étape est de caractériser le contexte de l'interopérabilité et d'analyser la structure des données partagées. Ensuite, nous caractérisons un ensemble de profils de capacités logicielles, représentant les fonctions qui pourraient traiter le flux de données. Enfin, nous générons une collaboration en associant les données et les profils de capacités.

Nous proposons un framework capable de gérer le cycle de vie d'une opportunité de collaboration entre systèmes d'information. Ce framework est capable d'identifier et d'analyser les fonctionnalités pouvant assurer le traitement des données et de les présenter à l'aide d'un profil de capacités logicielles. Une fois les capacités de l'application répertoriées, le framework peut recevoir un flux de données, et l'associer à la fonctionnalité. Il formalise les modifications à réaliser sur le flux de données pour assurer son traitement.

L'architecture du framework, visible dans la Figure 2.4, est divisée en quatre blocs, expliqués dans les sections suivantes.

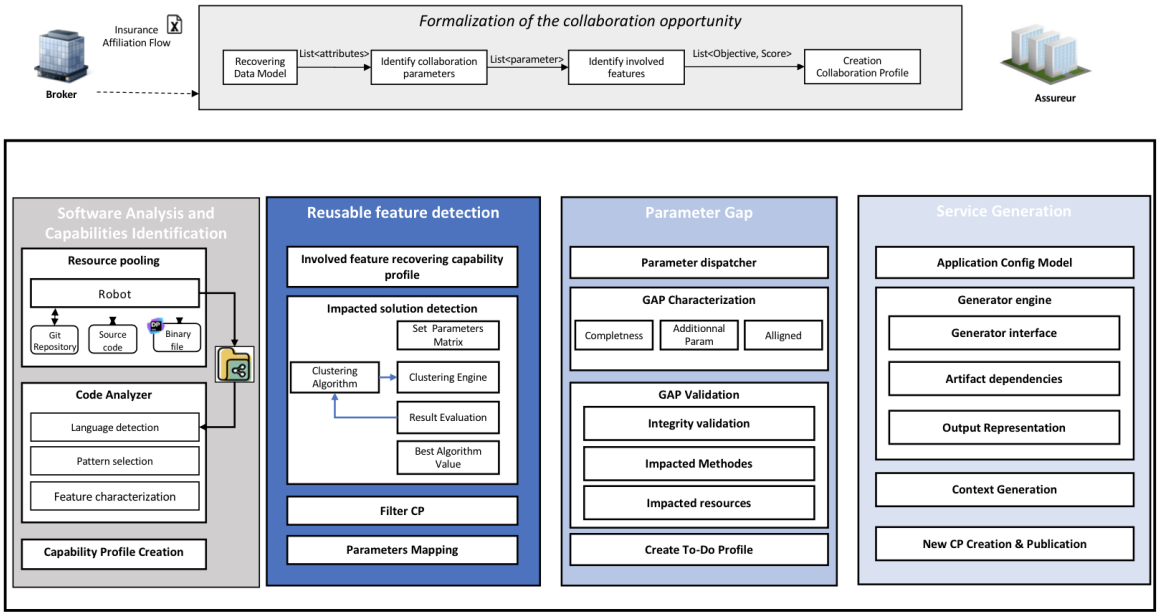

FIGURE 2.4 – Modèle du framework

2.3.2.1 Identification des fonctionnalités accessibles

Cette partie du framework vise à référencer les logiciels existants, les analyser et déposer leur description dans un référentiel commun.

L'objectif ici est d'extraire les fonctions accessibles et d'en sortir les paramètres et le type de sortie. L'algorithme développé recherche dans le code source, ou dans les binaires décompilés les fonctions accessibles et les décompose selon le modèle décrit dans la Figure 2.5. Ce modèle contient les informations suivantes :

— Feature name : Nom de la fonction,

— Scope : spécifie la visibilité de l'entité.
— Paramètres d'entrée : spécifie la liste des paramètres d'entrée. Ils sont constitués d'un nom et d'un type de données.
— Fonctions appelées : spécifie les fonctions invoquées au cœur de la fonction analysée. Cette analyse permettra de définir l'arbre de dépendance des fonctionnalités.
— Paramètre de sortie : spécifie le paramètre de sortie. Il est constitué d'un nom et d'un type de données.

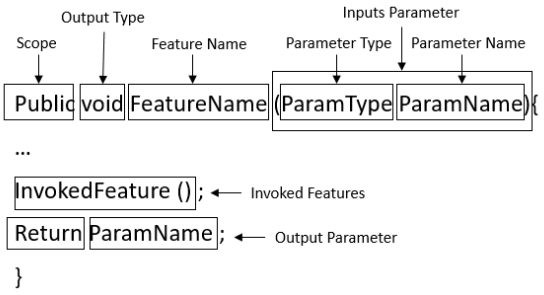

FIGURE 2.5 – Modèle d'analyse des fonctions

Ces résultats sont utilisés pour compléter un profil de capacités pour chaque fonctionnalité. La structure du profil de capacité est inspirée de la norme ISO 16100 [ISO Central Secretary 2018], visible en Figure 2.6. La structure a été modifiée en fonction de notre besoin.

L'analyse des fonctions permet de remplir partiellement les profils de capacités. Certaines informations, comme la performance, restent manquantes. Elles sont à compléter manuellement. Nous pouvons envisager d'améliorer le processus en considérant la documentation et l'environnement d'exécution des applications pour remplir les champs vides.

Le profil de capacité est ensuite stocké dans un référentiel. Il sera consulté lors de la recherche de fonctionnalités pour le traitement de données.

2.3.2.2 Formalisation de l'opportunité de collaboration

La formalisation de l'opportunité de collaboration consiste à associer les éléments du profil de capacité avec l'exigence. L'exigence est considérée ici comme un flux de données qui doit être traité.

Un algorithme récupère le flux de données et analyse ces données pour en extraire l'ensemble des types à traiter. Ensuite, il cherche dans le référentiel l'existence de ces types de données. Si les paramètres sont présents, un profil d'opportunité de collaboration est créé. Ce profil agit comme un filtre et ne considère que les fonctions compatibles avec le type de données considéré.

Le profil d'opportunité enregistre les informations selon la formalisation suivante :

— Identification de la collaboration,
— Date de traitement,
— Objectif de la collaboration : cet attribut décrit le but de la collaboration,
— Inputs : cet attribut décrit la liste des paramètres d'entrée,

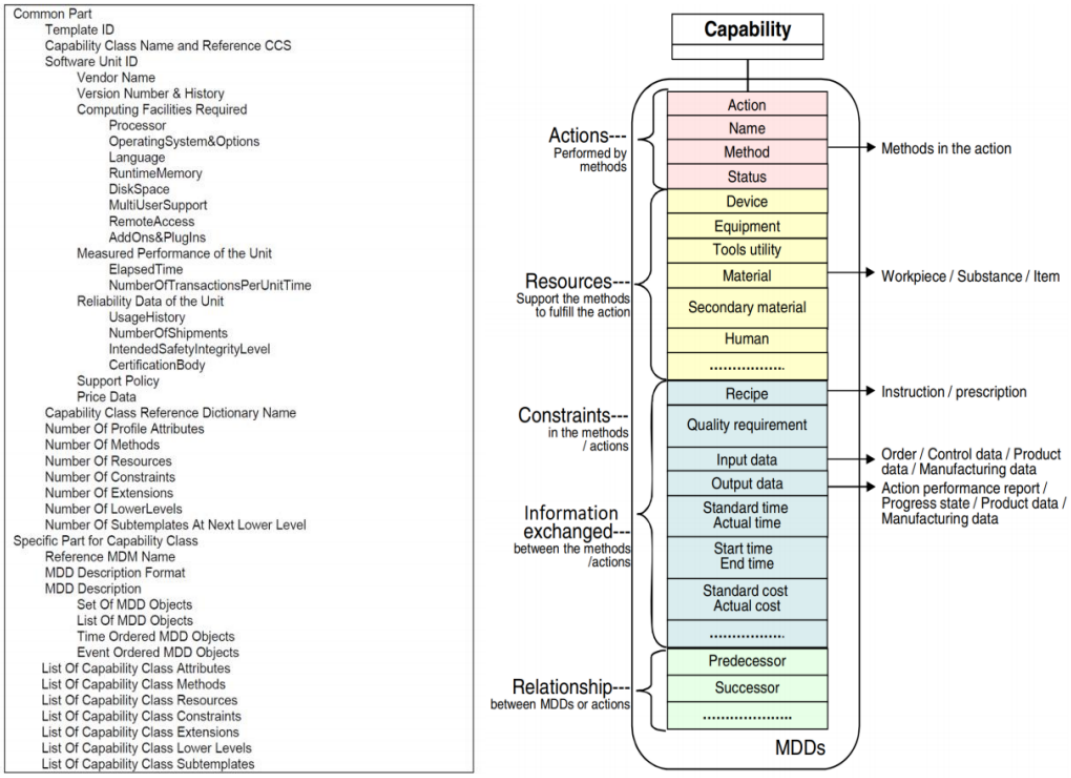

FIGURE 2.6 – Profil de Capacité de l'ISO 16100

. Type de données : cet attribut décrit les types de données d'entrée,

. Fonctionnalités impliquées : l'ensemble des fonctionnalités détectées comme compatibles avec le type de données considéré,

— Paramètre de sortie : cet attribut décrit le paramètre de sortie,

2.3.2.3 Évaluation des opportunités de collaboration

À ce niveau, les fonctions sont classées pour définir la solution logicielle la plus adaptée pour la collaboration. Nous commençons par établir un croisement entre les paramètres qui apparaissent dans le profil de capacité et les paramètres du profil de collaboration. L'objectif est de trouver la fonction la plus proche de l'exigence recherchée.

Chaque paramètre est ensuite associé à une catégorie :

— Situation d'alignement : Le paramètre existe dans le profil collaboratif et dans le profil de capacité. Cela signifie que la donnée semble être utilisable par la fonctionnalité.

— Donnée supplémentaire : La donnée existe dans le profil de collaboration et n'existe dans aucun profil de capacité. Il s'agit des données qui ont été identifiées dans le flux, mais qui ne correspondent à aucun paramètre de la fonction décrite dans le profil de capacité.

— Problème d'exhaustivité : Le paramètre n'existe pas dans le profil de collaboration et existe

dans le profil de capacité. Dans ce cas, la fonction attend un paramètre qui ne se trouve pas dans le flux de données.

Cette catégorisation permet de caractériser l'écart entre le profil de collaboration et les profils de capacité. L'écart permet d'établir une stratégie de génération du service :

— En situation d'alignement, les données peuvent être traitées.

— En cas de donnée supplémentaire, l'algorithme reprend son analyse pour identifier une fonctionnalité permettant de traiter cette donnée. S'il trouve la fonctionnalité, l'état passe en situation d'alignement, sinon, l'architecte doit définir le niveau d'importance de la donnée (nécessaire ou facultative). Si elle est nécessaire, alors un développement spécifique est à envisager, si elle est facultative alors elle peut être abandonnée.

— S'il y a un problème d'exhaustivité, la donnée doit être créée pour pouvoir utiliser la fonctionnalité. La stratégie est à définir au niveau métier.

2.3.2.4 Génération du service

Une fois la solution identifiée, nous proposons un générateur de structure du nouveau service. Elle consiste en l'implémentation d'une définition d'un service déclarant les types et formats d'entrées et de sorties. Elle déclare également les dépendances entre les artefacts.

2.3.3 Expérimentation

Nous avons pu mettre en oeuvre chacune des étapes du framework et valider son fonctionnement sur un cas d'utilisation du domaine de l'assurance. Nous avons ciblé le processus d'adhésion à l'assurance. Nous nous sommes intéressés particulièrement à l'affiliation à l'assurance collective. Dans ce type d'affiliation, les entreprises confient la procédure d'adhésion à des courtiers d'assurance. Les entreprises transmettent leurs données sur les salariés aux courtiers, ces derniers s'engagent à trouver l'assureur qui aura la meilleure opportunité. La spécificité de ce type de processus d'assurance est que les données ne correspondent pas toujours aux exigences des systèmes d'information des assureurs.

2.3.4 Synthèse

Ce travail de recherche fournit un framework de définition de profil de capacité à partir de systèmes hérités et permet de construire une collaboration entre systèmes d'information via la réutilisation de fonctions exposées.

Ce framework est construit à partir de fonctions disponibles dans les systèmes d'information existants, et de flux de données ne pouvant être traités en l'état. Les fonctions sont qualifiées avec une perspective technique en utilisant le standard ISO 16100, permettant l'alimentation d'un référentiel contenant les profils de capacités.

Le processus d'exploitation proposé guide l'architecte du système dans la formalisation et la consolidation du traitement des informations par des processus déjà existants dans l'entreprise.

Cette approche est appliquée au secteur de l'assurance. Ce secteur subit régulièrement des changements d'ordre juridique ou des fusions-acquisitions qui impliquent des changements importants dans le traitement des données des adhérents.

2.4 Bilan de l'axe et perspectives

Pour pouvoir répondre aux nouvelles exigences, les systèmes d'information doivent interopérer avec des solutions nouvelles. La collecte et la qualification des composants logiciels provenant des solutions internes ou externes, permet de réutiliser des fonctionnalités. L'analyse de ces fonctionnalités doit inclure la qualification des processus afin de pouvoir déterminer l'écart fonctionnel [Ostadzadeh 2015].

Par conséquent, nos travaux s'attachent à améliorer la description des capacités des logiciels selon différents niveaux de description, ainsi que leur exploitation. Notre objectif est de maximiser la réutilisation des logiciels internes et externes d'une organisation en améliorant la découverte et la durabilité des solutions existantes, et en recherchant la compatibilité fonctionnelle et non fonctionnelle la plus élevée.

Le principal problème que nous avons abordé dans cet axe est de savoir comment gérer la complexité de l'exploitation des profils de capacités logicielles. D'une part, j'ai abordé le problème sur la base d'un alignement des exigences et des processus d'ingénierie, pour sélectionner les meilleurs composants candidats et les intégrer comme blocs de construction dans un nouveau système. D'autre part, j'ai exploité la qualification de fonctions d'un système d'information dans un cadre collaboratif restreint, c'est à dire sans possibilité d'aller chercher de nouvelles applications.

Ce travail de recherche ne représente qu'une partie de la conception des systèmes d'information. Chaque situation s'est révélée être restreinte par le domaine d'application. La première contribution n'est appliquée que sur des architectures orientées service. Cette particularité nous a permis de concentrer nos efforts sur la corrélation entre les exigences et les services exposés. Cependant, le framework proposé a été conçu de manière à être extensible. La deuxième contribution se trouve dans un contexte où les systèmes d'information doivent communiquer ensemble alors qu'ils n'ont pas été conçus dans ce sens à l'origine. Nous avons donc cherché des moyens de les faire interopérer, par l'utilisation de RPA, et avons maximisé l'automatisation de la construction de la collaboration.

Actuellement, nous avons identifié plusieurs extensions à court terme aux approches proposées :

— Les exigences du système émergeant sont parfois en contradiction avec les exigences des systèmes constituants en termes de performances, sécurité ou tolérance aux pannes. Les systèmes hérités par exemple ont moins de points d'entrée et ne sont pas adaptés à être exposés. Lorsqu'on les expose via des services, leur stabilité peut être mise en jeu.

— La base référentielle, notamment l'EAKR, est alimentée par notre processus. Une connexion avec un référentiel externe (par exemple "ProgrammableWeb.com" qui a documenté plus de 12.000 API web ouvertes et des milliers d'applications dans son référentiel) permettrait au framework de fournir des solutions plus adaptées aux exigences exprimées.

— Dans les deux approches présentées, nous avons abordé la problématique en considérant chaque fonctionnalité séparément. La complexité peut augmenter rapidement avec l'ajout de plusieurs composants. Les approches pourraient intégrer des méthodes de factorisations des exigences. Nous pourrions développer une approche holistique de manière à traiter le système comme un tout et analyser l'ensemble des exigences du système.

2.5 Synthèse des contributions

Chacune des deux problématiques a fait l'objet de plusieurs publications.

La section 2.2 a été présentée dans :

— [23] : présentation conceptuelle de l'utilisation de l'ISO 16100 pour la qualification des exigences.

— [20] : utilisation de l'utilisation de l'ISO 16100 pour la qualification des exigences.

— [91] : validation de l'utilisation de l'ISO 16100 pour la qualification des exigences.

— [17] : présentation du référentiel "Enterprise Architecture Knowledge Repository".

— [89] : validation du référentiel "Enterprise Architecture Knowledge Repository".

— [90] : validation du framework global.

La section 2.3 a été présentée dans :

— [24] : présentation d'un premier framework d'exploitation des données utilisant l'ISO 16100.

— [21] : utilisation de l'ISO 16100 pour élaborer des profils de capacités.

— [18] : présentation du framework global.

CHAPITRE 3

Axe 2 : Contribution à la supervision des systèmes d'information distribués

Contents

3.1 Contexte **30**
3.2 Supervision des échanges de données d'un système d'information distribué **32**
3.3 Gestion des vulnérabilités de la cyber-sécurité à l'aide de réseaux de neurones . . **44**
3.4 Bilan de l'axe et perspectives **53**
3.5 Synthèse des contributions **53**

Le meilleur alpiniste au monde est celui qui s'amuse le plus.
[Alex Lowe]

En quelques mots...

Ce chapitre introduit mes contributions à la supervision des systèmes d'information distribués. Dans cet axe de recherche, nous considérons la dynamique d'un système d'information. Dans la phase d'exploitation, chaque constituant évolue, se connecte à différentes plateformes, reçoit et émet des données. Chacun est indépendant. Le système d'information doit garder sa stabilité et sa fiabilité. Pour contrôler son comportement, nous utilisons la supervision. J'ai travaillé sur deux aspects de la supervision. D'une part, la supervision fonctionnelle, interne au système, pour laquelle je supervise un contrat de collaboration avec l'entreprise Berger-Levrault. D'autre part, la supervision non fonctionnelle, externe au système, en abordant l'identification des vulnérabilités à partir des informations issues de plateformes spécialisées. Ce travail est supporté par la thèse de Houssem Gasmi.

Ces travaux ont été présentés dans les publications [1, 11, 12, 16, 19, 43, 46, 60].

3.1 Contexte

La phase d'exploitation des systèmes de systèmes lève de nombreux verrous scientifiques. Notamment, nous avons relevé les problématiques de stabilité et de fiabilité de comportement lorsque les constituants évoluent : chaque constituant a une indépendance managériale et peut être ajouté ou supprimé du système. Il peut apparaître des incidents sur l'ensemble du système de systèmes, ou des comportements émergents. Cette dynamicité ne permet pas d'anticiper le comportement global du système au moment de sa conception [Ceccarelli 2016].

Une fois que le système de systèmes (système d'information distribué dans notre cas) est mis en production, ses constituants sont capables de partager et d'échanger des informations sans dépendre d'un acteur particulier et de fonctionner indépendamment les uns des autres, nous les qualifions de systèmes interopérables [Chen 2008]. L'interopérabilité est devenue un atout important pour les entreprises et il est maintenant proposé de la considérer comme un indicateur de performance essentiel des systèmes de gestion de la performance des processus métiers [Verdecho 2018].

Le problème de l'interopérabilité entre différents systèmes hétérogènes existe déjà et se trouve amplifié par le fort déploiement des applications réparties et de l'Internet des objets. Pour relever ce défi, les entreprises mettent l'accent sur l'utilisation de standards ouverts pour le format de données ainsi que pour les protocoles de communication.

Certains problèmes d'interopérabilité des données sont liés à la charge. Les systèmes interopérables peuvent échanger de grandes quantités de données tout en nécessitant une faible latence. C'est le cas lors de l'interaction avec des systèmes IoT [Buyya 2016]. L'augmentation des volumes de données échangées implique la mise en œuvre d'architectures d'échanges capables de supporter non seulement la charge, mais également la variabilité de la fréquence de production des données. Cela nécessite des architectures distribuées (à la fois en infrastructure et en flux) qui peuvent être adaptables ou même autoadaptables [Gascon-Samson 2015] afin de renforcer la résistance du système aux dysfonctionnements tout en évitant les phénomènes de congestion potentiels et en garantissant ainsi des interactions d'interopérabilité fiables.

Les architectures orientées évènements (EDA) sont des architectures mettant en œuvre des services réagissant à des évènements externes. L'utilisation d'évènements permet à l'architecture d'avoir un fonctionnement asynchrone et un couplage faible. Une architecture orientée évènement repose sur un bus disposant de fonctionnalités d'abonnement et de publication (abonnement/publication). Les courtiers de messages (message broker) sont généralement utilisés pour découpler des étapes distinctes d'une architecture logicielle. Ils permettent la communication de manière asynchrone qui favorise le découplage des applications, en utilisant le paradigme abonnement/publication. Ces courtiers de messages trouvent également de nouvelles applications dans le domaine des périphériques IoT et peuvent également être utilisés comme méthode pour mettre en œuvre une architecture de traitement pilotée par les évènements. La multiplicité de ce type d'échanges de données génère de la complexité et fait ressortir des besoins de contrôle.

Par ailleurs, les systèmes d'information sont de plus en plus exposés à diverses menaces de sécurité qui nécessitent une attention constante de la part des entreprises. Pour mesurer les risques de sécurité, les approches actuelles de gestion des risques font défaut dans certains domaines, comme le besoin de connaissances détaillées sur l'environnement de l'entreprise et le corpus de connaissances sur la cybersécurité [Finkel 2005]. Pour rester à jour avec les dernières alertes en matière de sécurité, un système d'information doit être mis à jour en fonction des dernières informations de sécurité telles

que les vulnérabilités, les attaques, etc., ce qui est une tâche difficile, car les informations sont mises à jour quotidiennement.

D'après le rapport [Dynatrace 2020], en 2019, les équipes en charge des opérations cloud ont traité en moyenne 2 973 alertes par jour remontées par leurs outils de monitoring. Elles consacrent en moyenne 15% de leur temps à faire un tri pour identifier les alertes sur lesquelles se concentrer. Selon une autre étude [Cynet 2020], entre 20% et 40% des alertes générées par les solutions de sécurité sont ignorées quotidiennement.

Les approches et les outils de surveillance traditionnels ne permettent pas d'appréhender le volume, la vitesse et la variété des alertes générées aujourd'hui. Parmi ces verrous impactant la continuité de service, j'ai abordé deux aspects : la supervision interne d'un système d'information distribué illustré par une collaboration avec l'entreprise Berger-Levrault et la supervision des éléments externes pouvant avoir un impact sur le système illustré par la thèse de Housem Gasmi, dont la soutenance est prévue en 2020.

Par la supervision des échanges de données d'un système distribué : La section 3.2 décrit nos travaux dans la supervision des échanges de données d'un système d'information distribué. Dans ce travail, nous abordons la collecte, la modélisation et le traitement des données de manière à analyser et identifier l'évolution d'architecture du système et des problèmes pouvant survenir. Le framework proposé prend en charge différents protocoles d'échange de données. Il permet de naviguer dans la structure proposée, de requêter et de visualiser l'architecture. Les problèmes d'architecture sont identifiables. Ce travail nous a menés à démarrer une thèse CIFRE en 2020.

Par la gestion de la vulnérabilité de la cybersécurité : La section 3.3 décrit nos travaux dans l'analyse des alertes de sécurité des systèmes d'information. La diffusion publique d'informations de sécurité importantes se produit souvent d'abord dans différentes sources en ligne comme les blogs, les bulletins des fournisseurs et les bases de données en ligne [Bridges 2013]. Les informations sont ensuite collectées et stockées dans des bases de données de vulnérabilité semi-structurées telles que la National Vulnerability Database (NVD) [1]. L'analyse des informations de cybersécurité nécessite une extraction automatisée des informations à partir des sources web. Cela est particulièrement difficile car les informations ne sont pas structurées. Nous avons proposé un framework permettant d'extraire ces informations et de les représenter dans une ontologie.

1. https://nvd.nist.gov/

3.2 Supervision des échanges de données d'un système d'information distribué

3.2.1 Objectifs de recherche

Les architectures orientées évènements (EDA) permettent un couplage lâche, mais limitent la capacité à comprendre le comportement global de l'architecture. La multiplicité des échanges de données génère de la complexité et fait ressortir des besoins de supervision auxquels il est possible de répondre en établissant des systèmes de surveillance et d'analyse [Brand 2019]. Un système de surveillance est défini comme un processus ou un ensemble de processus répartis comprenant la fonction et la collecte, l'interprétation et le traitement dynamique d'informations relatives à une application en cours d'exécution.

Les structures existantes (par exemple Apache Camel [2] [3], NServiceBus [4]) fournissent des consoles de surveillance. Cependant, ils se concentrent principalement sur les informations de contrôle de bas niveau telles que la fréquence des messages, les indicateurs de performance ou l'utilisation de la mémoire.

RabbitMQ [5] offre une console de gestion contenant des informations relatives à la structure du système de messagerie et à l'état des messages [Dossot 2014]. Il présente des listes de ressources existantes (canaux, échanges, files d'attente ...), leur contenu, leurs caractéristiques et un ensemble de statistiques. Il est par exemple possible d'accéder aux files d'attente et de vérifier les messages en attente. Sur la base de notre expérience, la console RabbitMQ peut être utilisée pour la surveillance en temps réel et convient à des requêtes spécifiques dans lesquelles le responsable de la maintenance connaît les files d'attente ou les échanges à suivre. Cependant, il ne permet pas d'effectuer des recherches et des filtres avancés sur les sources et les messages en transit. Les messages consommés ne sont plus visibles dans la console de gestion. De plus, les canaux de messagerie tels que les exchanges, les files d'attente et leurs liaisons sont volatiles et peuvent être supprimés lorsque le consommateur se déconnecte.

L'absence de ces éléments de contrôle complique l'analyse et le diagnostic des dysfonctionnements, tels que l'inactivité des éditeurs et des consommateurs, la non-validité des formats d'échange ou l'indisponibilité des données. Par conséquent, il est difficile pour les responsables d'identifier le contexte et l'origine du problème, en se basant uniquement sur la console de gestion RabbitMQ.

Notre approche utilise la surveillance pour effectuer une analyse capable de définir une sorte de classification des causes potentielles des problèmes survenant dans les échanges par ordre d'importance pour un problème donné. Dans une première étude, nous préconisons que le comportement de l'échange de données entre les applications en communication donne une indication du niveau d'interopérabilité de leurs données. Un système de surveillance devrait fournir des éléments permettant de maintenir un bon niveau d'interopérabilité sur la base des exigences d'interopérabilité [Mallek 2010].

Nous proposons donc de combiner les informations fournies par les EDA, afin d'effectuer une surveillance et une interrogation avancées. Cela fournirait des indicateurs permettant de déterminer les actions de maintenance.

2. http://sksamuel.github.io/camelwatch/
3. http://rhq-project.github.io/rhq/
4. https://particular.net/nservicebus
5. https://www.rabbitmq.com/

Dans ce contexte, le principal problème de recherche abordé est d'assurer la supervision d'un système d'information distribué à partir des métadonnées des messages transitant dans le système. Les objectifs de recherche que nous avons abordés sont les suivants :

— Collecter les métadonnées des messages transitant dans le système d'information pour pouvoir analyser l'architecture du système d'information. Pour cela nous avons proposé le framework d'analyse nommé Pulse. Il collecte des métadonnées afin de pouvoir (i) suivre les messages échangés, (ii) simplifier la visualisation des échanges, (iii) améliorer la maintenabilité en détectant des exceptions (p. ex. les problèmes de transfert d'un message), et en précisant le contexte et l'origine du problème.

— Organiser les métadonnées collectées pour pouvoir les traiter d'une manière cohérente. Pour cela, nous avons proposé un métamodèle générique et extensible adapté aux EDA. Il est capable d'analyser les protocoles AMQP, MQTT, KAFKA et CoAP. Le modèle de données de messagerie décrit la structure de messagerie mise en œuvre via la mise en file d'attente des messages et le système d'échange. Il est utilisé pour collecter des métadonnées provenant des services de journalisation offerts par les infrastructures d'échange et assure le suivi des messages échangés.

— Identifier les problèmes d'interopérabilités et aider à les résoudre. Pour cela, nous avons réalisé une étude des causes et effets des problèmes tout en identifiant les indicateurs liés à l'interopérabilité.

— Prendre en compte le cycle de vie des différents composants. Pour cela, la structure Pulse intègre des fonctionnalités de modélisation de la dynamique, où le cycle de vie de différents constituants de l'architecture est décrit en incluant une historisation et les dates de création et de suppression des constituants.

3.2.2 Termes et vocabulaire

Nous avons abordé la supervision des systèmes d'information distribués à partir de quatre protocoles :

— AMQP [Vinoski 2006] : c'est un protocole de communication asynchrone au travers d'un médiateur. Il vise à créer un standard ouvert permettant de transporter l'information entre les applications. Un message est publié dans un échange par un producteur de messages, puis acheminé, par le courtier de messages, vers une ou plusieurs files d'attente en fonction de clés de routage prédéfinies. Les consommateurs se connectent à la file d'attente pour récupérer le message.

— MQTT [Standard 2014] : c'est un protocole de communication dédié à l'IoT. Il a été conçu comme un protocole de communication asynchrone extrêmement léger. Il peut fonctionner de la même façon que AMQP. Le modèle abonnement / publication permet de dissocier le producteur qui envoie un message du consommateur recevant les messages. Par rapport à AMQP, MQTT intègre un paramètre de qualité de service. Le niveau de service détermine le type de garantie qu'un message a pour atteindre le destinataire souhaité.

— Kafka [Garg 2013] : ce protocole a été optimisé pour diffuser les données entre les systèmes et les applications aussi rapidement que possible et de manière évolutive. Il est également basé sur un modèle de communication abonnement / publication. Les données sont organisées en sujets. Les sujets sont multiabonnés, divisés en partitions et chaque courtier de messages peut posséder une ou plusieurs de ces partitions.

— CoAP [Shelby 2014] : c'est un protocole de communication spécialisé pour les périphériques soumis à des contraintes, défini dans la RFC 7252. L'un des principaux objectifs de CoAP est de concevoir un protocole générique pour les besoins d'environnements contraints, notamment en ce qui concerne l'énergie, l'automatisation des bâtiments, et des applications machine à machine (M2M). CoAP utilise le protocole UDP.

Dans cette section, nous utilisons certains termes spécifiques liés à ces quatre protocoles :

— Connection : une connexion réseau TCP entre une application et le courtier de messages.
— Channel : un flux de communication entre deux constituants.
— Message : un message est composé d'un en-tête et d'un corps. L'en-tête contient les propriétés du message présenté dans un type de format spécifique. Le corps ou la charge sont les données d'applications également présentées dans un type de format spécifique.
— Exchange : une entité nommée qui reçoit les messages des producteurs et les achemine vers les files d'attente.
— Queue : une entité qui contient les messages et les remet aux consommateurs.
— Routing key : adresse virtuelle qu'un Exchange peut utiliser pour acheminer des messages vers des files d'attente spécifiques.
— Publisher : une application cliente qui publie des messages pour les échanges.
— Consumer : une application cliente qui demande des messages aux files d'attente.

3.2.3 Le framework Pulse

Pour collecter, analyser et identifier les problèmes de communication des architectures de messagerie abonnement / publication, nous avons défini un framework, illustré à la Figure 3.1. Le framework de supervision que nous avons proposé est composé de quatre niveaux : (i) un niveau d'importation de données, (ii) un niveau de gestion du temps et de gestion des versions de modèle, (iii) un niveau de persistance, (iv) un niveau d'analyse.

3.2.3.1 Importation de données

L'un des principaux problèmes est d'importer des données de différentes sources ayant des formats différents. Pour cela, nous avons défini un métamodèle représentant la structure des systèmes étudiés. Le métamodèle est détaillé dans la Section 3.2.4. Des importateurs dédiés consomment les messages ou les logs et instancient le modèle avec les informations récoltées.

3.2.3.2 Gestion du temps et des versions

Lorsque l'on importe les données des différentes sources, il est difficile de représenter les aspects dynamiques du système. Par exemple, l'architecture RabbitMQ et ses messages sont volatiles, donc pas visibles au niveau de la console d'administration. Nous avons donc développé une forme d'historisation pour pouvoir comprendre les évènements précédents et favoriser une meilleure analyse.

Une méthode triviale pourrait consister à intégrer un horodatage pour la création, la suppression et la mise à jour de chaque entité du modèle. Le problème avec cette approche est la stratégie pour

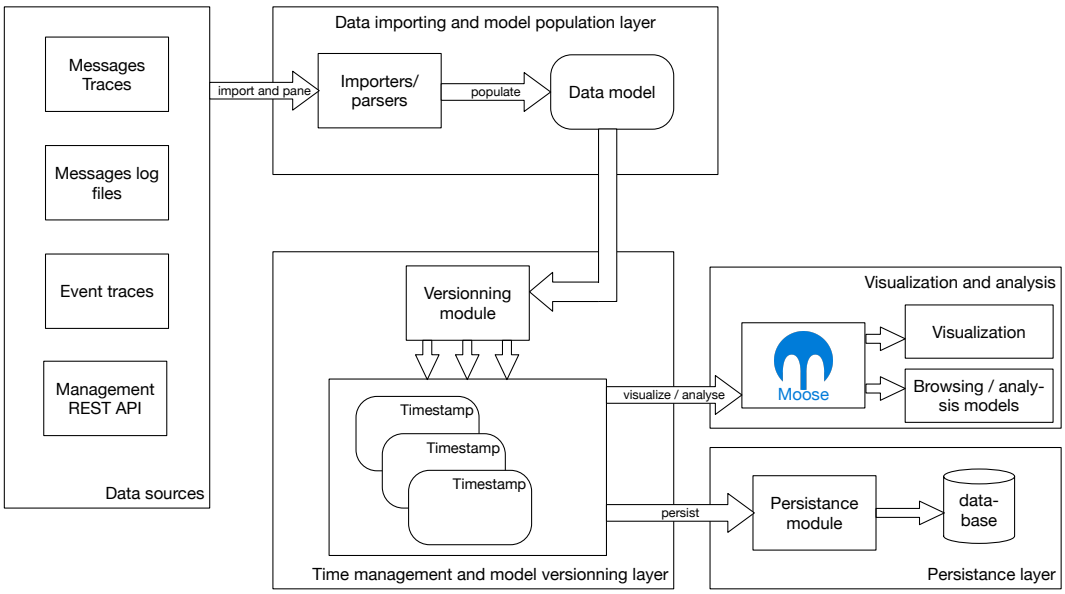

FIGURE 3.1 – Le framework Pulse

créer un statut spécifique à un moment donné. Une autre méthode peut être la création d'un modèle à chaque fois, ce qui prend beaucoup de place à chaque nouvelle donnée provenant des analyseurs.

Nous avons considéré une autre solution, basée sur Orion [8]. Orion est un modèle qui permet de créer différentes versions d'un modèle de données en tenant compte du suivi des modifications apportées à ce modèle. Le principe de base d'Orion est que chaque modification déclenche une action Orion chargée de l'ajouter au modèle de données. Chaque modification peut entraîner une nouvelle version du modèle de données. Orion optimise la persistance des différentes versions du modèle, où il gère les deltas et les pointeurs vers les versions antérieures. Orion copie les seules entités touchées par un changement.

La Figure 3.2 illustre notre stratégie de gestion de versions. Une version d'Orion inclut les nouvelles modifications et informations sur l'action effectuée pour créer cette version. Cette gestion de version du modèle de données nous permet de suivre l'évolution de l'architecture de messagerie au fil du temps, où chaque version représente une image de l'architecture à un moment donné. Pour l'utilisateur, chaque version représente une capture d'écran du système surveillé à une heure spécifique. Nous avons défini une stratégie pour créer une version chaque fois que cela est nécessaire. Dans le cas du système d'échange de messages, nous avons défini deux types d'évènements :

— Une version d'Orion est créée lorsqu'une modification de l'architecture ou des configurations / paramètres du système supervisé apparait(création de file d'attente, suppression de la file d'attente, droits d'utilisateur modifiés, etc.). Le statut du système avant et après le changement doit être conservé.

— Une entité dédiée dans la version actuelle du modèle est instanciée lorsqu'une nouvelle trace (nouveau message publié, message reçu, nouvelles connexions, etc.) apparaît.

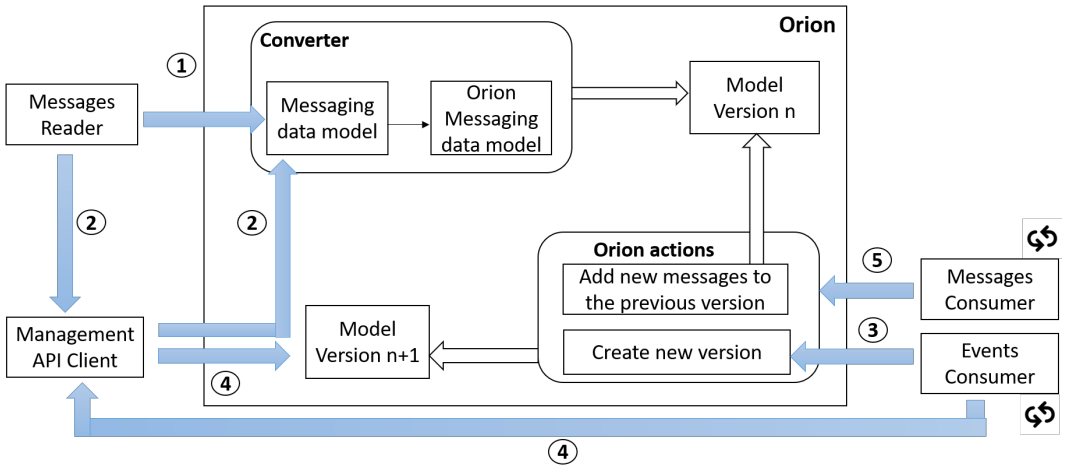

FIGURE 3.2 – Processus de gestion de version avec Pulse

3.2.3.3 Persistance

Orion conserve plusieurs versions du modèle dans l'environnement d'exécution. Ce fonctionnement est lié aux références entre les entités de différentes versions qui ne sont pas copiées d'une version à l'autre.

Lorsque notre module de persistance est appelé, toutes les versions du modèle Orion sont stockées dans un fichier JSON. Il peut être étendu pour produire d'autres types de données structurées. Grâce à cette fonctionnalité, nous permettons à un système externe tel que Grafana [6] de calculer des métriques et de les utiliser pour afficher certaines visualisations.

3.2.3.4 Analyse

La mise en œuvre de la gestion des versions de modèle nous permet de réaliser deux choses : analyser et visualiser les modifications apportées au système supervisé en temps réel, et revenir à un état antérieur du système pour visualiser et analyser les modifications et leur impact. Ces deux fonctionnalités nous permettent non seulement de détecter les problèmes d'interopérabilité au fur et à mesure qu'ils se produisent, mais également d'identifier la source potentielle d'un problème en revenant aux états précédents.

3.2.4 Métamodèle Pulse

Le métamodèle Pulse a pour objectif de représenter trois aspects du système d'information :

— Une représentation statique : l'architecture du système mise en œuvre, en représentant les files d'attente et les canaux que prennent les messages.

— Une représentation dynamique, où les messages sont représentés du composant source à la destination. Elle représente les connexions, les messages, et toute entité volatile.

6. https://grafana.com/

— La représentation du cycle de vie de l'architecture, où les composants (par exemple, file d'attente, échange. . .) sont créés, modifiés, supprimés.

Plusieurs protocoles peuvent être utilisés dans le même système d'information, l'un des objectifs de ce métamodèle est d'être suffisamment générique pour prendre en compte différents protocoles, et être extensible. Après avoir présenté un premier métamodèle [19] basé sur AMQP, nous avons analysé et comparé les différences avec d'autres protocoles et réalisé un métamodèle permettant d'intégrer les protocoles AMQP, MQTT, Kafka et CoAP. La Figure 3.3 illustre la structure des métadonnées accessibles dans les messages et accessibles dans le système d'information.

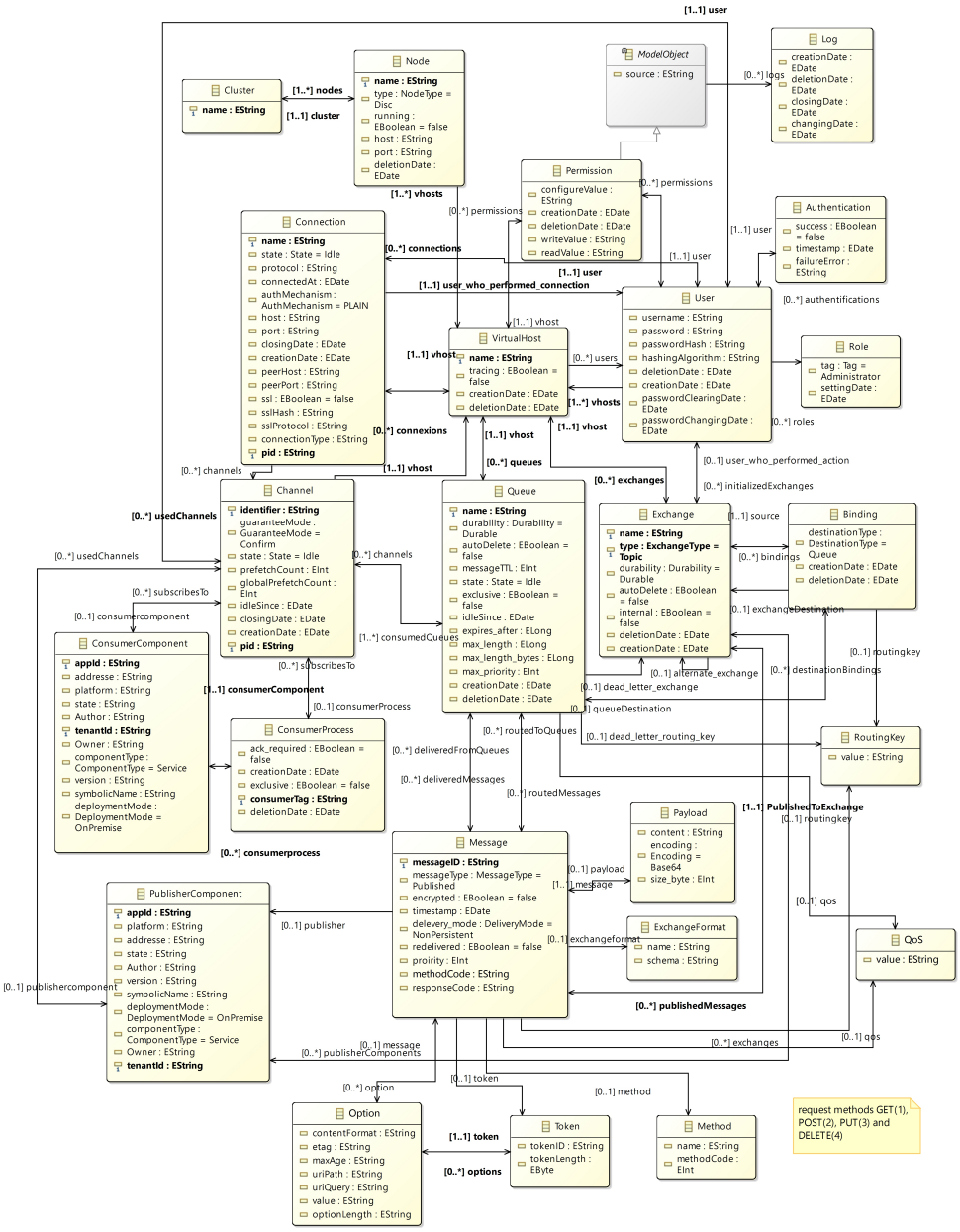

FIGURE 3.3 – Le Métamodèle Pulse

Le métamodèle regroupe des informations provenant de plusieurs sources :

— Les traces de messages fournies par la console d'administration des outils, par exemple le plug-in de traçage RabbitMQ [7], cela permet d'identifier pour chaque message :
  . Le noeud par lequel il transite, ainsi que les informations de connexion et d'hôte.
  . L'échange dans lequel il a été publié ou consommé, les files d'attente vers lesquelles il est routé ou la file d'attente depuis laquelle il est consommé, ainsi que les clés de routage associées.
  . L'utilisateur publiant ou consommant le message.
  . Son horodatage, son type (publié / reçu) et son mode de livraison (persistant ou non).
— La configuration actuelle du courtier de message, par exemple via l'API de gestion RabbitMQ REST [8].
— Les évènements de création et de suppression de ressources, d'hôtes, d'utilisateurs et d'autorisations ; création et fermeture de connexions et de canaux et tentatives d'authentification de l'utilisateur, par exemple par le plug-in RabbitMQ Event Exchange [9].
— Les éléments contextuels sur les caractéristiques des applications communicantes fournies par les métadonnées métiers.

3.2.5 Requêtes et indicateurs de supervision du système

À partir des informations récoltées, nous pouvons assurer la supervision du système. Voici un sous-ensemble des requêtes et des indicateurs qui peuvent être définis :

Requêtes au niveau métier : Les métadonnées peuvent contenir des traces fournissant des informations de niveau métiers, c'est-à-dire des informations ayant une signification identifiable, telle que : identificateurs d'application, d'utilisateurs, sujets et formats d'échange. De même, les sujets pour lesquels des messages ont été publiés lors de l'utilisation d'un modèle de messagerie de publication / abonnement peuvent être spécifiés. Ces informations aident les responsables à spécifier le contexte des interactions. Cela permet, par exemple, de prendre en compte l'identification de l'utilisateur de l'application qui publie ou consomme des messages.

Filtrage des messages : Les messages peuvent être filtrés en fonction de leurs caractéristiques telles que : identificateur, horodatage, échange, file d'attente, éditeur, consommateur, utilisateur, état, taille, encodage ou format d'échange. Cela permet d'avoir plus de précision lors de la recherche dans les traces de messages. Par exemple, le filtrage des messages qui transitent dans un intervalle de temps entre un producteur et un consommateur permet d'analyser le comportement de l'interaction pendant une période où une panne se produit. Nous pouvons également vérifier si un message est dupliqué, redéposé ou rejeté.

Contrôles de sécurité : superviser la sécurité des échanges de messages peut être amélioré en vérifiant plusieurs éléments :

— les horodatages d'authentification de l'utilisateur et le succès ou l'échec de l'authentification. Cela contribue à la détection des tentatives de violation.

7. https://www.rabbitmq.com/firehose.html
8. https://pulse.mozilla.org/api/
9. https://www.rabbitmq.com/event-exchange.html

— les autorisations des utilisateurs et des consommateurs sur chaque ressource, afin d'éviter les fuites de données potentielles.

— le contenu de la charge utile permet de vérifier si les éléments sont cryptés ou non.

Évolution de l'architecture : la possibilité de suivre la création et la suppression de ressources permet de décrire les configurations du système au cours du temps. De plus, en nous basant sur les informations métiers fournies, nous pouvons identifier les applications communicantes même si leurs identifiants techniques ont changé au niveau du courtier de messages. Nous pouvons par conséquent fournir une visualisation plus précise des constituants du système et de leurs interactions, ainsi que leur évolution dans le temps.

Indicateurs d'interopérabilité : le comportement de l' échange de données entre applications donne une indication du niveau d'interopérabilité de leurs données. Si, par exemple, une ressource est restée inactive pendant un certain temps, cela peut indiquer l'inutilité ou l'obsolescence de l'interaction. Il peut en outre indiquer un changement de configuration (consommation sur un autre canal) ou un changement de processus au niveau de l'application. Nous décrivons dans la Table 3.1 comment nous pouvons mettre en évidence certains problèmes d'interopérabilité des données par des indicateurs contextualisés ou des requêtes possibles sur le métamodèle proposé. Nous indiquons également les causes potentielles et soulignons les corrélations existantes entre les problèmes d'interopérabilité.

3.2.6 Application à l'analyse de l'interopérabilité

Les modèles de maturité [Group 1998, Clark 1999, Tolk 2003, Guédria 2009, de Soria 2009, Kingston 2005] ont d'abord été utilisés comme approches d'évaluation de l'interopérabilité. Ils permettent d'évaluer l'interopérabilité en respectant un ensemble de niveaux prédéfinis et fournissent des recommandations pour passer d'un niveau à l'autre afin d'atteindre le niveau requis ou d'atteindre la pleine maturité. Cependant, les modèles de maturité ne permettent pas d'indiquer avec précision les causes de la non-interopérabilité et se concentrent principalement sur les notions qualitatives générales [Mallek 2010].

Certaines approches [Daclin 2008, Ford 2007] fournissent les concepts fondamentaux qui permettent de formaliser et d'évaluer l'interopérabilité en indiquant si des problèmes d'interopérabilité existent ou non. Sur la base de ces concepts et afin de localiser avec précision les problèmes d'interopérabilité parmi les processus collaboratifs, Mallek et al. [Mallek 2012] définissent un ensemble d'exigences d'interopérabilité qui doivent être vérifiées afin de réaliser l'interopérabilité. Les exigences sont structurées en termes de propriétés de compatibilité, de réversibilité et d'interopérabilité définies pour chaque problème d'interopérabilité (données, services, processus et activités). Cette formalisation des exigences d'interopérabilité nous permet de sélectionner les exigences d'interopérabilité appropriées à vérifier.

Nous proposons de mettre l'accent sur les exigences relatives aux données afin d'évaluer une interopérabilité des données en analysant les échanges de données existants entre les constituants. Ces exigences sont liées à des éléments qui peuvent varier ou être modifiés lors des interactions. Ils doivent donc être surveillés. Pour ce faire, nous inspectons le comportement des échanges de données. Cela nous permet de détecter les problèmes d'interopérabilité et de trouver leurs causes potentielles. Nous pouvons résumer les verrous en deux catégories : le monitoring des échanges de données et l'évaluation des échanges.

Des indicateurs et des requêtes sont utilisés pour détecter et localiser les problèmes d'interopérabilité afin de maintenir un bon niveau d'interopérabilité. Pour définir ce qu'est un bon niveau d'interopérabilité, nous nous basons sur des exigences d'interopérabilité, c'est-à-dire sur ce qui doit être entrepris et maintenu lors de l'établissement d'interactions d'interopérabilité entre des systèmes en communication.

Les problèmes d'interopérabilité représentent des situations dans lesquelles les exigences ne sont pas vérifiées. En plus des indicateurs qui mettent en évidence ces problèmes, nous fournissons des causes potentielles et soulignons les corrélations existantes entre les problèmes d'interopérabilité (en référençant d'autres problèmes dans les causes). Si, par exemple, les partenaires ne montrent aucune interaction au niveau des données sur une certaine période, cela peut indiquer l'inutilité ou l'obsolescence de l'interaction due à un changement de configuration (consommation sur un autre canal) ou un changement au niveau du processus. Cela peut en outre indiquer un problème d'autorisation ou l'invalidité du format d'échange.

La Table 3.1 liste quelques exemples qui montrent comment nous pouvons mettre en évidence les problèmes d'interopérabilité des données en fournissant des indicateurs ou des requêtes possibles qui définissent avec précision le contexte du problème. Par exemple, une fuite de données peut être détectée par la présence de consommateurs non autorisés ou être empêchée en surveillant les tentatives d'authentification.

TABLE 3.1 – Exemples d'exigences d'interopérabilité, problèmes d'interopérabilité, indicateurs et causes potentielles

Exigences	Problème d'interopérabilité	Indicateur opérationnel	Indicateur statistique	Causes potentielles
Les données reçues sont conformes à celles requises	Format d'échange invalide	. Messages rejetés sur les files d'attente concernées . Format d'échange du dernier message accepté sur les files d'attente	. Taux de messages non conformes . Taux de messages dupliqués	. Format d'échange incompatible . Évolution de format d'échange non communiqué
Les données échangées sont disponibles	Données attendues non disponibles	. Messages perdus sur les canaux concernés (filtrage en fonction des éléments connus du message). . Indicateurs d'interaction manquants	. Taux de messages transmis . Taux de messages perdus . Taux de messages rejetés	. Inutilité / Obsolescence de l'interaction due à un changement de configuration d'architecture ou à un changement de processus . Problèmes liés au manque d'interaction
Les données sont échangées entre les partenaires	. Impossibilité des consommateurs à recevoir plus de messages . Partenaire producteur manquant . Partenaire consommateur manquant	. Ensemble de messages durant la période concerné . Présence de l'utilisateur lié au Producteur . Présence des Exchanges liés au Producteur	. Taux de messages transmis . Taux de messages non consommés . Taux de messages rejetés . Taux de messages perdus	. Problèmes liés au manque d'accusés de réception . Le consommateur n'est pas prêt à recevoir . Disparition des files d'attente (suppression, panne) . Impossibilité de restaurer le nœud contenant les ressources . Arrêt de l'application . Problèmes de permissions

3.2.7 Mise en oeuvre du framework

Nous avons mis en place un prototype d'outil du framework Pulse. Il permet de visualiser certains des comportements du système d'information. Ce prototype a pour objectif de permettre de comprendre le statut de chaque message et de fournir des indications facilitant l'analyse des problèmes d'interopérabilité. Le prototype est implémenté au-dessus de Moose [10], plateforme d'analyse de données [Demeyer 2001, Ducasse 2009] open source, dans lequel nous avons implémenté le métamodèle Pulse. L'extension de Moose est un atout majeur pour notre approche de mise en œuvre, car elle nous permet de nous appuyer sur les outils existants et de les réutiliser.

Le prototype montre la faisabilité et l'intérêt de notre approche. Sa mise en œuvre est composée de toutes les parties du framework (Figure 3.1) : le métamodèle Pulse, le développement des importateurs, la gestion du versionning grâce à Orion [8], la couche de persistance, et quelques visualisations.

L'instanciation avec les informations récupérées du système d'information suit une séquence prédéfinie de manière à construire une cohérence dans l'historique du modèle (Figure 3.4) :

10. http://www.moosetechnology.org/docs/famix

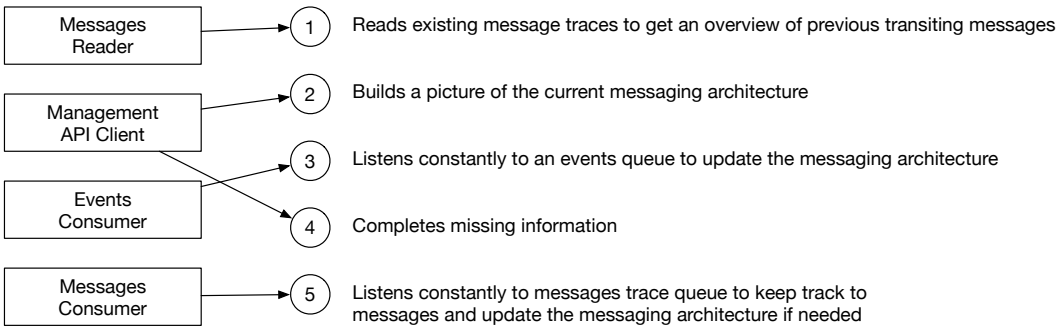

FIGURE 3.4 – Processus de récupération des informations avec Pulse

— Message Reader : il analyse les fichiers journaux des traces de messages et complète le modèle de données. Il est appelé une seule fois au début de la supervision pour donner un aperçu des messages envoyés / reçus dans le passé.

— Management API Client : il utilise l'API REST, si elle existe. Il est appelé une première fois pour compléter les informations manquantes lors de la lecture des fichiers journaux. Il est appelé ensuite à la demande lorsque de nouveaux éléments apparaissent.

— Event Consumer : il écoute en permanence et capture des évènements (évènements tels que la définition d'un nouveau consommateur, la modification des configurations utilisateur, etc.) de la file d'évènements et effectue les modifications sur l'élément correspondant dans le modèle de données.

— Messages Consumer : il écoute en permanence, capture les nouvelles traces de messages et les ajoute au modèle de données dès leur arrivée.

L'approche est expérimentée sur des données générées par une implémentation de RabbitMQ existante de l'entreprise Berger-Levrault. Nous avons fourni des éléments d'analyse en réponse à certains besoins d'évaluation de l'interopérabilité exprimés par l'entreprise :

— Avoir une visualisation claire des traces des messages en transit dans chaque file d'attente et de leurs caractéristiques. Ceci peut être utilisé pour vérifier l'état des messages pendant des périodes définies (Figure 3.5).

— Avoir une vision globale de l'architecture du système d'information et des chemins des messages. Ceci afin d'assurer la présence et l'activité des producteurs et des consommateurs et les interactions attendues entre eux (Figure 3.6).

— S'assurer de l'exactitude des abonnements et de l'absence de fuites de données (Figure 3.6).

3.2.8 Synthèse

Nous avons présenté dans cette partie le moyen de superviser les échanges au sein d'un système d'information distribué, au travers d'une collaboration avec l'entreprise Berger-Levrault. Le framework proposé exploite les services des courtiers de messages pour extraire des informations sur le comportement du système d'information. Nous avons proposé un métamodèle utilisé pour rassembler et agréger les informations recueillies à partir de plusieurs sources. Le peuplement du métamodèle

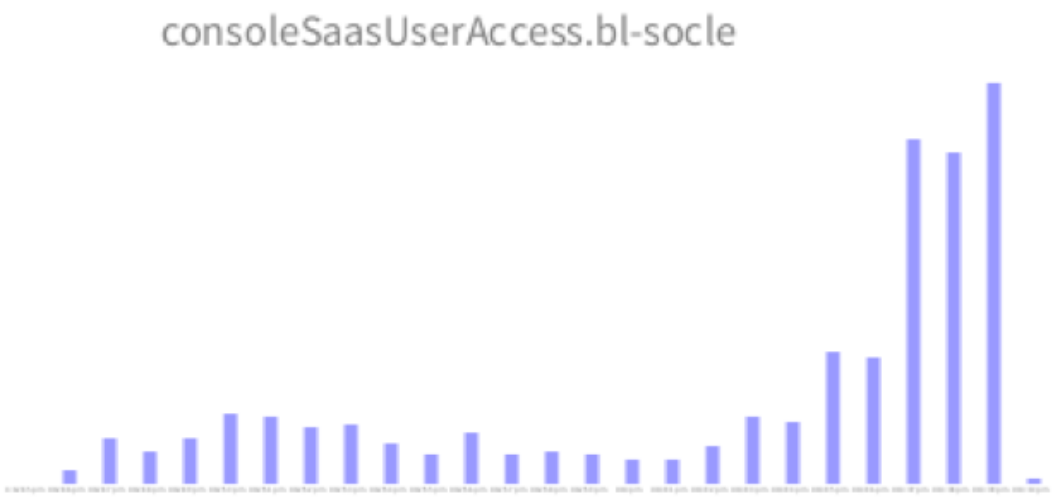

FIGURE 3.5 – Nombre de messages par seconde sur une file d'attente

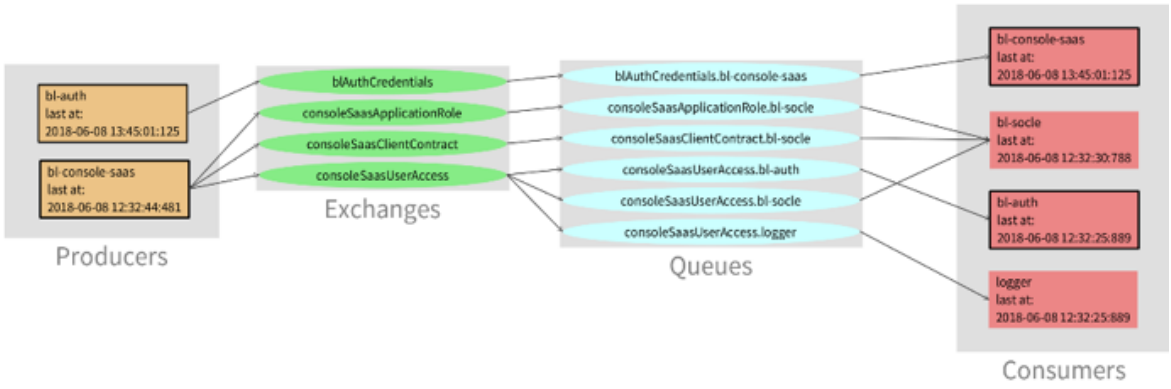

FIGURE 3.6 – Structure du système analysé

permet de fournir des indicateurs contextuels permettant d'évaluer les besoins d'interopérabilité liés aux données et de localiser précisément les problèmes d'interopérabilité.

Nous avons implémenté le métamodèle et les fonctions d'analyse en étendant le métamodèle Moose en profitant de son extensibilité. Nous avons élaboré un importateur et un analyseur de données de traces pour remplir le métamodèle. En réponse à certains besoins d'analyse de Berger-Levrault, nous avons utilisé les données structurées par le métamodèle pour construire deux visualisations de données qui représentent plusieurs indicateurs.

3.3 Gestion des vulnérabilités de la cyber-sécurité à l'aide de réseaux de neurones

3.3.1 Objectifs de recherche

La partie précédente aborde la supervision des données issues des communications au sein d'un système d'information distribué. Pour autant, les systèmes présentés (e.g. AMQP, MQTT, CoAP) révèlent de nombreuses vulnérabilités [McAteer 2017,Maggi 2018]. Ces vulnérabilités peuvent rendre l'architecture du système instable.

Dans cette section, je présente les travaux que nous avons initiés sur le traitement des vulnérabilités. Pour cette recherche, portée par la thèse d'Houssem Gasmi, nous avons appliqué les réseaux de neurones dans la recherche de vulnérabilité dans des bases de données spécialisées dans la diffusion de vulnérabilité des systèmes informatiques. L'objectif de cette recherche est d'évaluer le modèle de réseau neuronal LSTM dans l'extraction d'informations pour ce domaine. Ils nous ont permis de mettre en évidence l'intérêt des réseaux de neurones pour obtenir rapidement l'information des vulnérabilités d'un système d'information.

Il existe plusieurs manières d'informer sur les vulnérabilités, notamment, le Computer Emergency Response Team Coordination Center (CERT/CC) est un centre officiel qui informe publiquement de l'apparition des vulnérabilités. Les personnes qui découvrent des vulnérabilités mettent également ces informations à disposition via plusieurs canaux tels que les blogs, les sites de fournisseurs et les médias sociaux. Plusieurs bases de connaissances ont été établies pour favoriser l'exploitation de cette connaissance par les experts. Des bases de connaissances telles que Common Vulnerability Enumeration [11] (CVE) et National Vulnerability Database [12] (NVD) font référence. Dans ces bases de connaissances, les informations sur les vulnérabilités sont collectées à partir de diverses sources telles que les blogs et les bulletins de sécurité. Les vulnérabilités sont principalement stockées sous forme de texte libre dans une structure plus formalisée et reçoivent généralement un identifiant unique, un type de menace et un niveau de menace.

Le nombre de vulnérabilités de cybersécurité découvertes et signalées augmente chaque année. Par exemple, le Common Vulnerabilities and Exposure a publié un total de 2230 vulnérabilités sur une période de moins de deux mois entre le 1er janvier 2018 et le 19 février 2018.

Parmi les méthodes de lutte informatique défensive (LID), présentées dans le référentiel du pôle d'excellence CYBER [CYBER 2019], la veille active des menaces externes sur les sites d'information non officiels reste un verrou. D'après ce référentiel, l'utilisation de l'intelligence artificielle pour cette veille est en forte croissance et "va prendre une importance majeure dans les années à venir".

La collecte et le traitement en temps réel d'informations sur la cybersécurité à partir de diverses sources web en ligne, telles que les actualités, les bulletins des fournisseurs, les blogs, les forums et les bases de données en ligne, est essentielle pour de nombreux types d'applications. Pour fournir une gestion des vulnérabilités, nous devons extraire les connaissances de ces différentes sources et les présenter dans un format qui permet aux professionnels de la sécurité de prendre rapidement conscience des vulnérabilités. La modélisation des connaissances des vulnérabilités peut simplifier le travail des analystes [Piskorski 2013] pour la détection des intrusions et la connaissance de situations

11. https://cve.mitre.org/
12. https://nvd.nist.gov/

à problèmes [Joshi 2013].

La description principale de la vulnérabilité est toujours en texte libre. Par conséquent, il est nécessaire de s'appuyer sur des techniques de traitement du langage naturel pour transformer les informations textuelles non structurées en une base de connaissances structurées. En les représentant comme une ontologie, cela permet de nombreuses applications telles que la recherche sémantique, la navigation et la visualisation, etc.

Les deux tâches principales de l'extraction d'informations sont la reconnaissance d'entités nommées (Named-entity recognition – NER) et l'extraction de relations (RE).

Les principales méthodes d'extraction utilisées pour le NER s'appuient sur des modèles statistiques et les entités sont décrites grâce à l'ingénierie des fonctionnalités. Elles donnent de bons résultats. Cependant, les outils sont généralement spécifiques au domaine, ce qui présente certains inconvénients. Tout d'abord, il a besoin d'une personne qui possède une connaissance approfondie du domaine et passe du temps à définir les caractéristiques du texte par un long processus d'essais et d'erreurs. Deuxièmement, ce travail d'ingénierie s'appuie sur des dictionnaires ou des recherches pour identifier les entités du domaine [Mulwad 2011]. Ces dictionnaires sont difficiles à créer et à entretenir, en particulier dans les domaines fortement dynamiques, tels que la cybersécurité.

D'autre part, le but de l'extraction de relations (RE) est d'extraire des connaissances sur des entités liées à partir de sources non structurées telles que du texte et de représenter ces connaissances sous la forme d'un triplet qui a la forme : (sujet, prédicat, objet). Par exemple, à partir de la description de la vulnérabilité de la Figure 3.7, nous pouvons extraire le tuple (IBM, is_vendor_of, webSphere), qui est une instance du type de relation : (fournisseur de logiciels, is_vendor_of, produit logiciel).

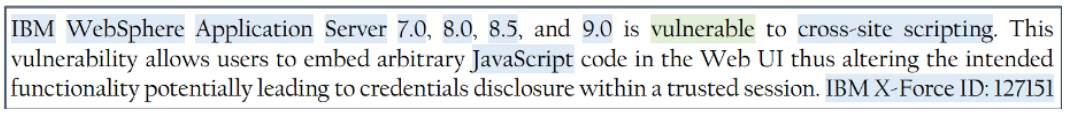
IBM WebSphere Application Server 7.0, 8.0, 8.5, and 9.0 is vulnerable to cross-site scripting. This vulnerability allows users to embed arbitrary JavaScript code in the Web UI thus altering the intended functionality potentially leading to credentials disclosure within a trusted session. IBM X-Force ID: 127151

FIGURE 3.7 – Exemple de description de vulnérabilité

Les travaux précédents ont montré que les outils de traitement automatique du langage naturel (Natural Language Processing – NLP) standard ne sont pas en mesure d'extraire les entités liées à la sécurité et leurs relations correspondantes avec une précision satisfaisante [Liao 2016, More 2012].

Une catégorie de réseaux de neurones appelés réseaux de neurones récurrents (RNN) a obtenu de bons résultats dans le domaine de NLP [Gers 1999]. Les RNN ont la capacité de détecter et d'apprendre à partir d'une séquence de données. Bien que les RNN fonctionnent bien avec une séquence courte, ils souffrent d'un problème appelé problème de classement lorsque la séquence traitée devient trop longue.

Parmi les RNN, les réseaux de neurones LSTM (Long short-term memory), utilisant des méthodes d'apprentissage en profondeur, sont devenus l'alternative privilégiée pour le traitement de texte [Huang 2015, Chiu 2016, Ma 2016].

L'objectif de notre recherche est de concevoir un framework qui fournit des informations à jour et structurées sur les vulnérabilités logicielles en utilisant les réseaux de neurones LSTM.

Le principal problème de recherche abordé est de savoir comment concevoir et construire un réseau de neurones LSTM pour effectuer des tâches d'extraction d'informations du domaine de la gestion des vulnérabilités en cybersécurité et de les comparer aux méthodes traditionnelles. Par consé-

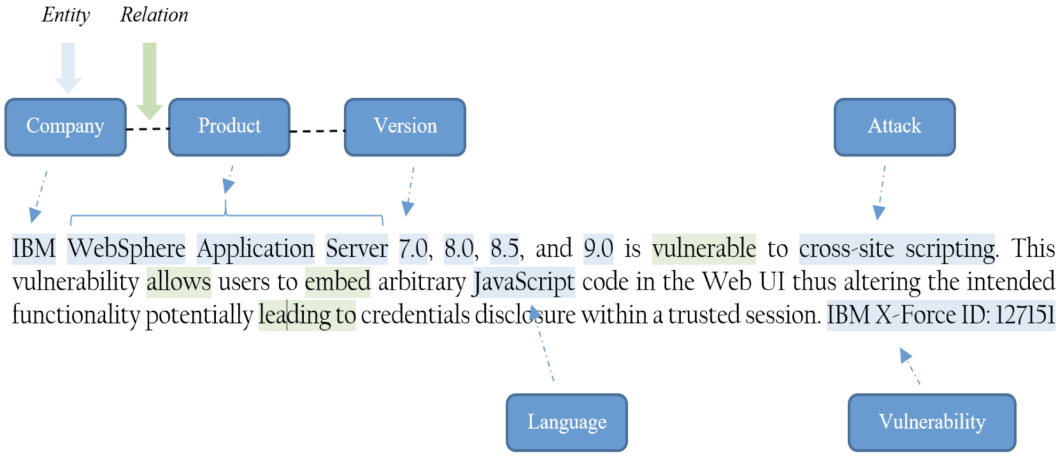

FIGURE 3.8 – Principe de la reconnaissance d'entités nommées et de l'extraction de relations

quent, nos objectifs de recherche sont les suivants :

— Concevoir un framework d'apprentissage basé principalement sur le développement de réseaux de neurones spécialisés dans le traitement de texte naturel.

— Obtenir ou construire un corpus d'apprentissage pour le domaine de la cybersécurité en utilisant des techniques de NLP et l'intégrer dans LSTM.

— Construire un modèle basé sur LSTM pour la reconnaissance d'entités nommées (NER) des documents textuels du domaine de la cybersécurité.

— Évaluer le modèle LSTM pour NER en comparaison avec le modèle basé sur CRF(Conditional Random Field) pour déterminer si les modèles LSTM conviennent à la tâche.

— Construire des modèles basés sur LSTM pour extraire les relations entre les entités de cybersécurité à partir de documents textuels.

— Évaluer les modèles LSTM populaires pour l'extraction de relations afin de déterminer le plus approprié pour le domaine de gestion des vulnérabilités.

Pour évaluer les performances des LSTM pour les tâches NER, nous avons comparé les performances d'un modèle basé sur LSTM et d'un modèle basé sur CRF (Conditional Random Field) sur le même corpus de descriptions de vulnérabilité collectées à partir de diverses sources en ligne. CRF est un modèle statistique particulièrement utilisé pour le traitement du langage naturel.

Pour l'extraction des relations, nous avons comparé trois architectures LSTM récentes en termes de performances. Ces architectures sont "LSTM along shortest dependency paths", "LSTM on least common ancestor sub tree", et "LSTM on sequences and tree structures". Les modèles sont comparés en termes de précision, de rappel et de scores F1.

3.3.2 Le framework

Le principal objectif du framework, présenté dans l'article [12], consiste à extraire les informations des descriptions de vulnérabilités. L'extraction d'informations comprend l'extraction d'entités

de domaine ou Named Entity Recognition (NER) et l'extraction de relations (RE). Le framework utilise une ontologie pour identifier les vulnérabilités.

En supposant que nous avons une ontologie vide qui modélise le domaine des vulnérabilités, nous voulons la remplir à partir de ressources non structurées. La tâche consiste à construire les instances dans l'ontologie et les relations sémantiques entre elles à partir d'un texte décrivant une vulnérabilité dans un logiciel. Étant donné une entrée sous la forme d'une séquence de mots, le modèle doit prédire la classification la plus probable des mots et extraire les relations entre les mots pour le domaine.

Le texte principal décrivant une vulnérabilité dans un logiciel n'est pas structuré. Certaines bases de données telles que NVD ajoutent des informations structurées à la description de la vulnérabilité comme un identifiant, le niveau de gravité et la date d'identification de la vulnérabilité. Ces informations ne sont actuellement pas utilisées dans nos algorithmes.

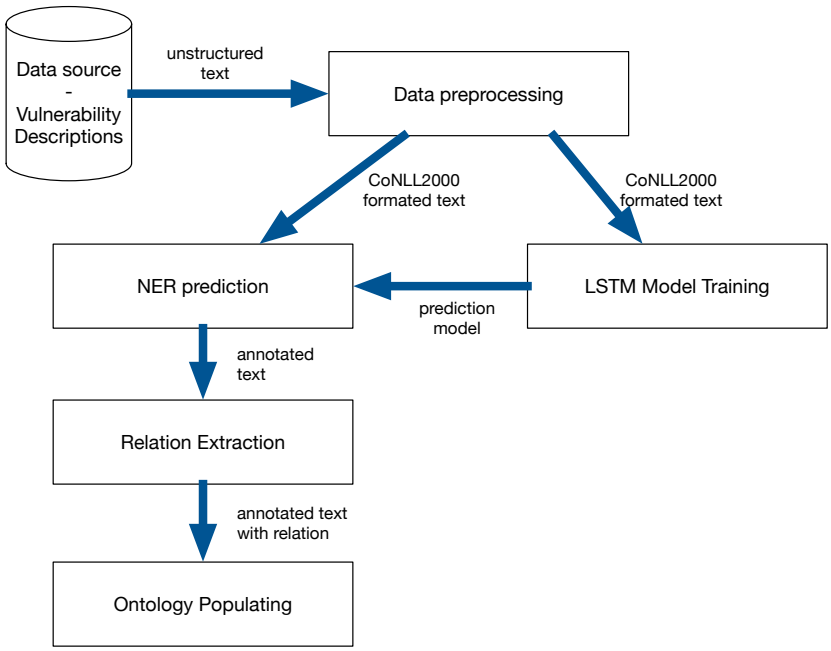

FIGURE 3.9 – Framework d'identification des vulnérabilités

La Figure 3.9 décrit le framework proposé qui convertit les descriptions des vulnérabilités en une représentation structurée sous la forme d'une ontologie. Le framework se compose principalement des parties suivantes :

— L'apport d'une source de données,

— La préparation du texte en l'extrayant de la base de données NVD puis en le formatant au format CoNLL2000 [Sang 2000] et en le convertissant en représentation "Word2Vec" de mots selon les besoins du réseau de neurones LSTM.

— L'entraînement du modèle de prédiction qui sera capable d'extraire des entités du texte (Named entity Recognition - NER). Pour entraîner le modèle, le corpus traité par NVD est divisé en 3 parties, 70% est utilisé pour l'entraînement du modèle, 10% est utilisé comme ensemble de validation et 20% est utilisé pour évaluer le modèle. Une fois le modèle entraîné, nous obtenons un modèle de prédiction qui peut être utilisé pour extraire les entités du texte.

— L'extraction de relations (RE) prend le texte annoté et attribue une relation entre les entités détectées à partir d'une liste prédéfinie de relations potentielles.

— Une fois les entités et les relations extraites, l'ontologie est remplie avec les informations extraites

3.3.2.1 Définition de l'ontologie

En tant que preuve de concept, nous avons conçu une ontologie simple, visible dans la Figure 3.10. Cette ontologie des vulnérabilités se compose des classes représentant le domaine et leurs relations ; elle est décrite en utilisant le langage web sémantique OWL pour représenter les connaissances décrivant les entités du domaine et la relation entre elles :

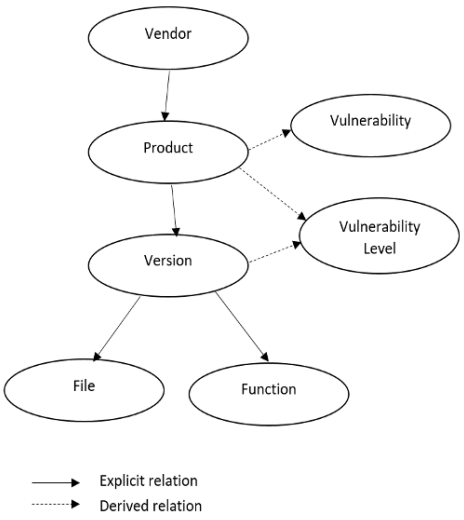

FIGURE 3.10 – Ontologie utilisée pour identifier les vulnérabilités

— Fournisseur (Vendor). Un fournisseur de logiciel.

— Produit. Un logiciel développé par un fournisseur qui peut être une application ou un système d'exploitation.

— Version. La ou les versions du produit affectées par la vulnérabilité.

— Fichier. Un fichier qui fait partie d'un produit et qui est lié à la vulnérabilité décrite.

— Fonction. Fonction faisant partie d'un produit et liée à la vulnérabilité décrite.

— Vulnérabilité. Identifiant unique de la vulnérabilité correspondant à l'ID CVE de NVD.

— Niveau de vulnérabilité. Un niveau de «faible», «moyen» et «élevé» sur le niveau de criticité en termes de vulnérabilités dans un produit.

Deux types de relations existent dans l'ontologie. Le premier type représente les relations explicites que nous avons définies dans l'ontologie, elles sont extraites du texte. Le deuxième type est constitué de relations implicites dérivées de règles d'inférence. L'ontologie définit ces règles à l'aide du langage SWRL pour déduire plus de connaissances à l'aide d'un moteur d'inférence. Par exemple, nous pouvons identifier que si le produit d'un fournisseur spécifié présente une vulnérabilité dans un

fichier, cela implique que le produit est vulnérable. Un autre exemple est de pouvoir définir qu'une version d'un produit a un niveau de vulnérabilité élevé si elle comporte au moins cinq fonctions vulnérables.

3.3.2.2 NER (Named Entity Recognition) avec LSTM

Le NER a pour objectif la détection des entités pertinentes du domaine, telles que le produit concerné par la vulnérabilité, la version, le nom de l'attaque, etc. Nous avons utilisé l'architecture LSTM-CRF telle que présentée par Lample et al. [Lample 2016].

La Figure 3.11 montre l'architecture de LSTM-CRF [16]. Il se compose de trois couches. La première couche de l'architecture prend la séquence d'entrée des mots et en produit un vecteur de mots. La couche suivante est la couche LSTM. Elle utilise un algorithme LSTM bidirectionnel. Le LSTM bidirectionnel est efficace dans des applications de NER [Huang 2015]. La dernière couche est une analyse CRF [Lou 1995]. Le résultat de cette architecture est une instanciation de la classe ayant la probabilité la plus élevée pour le mot. Dans le cas de la Figure 3.11, le texte "Apple QuickTime before 7.7" retourne Apple comme instance de la classe "Vendor", QuickTime comme instance de la classe "Application", et 7.7 comme instance de la classe "Version". L'algorithme est expliqué en détail dans l'article [16].

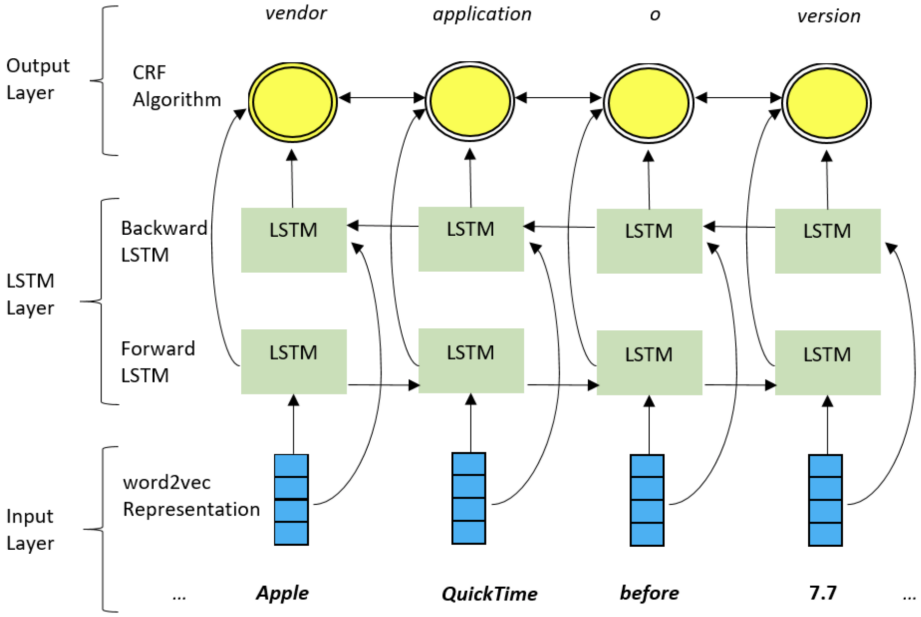

FIGURE 3.11 – Fonctionnement de l'architecture LSTM-CRF

Nous avons réalisé une comparaison du modèle LSTM-CRF avec le modèle CRF sur des données de cybersécurité. Pour l'évaluation, nous avons utilisé un corpus de plusieurs sources, principalement de NVD, contenant 40 types d'entités. Nous avons analysé les performances moyennes des deux modèles par rapport à l'ensemble des types d'entités, puis nous avons considéré uniquement les entités les plus fréquentes du domaine et avons évalué les modèles sur cet ensemble.

Le modèle LSTM-CRF a atteint une précision globale supérieure au modèle CRF (voir Table 3.2 et Table 3.3). Le modèle LSTM-CRF a donné de meilleurs résultats pour cinq artefacts et le modèle CRF

a eu de meilleures performances pour deux artefacts. L'explication probable de ce comportement est que les identifiants de vulnérabilité (Vulnerability ID) ont un format spécifique, tel que CVE-2019-12623, ce qui facilite l'apprentissage du modèle LSTM tandis que la balise "Edition" est un mot anglais utilisé dans différents contextes. Les détails de l'analyse se trouvent dans l'article [16].

Les résultats que nous avons obtenus ont montré que le modèle LSTM-CRF est le plus adapté compte tenu de notre domaine d'application [1].

TABLE 3.2 – Performance pour l'ensemble des entités

	Précision (en %)	**Rappel (en %)**	**Score F1 (en %)**
LSTM-CRF	85.16	80.70	83.37
CRF	80.26	73.55	75.97

TABLE 3.3 – Détails des performances des modèles LSTM-CRF et CRF

Entité	**Score F1 (en %)**		**Précision (en %)**		**Rappel (en %)**	
	LSTM-CRF	**CRF**	**LSTM-CRF**	**CRF**	**LSTM-CRF**	**CRF**
Vendor	93	92	94	94	92	90
Application	89	87	89	88	90	86
Version	98	95	98	95	98	95
Edition	60	80	76	87	50	75
OS	95	93	97	95	93	91
Hardware	46	63	57	79	39	52
File	99	84	1	85	99	84
Average	82.8	84.4	87.2	89	80.1	81.8

Suite à cette étude, nous avons fait varier les paramètres du modèle LSTM-CRF de manière à sélectionner la meilleure configuration pour le domaine de la cybersécurité [12]. Le tableau Table 3.4 montre la configuration utilisée.

TABLE 3.4 – Paramètres du modèle LSTM-CRF

Paramètre	**Valeur**
CRF	True
Lowercase words	False
Replace digits with 0	False
LSTM hidden layer size	100
Use a bidirectional LSTM for words	True
Dropout	0.5
Learning method	SGD

La Table 3.5 montre les variations de paramètres que nous avons effectuées et les résultats en termes de performance sur un ensemble de données complet.

TABLE 3.5 – Résultats des variations des paramètres du modèle

Paramètre	Valeur	Précision (en %)	Rappel (en %)	Score F1 (en %)
Valeurs par défaut		84	77	79
CRF	False	53	53	53
Lowercase words	True	82	79	79
Replace digits with 0	True	83	83	83
Hidden layer size	25	81	79	79
Hidden layer size	50	83	78	78
Use a bidirectional LSTM for words	False	53	54	54
Dropout	0	87	82	83
Leaning method	adam	87	86	84
Leaning method	adadelta	81	76	77

Les deux paramètres qui ont eu le plus d'impact sur les performances du modèle sont le CRF et "Bidirectional LSTM". Lorsque nous avons désactivé ces paramètres, le score F1 a considérablement chuté. Les résultats montrent également que l'algorithme d'apprentissage "adam" a obtenu de meilleures performances que les autres algorithmes. La variation des paramètres restants n'a pas eu un grand impact sur les performances du modèle.

3.3.2.3 RE avec LSTM

Pour la tâche RE, nous avons implémenté trois modèles pour identifier celui qui serait le plus adapté : l'architecture "LSTMs along Shortest Dependency Paths (SDP)" à la suite des travaux de Yan Xu et al. [Xu 2015], l'architecture 'LSTMs on Sequences and Tree Structures (STS)" et l'architecture "LSTM on the Least Common Ancestor Sub Tree (LCA)". Ces deux dernières architectures sont présentées dans l'article de Miwa et al [Miwa 2016]. Cette étude est décrite dans l'article [1].

TABLE 3.6 – Performance des trois modèles d'extraction des relations

Modèle	Précision de l'entraînement (en %)	Précision de la validation (en %)	Précision (en %)	Rappel (en %)	Score F1 (en %)
LSTM on Shortest Dependency Path	98.4	90.9	92.2	91	94.3
LSTM on LCA Sub Tree	74	87.5	80	83.1	82.1
LSTM on Sequence and Tree Structures	98.7	76.8	73	77	75

Les résultats obtenus pour les trois modèles varient considérablement. Les performances moyennes des trois modèles RE sont présentées dans le tableau Table 3.6. Nous pouvons voir que les mesures de performance en termes de précision, de rappel et de score F1 montrent que le modèle SDP est meilleur que les deux autres modèles. Il a été performant à la fois dans l'entraînement et la précision de la validation tout en obtenant le meilleur score F1 avec 94,3%. Le modèle STS a bien fonctionné à l'entraînement, mais n'a pas obtenu un score F1 élevé lors de la phase d'utilisation. Le score du modèle LCA était moyen en termes d'entraînement et de validation.

Par ailleurs, en terme de temps d'entraînement, le modèle STS a mis 21 minutes en moyenne à s'entraîner, tandis que le modèle SDP a été très rapide avec des temps de seulement 50 secondes. Ce phénomène est dû à la complexité plus importante du modèle STS.

3.3.3 Synthèse

Ce travail de recherche fournit un framework permettant d'extraire et de classifier les informations des descriptions des vulnérabilités disponibles dans les bases de connaissances.

Ce framework est construit sur l'utilisation de réseaux neuronaux de type LSTM. D'une part cette architecture permet d'extraire les entités des informations recueillies, et d'autre part elle permet de relier ces informations à l'ontologie.

Pour atteindre cet objectif, nous avons été amenés à étudier les différentes implémentations de LSTM pour choisir le plus optimal dans notre framework.

Évidemment, il y a des menaces à la validité de l'étude. L'évaluation est un problème majeur ; en ce qui concerne l'extraction des relations, il manque un corpus de référence dans le domaine de la gestion des vulnérabilités de cybersécurité qui permettrait la comparaison des différents algorithmes et assurerait la répétabilité de la comparaison. Créer un corpus suffisamment grand pour convenir aux algorithmes d'apprentissage en profondeur, annoté à la main par des professionnels du domaine, permettrait d'obtenir des études comparatives plus viables.

Cette étude des réseaux de neurones LSTM ouvre des perspectives importantes dans la supervision des systèmes d'information distribués. Il sera important dans un futur proche de rapprocher ces travaux de nos travaux présentés dans la Section 3.2. D'une part, nous comptons intégrer la recherche de vulnérabilité pour améliorer la stabilité du système d'information, en ciblant les vulnérabilités importantes. D'autre part, nous comptons utiliser les réseaux LSTM pour identifier les causes des problèmes du système d'information. Par exemple, pour un problème d'interopérabilité donné, le framework permettrait de trouver rapidement la cause.

3.4 Bilan de l'axe et perspectives

La supervision des systèmes d'information distribués implique d'intégrer l'aspect dynamique de l'architecture des systèmes d'information et la gestion de phénomènes imprévisibles, comme des problématiques de sécurité des constituants. La collecte d'informations provenant d'une part du système et d'autre part des vulnérabilités techniques des constituants est une première étape dans le maintien en exploitation d'un système de systèmes.

Le principal problème que j'ai abordé dans cet axe est de structurer l'information. Dans une première partie, j'ai structuré l'information provenant d'une architecture dynamique et je l'ai organisée dans un métamodèle en considérant la dimension temporelle. Dans une deuxième partie, j'ai travaillé sur l'extraction de données à partir d'informations non structurées de manière à en tirer une information pertinente grâce à l'implémentation de réseaux de neurones LSTM.

Ces travaux ont nécessité la mise en oeuvre de divers domaines, notamment la métamodélisation de données, l'utilisation d'ontologie et de raisonnement. Ils ne présentent évidemment qu'une partie du maintien en fonctionnement des systèmes d'information distribués. Nous avons identifié plusieurs verrous aux approches proposées :

— Nous avons réalisé les études dans deux contextes différents : un contrat de collaboration et une thèse. Or, il serait intéressant d'intégrer les deux approches pour compléter la supervision d'un système d'information distribué. Cela nous permettrait d'intégrer la sécurité des SI dans les causes potentielles des problèmes des systèmes.

— Dans l'état actuel des travaux proposés de la supervision des échanges de données, nous parvenons à détecter des évènements individuels pouvant indiquer des problèmes d'interopérabilité. Nous fournissons également un ensemble de causes potentielles. La recherche des causes réelles nécessite une analyse plus approfondie. Nous considérons des méthodes analytiques telles que le traitement d'évènements complexes [Fülöp 2010] pour analyser les corrélations dans de grandes quantités d'évènements afin de détecter les dysfonctionnements et leurs causes.

— LSTM s'est révélé être une approche intéressante pour la cybersécurité. Nous souhaitons étendre cette approche pour analyser les risques d'arrêt d'activité suite à une cyberattaque ou une modification malveillante du processus métier.

3.5 Synthèse des contributions

Chacune des deux problématiques a fait l'objet de plusieurs publications.

La section 3.2 a été présentée dans :

— [46] : présentation du cas d'étude industriel et premières briques du framework Pulse.

— [19] : première présentation du framework Pulse.

— [43] : extension du papier précédent dans un chapitre de livre.

— [60] : première présentation du modèle d'analyse des problèmes d'interopérabilité.

— [11] : présentation du framework Pulse et du métamodèle générique.

La section 3.3 a été présentée dans :

— [16] : présentation des premiers résultats de l'architecture LSTM-CRF sur l'analyse de vulnérabilités.

— [1] : présentation du framework et comparaison des différentes architectures LSTM.

— [12] : présentation des résultats du framework global.

CHAPITRE 4

Axe 3 : Contribution à l'outillage pour l'évolution des systèmes cyberphysiques

Contents

4.1 Contexte 56
4.2 Plateforme de simulation de CPS : le cas d'une architecture multirobot 58
4.3 Méthodologie de tests pour les CPS : application aux robots mobiles 64
4.4 Méthodologie de co-conception logiciel / FPGA 71
4.5 Bilan de l'axe et perspectives 76
4.6 Synthèse des contributions 77

Les montagnes n'appartiennent à personne, c'est bien connu, mais les expériences appartiennent à chacun. Beaucoup d'autres peuvent grimper sur les montagnes, mais personne ne pourra jamais s'emparer des expériences qui sont et demeurent nôtres.
[Walter Bonatti]

En quelques mots...

Ce chapitre introduit l'Axe 3 : Contribution à l'outillage pour l'évolution des systèmes cyberphysiques. Dans cet axe, j'aborde les méthodes de validation et de vérification (V&V) que j'ai mises en place pour assurer l'évolution de systèmes d'information distribués. Les travaux de ce chapitre sont appliqués à la robotique de service, faisant partie intégrante du domaine des CPS. Nous abordons, dans ce chapitre, trois approches : la réalisation d'un banc de simulation dans lequel nous insistons sur la reproductibilité des simulations, une méthodologie de tests dédiés aux CPS permettant de valider le comportement des constituants, et une méthodologie de co-conception logiciel / FPGA dans laquelle nous avons proposer la possibilité de construire, simuler, écrire et déboguer un circuit FPGA à partir d'un environnement de programmation orienté objet.

Ces travaux ont été présentés dans les publications [3, 9, 26, 31, 33, 34, 49, 51, 52, 62–64, 66, 66].

4.1 Contexte

La notion de système cyberphysique (CPS) nous permet de considérer l'intégration de systèmes physiques / analogiques avec des solutions logicielles informatiques / numériques. Ces systèmes sont eux-mêmes intégrés dans des systèmes plus larges et complexes (systèmes de systèmes). Avec la construction de ces nouveaux systèmes, émergent de nouvelles propriétés ou contraintes (architectures reconfigurables, dynamicité de l'environnement, évolution des usages), en plus des propriétés classiques, telles que la sûreté et la sécurité.

Dans un système de systèmes, pouvoir assurer l'évolution, piloter les transformations et gérer la maintenance tout en garantissant la continuité de service reste un défi important. Ces systèmes sont distribués, autonomes, sujets au changement dans leur environnement. L'intégration du matériel amène des défis supplémentaires de l'ordre de la reproductibilité des tests, de la validité des résultats ou encore du processus de débogage à distance.

Un défi majeur consiste à revoir les méthodes de Validation et de Vérification (V&V) pour prendre en compte ces nouvelles architectures et les nouvelles propriétés associées. Une des difficultés est que les techniques actuelles de V&V nécessitent la connaissance de la logique applicative pour garantir la qualité avant déploiement. Elles s'appuient sur des spécifications de référence, qui n'anticipent, a priori, pas les évolutions environnementales, les fonctionnalités émergentes ou les nouveaux usages. Garantir la qualité de ces systèmes reste un défi important. Une des pistes est de s'appuyer sur un processus de validation dynamique, pendant le déploiement, par exemple sur la forme de tests passifs ou de monitoring [Dadeau-FEMTO-ST 2014].

La simulation peut surmonter une partie des problèmes ci-dessus. Cependant, l'utilisation seule d'un simulateur ne suffit pas pour analyser les systèmes de CPS. Chaque simulation doit être reproductible afin de permettre de comparer les approches. La reproductibilité implique qu'il n'y a pas d'hypothèses cachées. De plus, la simulation doit être aussi réaliste que possible, car des repères sont définis pour évaluer les applications. Par exemple, la simulation d'un grand nombre de robots sur un seul ordinateur physique masque le fait de la consommation de bande passante réseau pour la communication entre robots. Nous discuterons de ce point précis dans la Section 4.2.

Parmi les travaux reposant sur la simulation, certains introduisent le matériel dans la boucle (hardware in the loop). Par exemple, Chen et al. [Chen 2011] proposent d'insérer une étape supplémentaire pour les tests hybrides entre simulation et tests dans le monde réel. La même idée a été poussée par Petters et al. [Petters 2008] qui présentent des stratégies de test multiniveaux pour des équipes de robots autonomes. Les auteurs proposent trois niveaux de tests : les tests de composants (automatisés), les tests en ligne (avec le jugement d'un humain) et les tests hors ligne (à l'aide de journaux d'activités). Chacun de ces niveaux peut être exécuté avec des robots simulés ou avec des robots réels.

De nombreuses approches utilisent la simulation pour tester le logiciel avant de le brancher sur le matériel. Par exemple, SITAF [Seong 2012] est un framework pour tester les composants du robot en simulant l'environnement. Il génère des cas de test basés sur une spécification donnée par le développeur. Cette génération de tests combinée à des simulations permet la répétabilité des tests. Il élimine également le problème de réutilisation des tests, car ils sont générés. Par ailleurs, travailler au niveau de la simulation permet de détecter des défauts sans prendre de risque ni pour l'homme ni pour le robot.

Parmi les travaux s'appuyant sur des tests, Son et al. [Son 2011] propose 3 niveaux de tests : les tests unitaires, les tests d'état et les tests d'API. Les 3 niveaux permettent d'évaluer les performances

des composants. Le système présenté permet de valider la sécurité des composants testés en mesurant leur durabilité par exemple. Koo Chung et al. [Chung 2007] propose une ligne directrice pour adapter la norme ISO 9126 pour les tests de logiciels aux composants pour la robotique. Plusieurs articles [Biggs 2010, Lim 2010] proposent des méthodes de tests dédiés aux composants du robot. Bien que ces approches puissent être réalistes à différents niveaux, des tests avec des CPS réels sont toujours nécessaires.

Par ailleurs, les applications développées pour les CPS se retrouvent déployées sur des systèmes hybrides (réseau / logiciel / matériel). Les moyens de les concevoir et de les déboguer restent une tâche difficile. Cette tâche nécessite de faire appel à un vaste domaine d'expertise allant du logiciel de haut niveau au matériel de bas niveau (par exemple les FPGA). Il en résulte un flux de conception complexe, qui implique l'utilisation d'outils issus de différents domaines d'ingénierie. Une solution est de proposer un environnement de conception hétérogène qui combine/intègre l'ensemble de ces outils. Cependant, l'hétérogénéité intrinsèque de cette approche peut compromettre la fiabilité du système et nécessite de mettre en place des outils de gestion de déploiement et de débogage.

J'ai choisi d'aborder dans ce dernier axe le développement d'outils pour assurer la validation et la vérification des CPS au sein de SdS. Ces travaux sont illustrés par 3 approches :

La simulation de CPS : La Section 4.2 décrit mes travaux dans la réalisation d'un banc de simulations de flottes multirobots. Ici nous cherchions à valider des algorithmes de gestion de flottes. Ce travail est le résultat de l'encadrement d'un postdoctorant (Zhi Yan). Nous avons montré la nécessité de réaliser des simulations reproductibles et nous proposons un banc de simulation permettant l'explicitation des hypothèses de la simulation.

Le développement de Tests pour les CPS : La Section 4.3 décrit mes travaux dans le développement d'une stratégie de tests dédiée aux CPS et du framework associé. Ce travail est le résultat de mes recherches au sein de l'école des Mines de Douai. Nous avons abordé la difficulté de tester, déboguer et maintenir une application déployée sur un CPS.

La co-conception logiciel / FPGA : La Section 4.4 décrit mes travaux dans la co-conception d'architectures logicielles orientées objet et d'architectures FPGA. Ce travail est le résultat de la thèse de Sang Le. Nous nous sommes intéressés notamment à la partie simulation et aux tests d'architecture FPGA grâce à un framework dédié. La génération et le déploiement du circuit sur le FPGA permettent d'envisager, à terme, l'évolution à chaud des architectures distribuées.

4.2 Plateforme de simulation de CPS : le cas d'une architecture multi-robot

4.2.1 Objectifs de recherche

La communauté scientifique a travaillé au développement de simulateurs capables de reproduire avec précision des environnements du monde réel, dans lesquels nous pouvons faire évoluer des représentations de robots et évaluer leur comportement. Certains projets représentatifs sont Stage [Gerkey 2003], Webots [Michel 2004], V-REP [Freese 2010], Gazebo [Koenig 2004], et MORSE [Echeverria 2012].

Ces environnements simulés aussi réalistes que possible sont nécessaires pour comparer différents algorithmes dans différentes conditions et obtenir des résultats fiables. Des bancs de simulations pour les systèmes multirobots à grande échelle ont déjà été réalisés, par exemple [Michael 2008] ou encore RoboNetSim [Kudelski 2013]. Ces bancs de tests visent la vérification expérimentale et la validation de différentes applications robotiques distribuées dans un environnement simulé. Dans ces approches, bien souvent, peu d'informations sont fournies concernant le processus d'analyse comparative et les défis que l'on doit relever lors de la construction d'un tel banc de simulation.

L'analyse comparative des systèmes multirobots est cruciale pour évaluer et comparer différentes stratégies de coordination. L'article [Faigl 2015] propose d'aborder le problème de recherche et sauvetage multirobots et a évalué cinq algorithmes pour des tâches d'exploration coopérative avec un nombre différent de robots. L'article [Couceiro 2014] propose une méthodologie pour évaluer l'exploration multirobots basée sur les frontières et a également comparé cinq méthodes de répartition des tâches utilisées dans le problème d'exploration. La méthodologie proposée comprend un cadre d'exploration, des repères et un protocole expérimental.

La principale exigence pour un banc d'essai est l'automatisation du processus d'analyse. Il commence par une conception de la simulation définie par l'utilisateur, dans laquelle ce dernier doit spécifier la valeur des paramètres et les données à collecter. Ensuite, les expériences sont effectuées automatiquement par le banc de simulation, sans aucune intervention manuelle. Enfin, les données collectées doivent être traitées afin de calculer les valeurs pour les métriques définies et de comparer divers paramètres (c'est-à-dire les valeurs des paramètres).

Également, un banc de simulation pour des systèmes multirobots doit être reproductible de manière à pouvoir reproduire rapidement des expériences et valoriser la comparaison des algorithmes. Notre banc de simulation est développé dans le but de fournir à la communauté un cadre reproductible pour comparer facilement différents algorithmes pour des systèmes multirobots.

Une autre exigence importante est la réutilisabilité du banc de simulation. Ce dernier doit être le plus indépendant possible de ses applications. Nous y parvenons en utilisant un middleware considéré comme standard.

Dans cette section nous apportons les contributions suivantes :

— l'architecture d'un banc de simulations pour l'analyse de systèmes multirobots

— la capacité de reproductibilité par l'explicitation des paramètres, des métriques et le nombre d'analyses à réaliser.

— une description des limites de l'architecture du banc de simulations.

Dans le cadre de notre expérimentation, nous avons déployé un système multirobot pour analyser les différents algorithmes d'exploration multirobot. Nous avons développé une architecture robotique à cet effet que nous avons ensuite utilisée dans le simulateur.

4.2.2 L'architecture utilisée

L'architecture de notre banc de simulations est composée de quatre parties : un simulateur, un moniteur, un ensemble de robots et le middleware ROS [Quigley 2009] utilisé pour les connecter. L'architecture est illustrée dans la Figure 4.1.

— Le middleware ROS : c'est un système open source dédié à la robotique, aujourd'hui reconnu comme une plateforme logicielle standard et largement répandue dans la communauté scientifique.

— Le simulateur : Nous utilisons le simulateur MORSE 3D. MORSE est un simulateur open source et indépendant de toute architecture robotique et de cadre de communication. Il est implémenté au-dessus d'un moteur physique 3D. Nous l'avons intégré dans un nœud ROS. Ceci permet d'une part de le connecter à l'environnement ROS, et d'autre part de pouvoir le remplacer par un autre simulateur.

— Le moniteur : il est également inclus dans un nœud ROS, et nous permet de superviser les processus. Plus précisément, il collecte les données et réalise les mesures associées. Il peut arrêter l'expérimentation lorsque la condition d'arrêt est déclenchée.

— Les robots : chaque robot est représenté par un contrôleur matérialisé par un ensemble de noeuds ROS. La structure d'un robot est détaillée dans la Section 4.2.4.

— La stratégie de coordination : elle est portée par le système multirobot. Chaque robot a la capacité de décider de ses actions en fonctions des différentes informations reçues des autres robots. La stratégie de coordination est la partie que nous souhaitons tester dans le cadre de la réalisation de ce banc de simulation.

Le banc de simulations a été déployé sur un cluster informatique pour prendre en charge la simulation de robots à grande échelle.

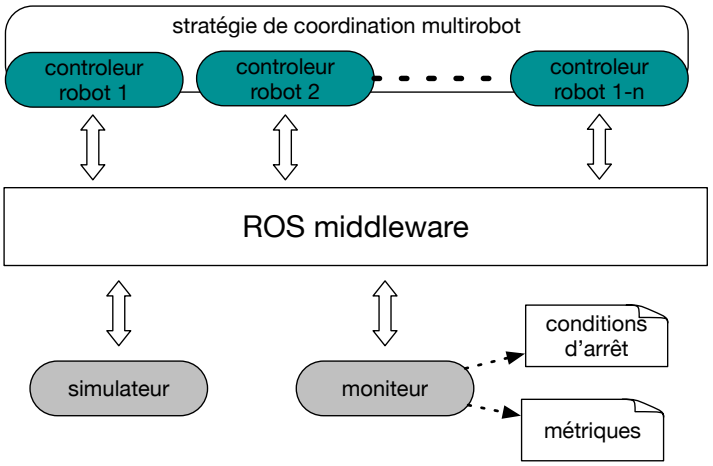

FIGURE 4.1 – Architecture du banc de simulations

4.2.3 La configuration d'une simulation

La connaissance de la configuration du banc de simulation est un des problèmes pour la reproductibilité de l'expérimentation. Nous avons défini un fichier de configuration (un exemple est visible dans la Figure 4.2) contenant plusieurs informations :

— Paramètres : il s'agit de l'ensemble de tous les paramètres du système étudié. Ces paramètres sont utilisés pour configurer chacun des noeuds présents dans le système. Pour chaque paramètre, l'utilisateur doit fournir une ou plusieurs valeurs qui seront utilisées dans l'exécution du banc de simulations.

— Vecteurs de paramètres : un vecteur de paramètres contient la configuration pour effectuer une exécution. L'utilisateur peut demander à répertorier tous les vecteurs de paramètres, ou fournir une fonction pour les calculer.

— Itérations : il s'agit du nombre de répétitions de chaque cycle expérimental. Cet élément est nécessaire dans le cas d'utilisation de composants non déterministes tels que les composants dépendant du bruit sur le balayage laser.

— Conditions d'arrêt : il indique quand arrêter chaque cycle expérimental. Cela peut être une condition pour évaluer si l'expérience est terminée avec succès, ou un délai pour évaluer les exécutions échouées afin d'éviter une boucle infinie.

— Mesures : ce sont les données à collecter pour chaque cycle expérimental. L'utilisateur doit indiquer la fréquence de chaque mesure ou une condition de déclenchement pour collecter des données lors de la survenance d'un événement.

— Métriques : ce sont les formules qui calculent les résultats de référence à partir des données collectées.

4.2.4 Structure d'un robot

Chaque contrôleur de robots se matérialise comme un ensemble de nœuds ROS. Dans un souci de réalisme, nous utilisons le même contrôleur que celui fonctionnant sur des robots réels. L'architecture logicielle comprend principalement quatre packages ROS (illustrée à la Figure 4.3) :

— gmapping [1] fait partie des packages ROS officiels, qui effectuent un algorithme de localisation et de cartographie simultanées (SLAM). Il fournit un nœud ROS qui prend les données de balayage laser, construit une carte et estime la position courante du robot. Profitant de ce package, nous avons extrait les informations de position de robot [2], puis les introduisons dans le package explore.

— explore [3] a été initialement développé par Bosch. Il met en œuvre l'exploration d'un robot unique. Nous l'avons étendu pour prendre en charge un système multirobots [4]. Il publie la carte explorée, et quelques métriques (le coût estimé et le gain d'informations attendu), et reçoit la carte d'autres robots via le package de coordination.

— La coordination a été développée en tant qu'interface au banc de simulations. Les utilisateurs peuvent mettre en œuvre leurs propres stratégies de coordination telles que la fusion de cartes [5],

1. http://wiki.ros.org/gmapping
2. http://wiki.ros.org/pose_publisher
3. http://wiki.ros.org/explore
4. http://wiki.ros.org/explore_multirobot
5. http://wiki.ros.org/map_merging

Parameters	**Robot**	Robot Model: Pioneer 3-DX
		Computing Capabilities: 2 CPUs, 2 GB RAM
		Maximum speed: 1.2 m/s, 5.24 rad/s
		Laser rangefinder: SICK LMS500
	Fleet	Number of robots: $[1, 30]$
		Homogeneity: homogeneous (Pioneer 3-DX)
		Robot initial positions: top left to bottom left corner, every 2 m
		Communication network: wired Gigabit Ethernet
		Communication range: 200 m
	Environment	Terrains: $loop, cross, zigzag, maze$
		Terrain size: 80 m $\times$ 80 m
		Obstacle height: 2 m
		Road width: 8 m
Parameter vectors	All possible combinations	
Iterations	5	
Stop conditions	99% of explored terrain	
	2000 s of elapsed time	
Measurements	Explored area per simulation step	
	% of CPU used by each robot every 5 s	
	% of RAM used by each robot every 5 s	
	Kbytes received by each robot every 5 s	
	Kbytes sent by each robot every 5 s	
Metrics	Exploration time	
	Exploration cost	
	Exploration efficiency	
	Map quality	

FIGURE 4.2 – Exemple de fichier de configuration du banc de simulation.

la planification des tâches et la planification des mouvements. Dans notre cas, il diffuse la carte fusionnée au package explore et aux autres robots du système.

— move_base [6] fait également partie des packages ROS officiels. Il implémente l'algorithme de Dijkstra pour la planification globale des chemins.

4.2.5 Discussion

D'une manière générale, le banc de simulation nous a permis de simuler et analyser certains aspects de l'exploration par des flottes multirobots. Notamment, nous avons obtenu des mesures de l'efficacité de l'exploration en fonction du terrain à explorer, du nombre de robots et de la stratégie d'exploration choisie (voir l'article [3] pour les détails).

Ces résultats sont encourageants, mais lèvent les discussions ci-dessous.

6. http://wiki.ros.org/move_base

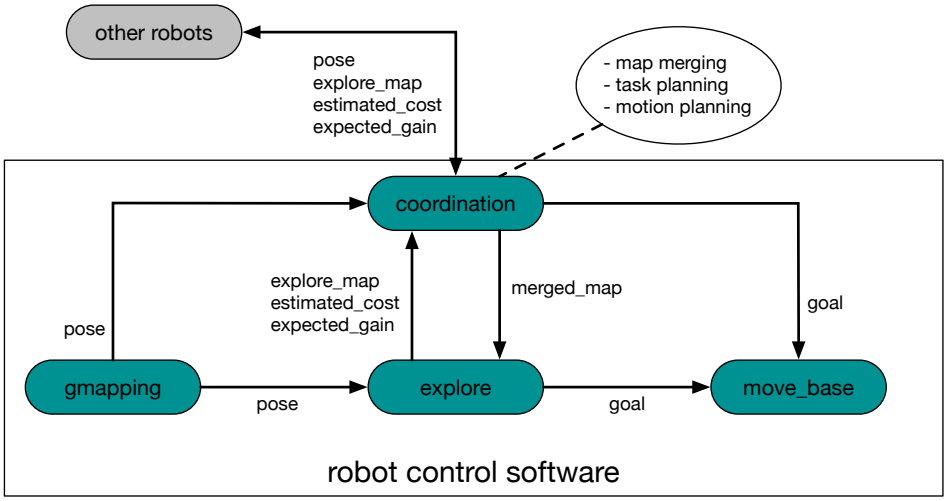

FIGURE 4.3 – Architecture logicielle d'un robot

Fiabilité du réseau : Dans les systèmes multirobots réels, la coordination repose souvent sur une communication directe via des interfaces réseau sans fil. Les robots forment un réseau mobile ad hoc sans fil qui peut être déconnecté lorsqu'ils s'éloignent les uns des autres pour effectuer leurs tâches [Bouraqadi 2009]. Ce manque de fiabilité peut avoir des répercussions importantes sur le système de systèmes.

La réalisation d'une simulation réaliste nécessite de prendre en compte cette caractéristique du réseau. C'est pourquoi dans notre banc de simulation, les robots ne peuvent communiquer que s'ils se trouvent à une certaine distance. Cependant, la portée n'est qu'une facette des réseaux sans fil. Il existe d'autres facteurs qui peuvent influer sur la fiabilité des communications, tels que les interférences radio, les retards de communication et l'atténuation du signal causée par les obstacles. Une simulation plus réaliste doit intégrer des modèles de propagation radio et des protocoles de réseau comme cela est fait dans RoboNetSim [Kudelski 2013]. Cependant, une conséquence directe de l'introduction de ces modèles est la nécessité d'augmenter la puissance de calcul du simulateur.

Bande passante réseau : Dans notre banc de simulation, le réseau est utilisé pour la communication intrarobotique entre le contrôleur de robots et le corps du robot, ainsi que pour la communication entre différents robots. Selon la topologie du réseau et sa bande passante, il peut y avoir des limitations sur le nombre maximum de robots. Dans un cadre réel avec un schéma de travail décentralisé, différents groupes de robots éloignés les uns des autres au-delà de la portée du réseau sans fil peuvent former différents clusters faisant émerger une communication en parallèle.

Dans le cadre de notre expérimentation, nous avons identifié des limitations réseau. Après étude de ces limites, nous avons constaté que les robots partagent la carte complète à chaque envoi. En effet, dans le développement de nos outils et la simulation du système multirobot, nous n'avions pas considéré les spécifications réseau. Il conviendrait de mettre en place une stratégie de communication plus appropriée où seule une carte partielle découverte est envoyée régulièrement, et une carte complète serait envoyée à une fréquence plus faible.

Matériel dans la boucle : Certains travaux établissent une séparation stricte entre les simulations et les tests sur les systèmes multirobots [Michael 2008], tandis que d'autres chercheurs proposent d'introduire du matériel dans la boucle [Jinhui 2009]. Ce dernier a effectué des expériences hybrides, dans lesquelles un sous-ensemble de pièces de robots simulées a été remplacé par du matériel réel.

Dans notre proposition, nous nous appuyons sur le middleware ROS. Les contrôleurs de robots, les capteurs et les actionneurs sont tous des nœuds ROS, qui communiquent via le middleware ROS. Les capteurs et actionneurs simulés peuvent être remplacés par des capteurs et actionneurs matériels. Cela peut être fait étape par étape et intégré dans une méthodologie de tests de robots telle que celle présentée dans [34] et dans la Section 4.3.

Les nœuds ROS peuvent également être déployés sur des robots réels, ce qui permet de basculer de la simulation au système réel et d'utiliser la puissance informatique embarquée des robots.

4.2.6 Synthèse

Ce travail de recherche fournit une plateforme de simulation multirobot en utilisant le middleware ROS. Cette plateforme permet l'explicitation des paramètres de simulation et par conséquent assure la reproductibilité de la simulation.

En utilisant le middleware ROS, nous garantissons une compatibilité entre les artefacts simulés et les robots réels. Ainsi, nous pourrions aisément, dans le cadre d'un autre cas d'étude intégrer le "hardware in the loop", stratégie pour laquelle on introduit des éléments réels dans le cadre de la simulation.

Les simulations réalisées nous ont permis de valider des algorithmes d'exploration, disponibles dans l'article [3], mais également de remonter des problèmes d'architecture que nous n'aurions pas identifiés avec des simulations moins grandes.

Nous avons montré que la simulation est importante dans de nombreux cas, mais n'est pas suffisante et nécessite de tester les CPS dans des environnements réels et incertains. Dans la section suivante (Section 4.3), j'aborde cette problématique en présentant une méthodologie de tests dédiés aux robots.

4.3 Méthodologie de tests pour les CPS : application aux robots mobiles

4.3.1 Objectifs de recherche

Pour définir les exigences de tests des CPS, nous nous sommes inspirés des tests de logiciels. Chung et al. [Chung 2007] proposaient déjà de construire des tests adaptés du génie logiciel. Le test logiciel est un processus bien connu qui présente de multiples avantages : il valide le comportement d'un morceau de code, il permet aux développeurs de refactoriser, de maintenir et d'améliorer le code source soutenu par des tests en tant que processus de validation. Les tests logiciels permettent également de valider les modifications de l'environnement. Appliqués au domaine des CPS, nous avons identifié et adapté trois exigences :

Sécurité : la sécurité est un point clé lors de l'exécution des tests. En effet, comme les tests sont exécutés sur des CPS potentiellement défectueux, il est important de s'assurer qu'ils ne blessent pas ou n'endommagent pas l'environnement. Cela signifie que les tests doivent être exécutés dans un ordre précis pour garantir d'abord la validité des fonctions de base avant celles de haut niveau qui sont plus complexes et moins prévisibles. Les tests doivent également valider le fait que les CPS fonctionneront en toute sécurité.

Réutilisation : les tests doivent être réutilisables pour des CPS hétérogènes construits à partir de composants différents, mais qui sont supposés présenter le même comportement. Par exemple, différents CPS peuvent être équipés de capteurs ou d'actionneurs différents, mais dont les données envoyées sont équivalentes. Pourtant, leur comportement doit être le même. Cela signifie que les tests doivent se concentrer davantage sur les comportements des CPS plutôt que sur leur structure interne.

Répétabilité : les tests doivent permettre aux testeurs de répéter les mêmes scénarios et faire en sorte que les CPS soient confrontés exactement aux mêmes situations. À cet égard, la dynamicité de l'environnement dans lequel les CPS sont installés est difficilement modélisable. Néanmoins, la répétabilité des tests est nécessaire pour pouvoir valider le comportement de chaque CPS. Une façon de gérer la répétabilité est l'automatisation. Les tests doivent être automatisés ou au moins semi automatisés pour garantir que les CPS seront confrontés exactement à la même situation à chaque test.

La principale contribution de cette section est une méthodologie de tests pour répondre à ces exigences. Cette méthodologie est le résultat de notre expérience et des leçons que nous avons apprises lors du développement d'une application de robotique de service impliquant des robots mobiles autonomes dans le cadre du projet Roboshop [7].

4.3.2 Framework de tests

Pour réaliser notre framework de test, nous avons identifié trois dimensions de tests : l'activité du CPS, la nature de l'environnement et les connaissances des testeurs sur l'environnement.

Activité du CPS : Est-ce que le CPS réalise de la détection uniquement ou active des actionneurs ? Soit un CPS peut détecter son environnement, soit il peut détecter l'environnement tout en

7. http://car.imt-lille-douai.fr/category/project/roboshop/

agissant. Étant donné que l'action repose sur la rétroaction obtenue par les capteurs, les tests impliquant des actionneurs nécessitent des capteurs entièrement fonctionnels.

Dans un environnement donné, il faut d'abord effectuer des tests de capteurs pour s'assurer qu'ils fonctionnent correctement. Ensuite, une fois que la détection peut être considérée comme fiable, les tests impliquant le mouvement et les actionneurs peuvent être effectués.

Nature de l'environnement : Est-ce que le CPS évolue dans un environnement dynamique ? L'environnement peut être statique ou dynamique. Un environnement statique ne change pas pendant que le CPS accomplit sa mission. Un environnement dynamique évolue avec le temps. En règle générale, il contient des entités en mouvement. Les modifications des données détectées peuvent être provoquées par le CPS lui-même ou par d'autres entités.

La sécurité nécessite que les tests en environnement statique soient effectués avant ceux liés aux environnements dynamiques. Le CPS doit d'abord pouvoir fonctionner et se déplacer dans un monde stable, avant de pouvoir faire face à des situations avec des entités mobiles.

Connaissance de l'environnement : Est-ce que le testeur est capable de modéliser l'environnement du robot ? Pour tester un robot, le testeur doit connaître l'environnement (par exemple les dimensions d'une arène de tests ainsi que les emplacements et les tailles des obstacles) avec des niveaux de précision. Sur la base de ces connaissances, les testeurs peuvent prédire des valeurs de données détectées qui reflètent un comportement correct du robot. Pourtant, pour tester un robot de manière plus approfondie, les testeurs doivent expérimenter différents environnements, y compris ceux pour lesquels ils ont peu de connaissances.

Pour aider à comprendre le comportement du robot, les tests doivent d'abord être effectués dans un environnement totalement contrôlé. Il est plus facile de détecter les comportements défectueux et de déclencher des interruptions de sécurité. Une fois qu'un robot réussit ses tests, les testeurs peuvent procéder à des expériences dans des environnements moins contrôlés.

Ces dimensions sont combinées et ordonnées selon leur niveau de sécurité. Elles sont donc regroupées en huit ensembles différents. Nous avons ordonné ces ensembles de tests pour aboutir à une hiérarchie de cinq niveaux de test. Pour la première dimension, nous considérons que la détection seule est plus sûre que lorsque l'action est impliquée. Pour la deuxième dimension, nous considérons que se déplacer dans un environnement connu est plus sûr que se déplacer dans un environnement inconnu. Pour la troisième dimension, nous considérons qu'un environnement statique est plus sûr qu'un environnement dynamique.

La Figure 4.4 montre l'ordre des tests organisé en cinq niveaux de sécurité. Chaque niveau est une condition préalable au niveau suivant. Cela signifie qu'un test ne doit pas être effectué sur un CPS tant que tous les tests des niveaux précédents ne sont pas réussis. Ainsi, notre méthodologie guide les développeurs pour définir des tests reproductibles et réutilisables couvrant différentes facettes d'un CPS. Le problème d'automatisation est également traité lors de l'étape de définition du test. Enfin, la méthodologie décrit le processus de réalisation des tests tout en maximisant la sécurité des opérateurs, du robot et des autres équipements.

Deux combinaisons ne sont pas visibles dans la méthodologie, ce sont les ensembles qui vérifient les capacités de détection dans un environnement inconnu qui peut être statique ou dynamique. Dans ces deux ensembles, l'environnement n'est pas contrôlé par les développeurs, il leur est donc difficile de définir des tests significatifs, car ils ne peuvent pas prédire des valeurs valides pour les données détectées.

Chacun des ensembles est expliqué ci-dessous :

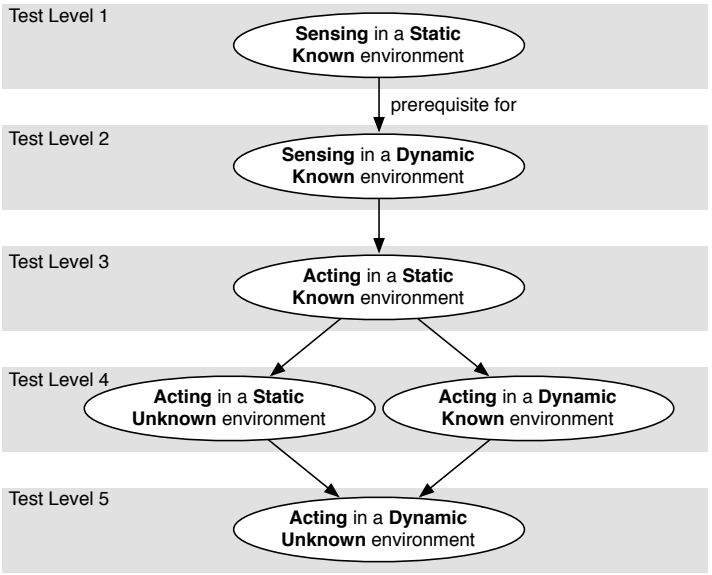

FIGURE 4.4 – Les cinq niveaux de tests de la méthodologie

1. Tester les capacités de détection du CPS dans un environnement statique connu (Niveau 1). Cette suite de tests garantit que les données produites par des capteurs correspondent au niveau de précision attendue.
2. Tester les capacités de détection des CPS dans un environnement dynamique connu (Niveau 2). Un tel environnement est contrôlé par les développeurs, même s'il est dynamique. Par exemple, une pièce avec des entités en mouvement suivant un chemin connu ou avec la lumière changeant selon certains motifs connus. Comparé à la détection dans un environnement statique, un environnement dynamique introduit le facteur temps. Dans un environnement dynamique, les tests portent davantage sur la fréquence des données collectées que sur leur précision. Par exemple, les tests de ce niveau peuvent évaluer si un CPS peut collecter des données assez rapidement pour détecter des obstacles en mouvement.
3. Tester l'action du CPS dans un environnement statique connu (Niveau 3). Ici, l'objectif est de s'assurer que les actionneurs du CPS fonctionnent correctement et que le CPS effectue des actions de base. En outre, les testeurs doivent s'assurer qu'ils ont des solutions pour arrêter le CPS en cas d'urgence (par exemple un bouton d'arrêt d'urgence sur un robot mobile).
4. Le niveau 4 contient deux types de tests complémentaires liés à l'action du CPS dans des environnements statiques inconnus et dynamiques connus. Les tests d'action du CPS dans un environnement statique inconnu ont pour objectif d'accroître la confiance dans le CPS testé, en le testant face à de nombreuses situations différentes. Les tests d'action du CPS dans un environnement dynamique connu permettent de prédire les données détectées. Étant donné que l'environnement est connu, les développeurs peuvent définir des tests en lien avec des événements et vérifier que le CPS effectue les bonnes actions.
5. Tester l'action du CPS dans un environnement dynamique inconnu (Niveau 5). Ce niveau représente les tests finaux. Il vise à confronter les CPS à autant de situations différentes que possible. Les tests sont considérés comme réussis si le CPS peut faire face assez rapidement aux événements rencontrés. Un exemple d'un tel test est la recherche d'un robot mobile pour une personne

basée sur la reconnaissance faciale, dans un endroit où de nombreuses personnes passent (par exemple une gare).

Une fois un niveau de test choisi, le développeur doit rédiger chacun des tests. Le framework impose différents éléments à donner pour la rédaction des tests :

— Niveau de test : Cette information permet d'identifier les tests prérequis qui doivent être exécutés avant le test en cours. Cela maximise la sécurité mais facilite également l'identification des problèmes.

— Conditions initiales : Elles donnent l'état initial de l'environnement (par exemple, intensité lumineuse, positions d'obstacles) et l'état du robot (par exemple sa position dans l'environnement). Ces conditions assurent la répétabilité du test.

— Résultat attendu : Chaque test doit évaluer la réception de données ou certains comportements d'un robot. Lorsqu'il s'agit de données, le résultat attendu doit être exprimé sous la forme d'un intervalle couvrant la plage de valeurs valides. Dans certains tests, le résultat attendu peut être plus qualitatif. Un exemple est "le robot doit suivre des trajectoires fluides". Dans ce cas, le contrôle et la validation par l'humain sont envisagés.

— Tâche à effectuer : Cet élément définit la tâche à effectuer par le robot avant que le résultat soit comparé aux exigences. La tâche peut être simple telle que de lire une valeur de capteur, ou une tâche plus complexe telle que chercher un objet dans une pièce.

— Évaluation : Elle est effectuée une fois que le robot a terminé la tâche à accomplir. L'évaluation se réfère à la comparaison du résultat réel avec celui attendu. Idéalement, cette comparaison est effectuée automatiquement. Cependant, une intervention humaine peut être nécessaire dans des situations telles que la mesure de la distance parcourue. L'évaluation humaine est même particulièrement difficile à remplacer pour les tests qualitatifs, qui sont généralement ceux effectués dans des environnements inconnus des niveaux 4 et 5.

— Nombre d'itérations : Les tests doivent généralement être exécutés plusieurs fois. Cela est vrai lors de l'évaluation lorsqu'une valeur se situe dans une certaine plage, ou que le comportement du CPS est non prédictif. C'est également le cas pour les tests qualitatifs.

4.3.3 Expérimentation

Pour l'expérimentation, nous étions en possession de deux robots identiques, pesant 30 kg chacun. Ce sont des robots mobiles à 2 roues motrices, dont la vitesse maximale est de 1 m/s. Ils ont plusieurs capteurs : 1 laser, 16 capteurs infrarouges, 9 capteurs ultrason, 2 webcams et 2 capteurs odométriques. Nous avons développé un nœud ROS pour communiquer avec ces robots et profitez de l'écosystème ROS. Nous avons développé ce nœud en utilisant PhaROS, un client ROS que nous avons développé à l'école des Mines de Douai.

Nous avons réalisé quelques tests pour les deux robots en utilisant le framework de tests.

Tests de détection dans un environnement statique connu (Niveau 1) : Nous avons réalisé des tests pour le capteur laser, un Sick S300 [8]. Il a un angle de balayage de 270 degrés. La distance pouvant être mesurée est de 30 mètres. Nous plaçons le robot dans un environnement statique : une boîte d'un mètre carré, illustré par la Figure 4.5.

8. http://www.sick.com/

Nous effectuons cinq tests à l'intérieur de la boîte, un avec le robot situé au centre de la boîte et les autres avec le robot à chaque coin. Ces tests sont semi-automatiques, car une intervention humaine est requise pour placer le robot aux positions appropriées dans la boîte. La mesure de distance et la comparaison avec les valeurs attendues sont effectuées automatiquement par un logiciel de gestion de tests.

Les résultats ont montré que les distances pour les angles 0 à 1 et pour les angles 269 à 270 sont incorrectes (moins de 18 centimètres) pour les deux robots. Après analyse, nous avons découvert que les lasers étaient mal montés. Le corps du robot est trop proche du laser.

FIGURE 4.5 – Testing the Robot Sensors in a Static Known Environment

Tests de détection dans un environnement dynamique connu (Niveau 2) : Ce test évalue la vitesse du laser embarqué. Notre objectif est de s'assurer que la fréquence de balayage est suffisamment élevée pour détecter les obstacles situés à moins de 20 cm du robot.

La vitesse de marche humaine moyenne est d'environ 1,49 m/s (5,0 km/h). La vitesse maximale de nos robots est de 1 m/s. Supposons que nous ayons un robot à vitesse maximale et un humain marchant se rapprochant. S'ils sont à 20 cm l'un de l'autre, la collision se produira dans environ 80 ms. Donc, pour assurer la sécurité, nous nous attendons à ce que la fréquence de balayage laser soit d'au moins 12,5 Hz.

Les données collectées par le laser sont publiées dans un topic ROS nommé "/scan". Le test mesure la fréquence des publications. Ainsi, notre test prend en compte les retards introduits par le traitement des données brutes du laser et la transformation dans le format requis pour la publication dans un topic.

Le test révèle que la fréquence des balayages est d'environ 15 Hz. Nous en concluons que la fréquence des balayages laser est suffisamment élevée pour permettre au robot de détecter les collisions potentielles.

Tests d'activités dans un environnement statique connu (Niveau 3) : Nous avons effectué des tests

pour vérifier si le robot peut se déplacer avec précision à différentes vitesses (0,25 m / s, 0,5 ms, 1 m / s), en se basant uniquement sur l'odométrie, dans un environnement statique connu.

— Se déplacer dans un couloir : l'environnement que nous avons utilisé dans cette série de tests est une section d'un couloir dont les deux extrémités ont été fermées. Nous effectuons différents tests où le robot se déplace à différentes vitesses sur une distance de 4 mètres en ligne droite. Une marge d'erreur de 3% a été considérée comme acceptable. Nous mesurons manuellement la distance parcourue. Et nous nous appuyons sur les données du laser pour vérifier automatiquement que les distances aux deux murs sont dans la plage de données acceptable. Nous avons répété l'expérience 10 fois pour chaque robot à différentes vitesses.

  Nous avons observé qu'un des deux robots avance avec une erreur moyenne de 1,87%. Pour le deuxième robot, le test a échoué 5 fois lorsque le robot s'est approché trop près du mur. Nous avons identifié que le problème venait d'une roue partiellement bloquée par un ressort cassé. Parce que ce robot n'a pas réussi ce test, nous n'avons pas effectué de tests de niveau supérieur sur celui-ci.

— Suivre un chemin en forme de S : Nous avons mis en place un test où le robot devait suivre un chemin en forme de S. Nous avons marqué au sol 4 points des coordonnées du chemin : (0,0), (1, -0,5), (3,0,5) et (4,0). Cette trajectoire correspond à une rotation cumulée de 180 degrés. Le robot devait passer par chacun de ces points. Nous avons considéré 3% comme une erreur de marge acceptable pour la distance parcourue, 10% pour la rotation. Le test nécessite une intervention humaine pour mettre le robot au point (0,0). Une intervention humaine est également nécessaire pour vérifier que le robot passe par les points.

  La mesure de la position finale a révélé que le robot répond à nos attentes. L'erreur de marge de distance parcourue est d'environ 1,9% et l'erreur de marge de rotation est d'environ 9,2%.

Tests d'activités dans un environnement dynamique connu (Niveau 4) : Nous illustrons ce niveau avec l'exemple d'arrêt d'urgence au cas où un obstacle se rapprocherait trop d'un robot en mouvement. L'expérience consiste à faire avancer un robot en suivant une ligne droite dans un couloir. Le couloir est initialement vide, mais lorsque le robot atteint une position particulière marquée au sol, un opérateur place un panneau en carton à 30 cm du robot. Le robot devrait s'arrêter immédiatement sans toucher la planche.

Nous avons répété l'expérience 10 fois. Il s'avère que le robot réussit ce test pour différentes vitesses allant de 0,25 m/s à 1 m/s.

Les tests effectués nous permettent de découvrir des problèmes sur les robots qui ne sont pas faciles à mettre en évidence sans tests.

4.3.4 Synthèse

Les tests des CPS doivent être effectués de manière à garantir la sécurité. Par ailleurs, les tests doivent être reproductibles et aussi automatiques que possible. La méthodologie que nous proposons répond à cet objectif.

La méthodologie s'appuie sur trois dimensions. La première dimension concerne les opérations effectuées par le CPS testé : détecter uniquement ou détecter et agir. La deuxième dimension concerne

les connaissances des testeurs sur l'environnement où des tests sont effectués : ils peuvent avoir une connaissance précise de l'environnement par rapport à une faible connaissance de celui-ci. La dernière dimension concerne l'environnement : il peut être statique ou dynamique. Ces trois dimensions nous permettent de regrouper les tests dans différents ensembles. Sur la base d'une analyse de ces ensembles, nous avons identifié cinq niveaux de test ordonnés de manière à maximiser la sécurité.

Nous avons expérimenté le framework avec deux robots similaires, en utilisant le middleware ROS [Quigley 2009].

Notre méthodologie nous a permis de développer une suite de tests semi-automatiques reproductibles. Cette suite nous aide à identifier les différents défauts des CPS et à les corriger.

4.4 Méthodologie de co-conception logiciel / FPGA

4.4.1 Objectifs de recherche

Dans les deux sections précédentes, nous avons mis en valeur la nécessité de simuler et de tester les CPS. Dans le cadre de cette section, j'aborde l'intérêt de la co-conception logiciel / matériel dans les environnements CPS, illustré dans la thèse de Xuan Sang Le. Dans ses travaux, il a mis en place un framework de co-conception pour FPGA dans un environnement de développement orienté objet. Notamment, il a mis en oeuvre un framework permettant d'ajouter un point d'arrêt sur un FPGA permettant de déboguer le système.L'objectif était d'exploiter les avantages des deux univers.

Il existe de nombreux avantages à utiliser des processeurs FPGA dans des environnements nécessitant de la puissance de calcul, notamment en robotique. Cependant, le problème d'interface entre FPGA et l'environnement informatique reste problématique. Surtout, lorsque l'on veut intégrer des FPGA dans un système de systèmes existant.

La conception de FPGA est une tâche compliquée. Le processus de conception nécessite une connaissance spécifique qui reste un défi pour les développeurs. En particulier, lorsque des FPGA sont utilisés avec des systèmes logiciels avancés, le problème de la co-conception devient un réel défi. Lorsque l'on examine les méthodologies de conception existantes pour ces systèmes, il est clair que les méthodologies de conception logicielle, telles que la conception orientée objet, sont en avance en termes de productivité. La conception matérielle reste un processus long et fastidieux.

De plus, le flux de conception est particulièrement rigide avec une séparation entre l'environnement de conception, composé de logiciels spécifiques, l'environnement de génération de circuit FPGA, et le FPGA lui-même séparé géographiquement.

Développer un environnement dédié pour prendre en charge le développement et le déploiement dynamiques de ces systèmes permettrait d'assurer :

— L'intégration et l'interopérabilité : l'environnement devrait favoriser l'intégration de la conception logicielle et matérielle dans un processus commun. Il permettrait également l'intégration de chaque nœud au système d'information au moment de l'exploitation. Un protocole de communication est nécessaire pour l'interfaçage logiciel / matériel ainsi que pour le contrôle des constituants FPGA.

— La reconfigurabilité et la maintenabilité : un système de système peut contenir des centaines de constituants pour certains sous la forme de FPGA. Le déploiement manuel, la reconfiguration ou la maintenance de chaque nœud est une tâche fastidieuse et avec peu de valeur ajoutée. Un mécanisme automatique doit être envisagé pour le déploiement et la configuration à distance de ces constituants.

Le contexte converge ici vers l'utilisation de dispositifs hybrides FPGA / processeur dans un système IoT. Il en résulte une proposition d'un flux de conception dédié, de middleware et d'outils pour simplifier le processus de développement SW / HW. Le but est de fournir un environnement qui permet aux développeurs de bénéficier du développement orienté objet pour développer facilement des applications dédiées aux FPGA, tout en ayant toujours la possibilité d'exploiter les fonctionnalités matérielles des FPGA. Par conséquent, notre contribution aborde le problème ciblé selon trois axes principaux :

Proposer le métamodèle OoRC. Le métamodèle OoRC est conçu avec les principes et le modèle de

conception orienté objet. Bien qu'il prenne pleinement en charge la modélisation des circuits FPGA en utilisant un DSL dédié, son objectif principal est d'améliorer l'expérience de conception du FPGA. On peut utiliser le métamodèle pour fournir une solution générique (c'est-à-dire indépendante de l'implémentation) à un problème spécifique afin qu'il puisse être adaptable dans différents contextes d'application.

Combler l'écart Logiciel / Matériel. OoRCBridge est une plateforme construite au-dessus du métamodèle OoRC. Il permet la co-conception Logiciel / Matériel. Il intègre des périphériques FPGA avec des environnements orientés objet. Le point clé d'OoRCBridge est l'intégration indépendante via la standardisation d'interface. Nous avons développé une interface de communication et un protocole standardisé. La génération automatique de code fournit une API pour l'accès matériel depuis un environnement logiciel de haut niveau et permet de réduire l'écart logiciel / matériel.

Développement centralisé et déploiement à distance automatique. CaRDIN fournit un environnement dédié pour développer des applications décentralisées. Il utilise OoRCBridge pour faciliter et standardiser l'intégration logiciel / matériel sur chaque nœud. De plus, les nœuds peuvent être reconfigurés à distance sans avoir besoin de les enlever du site de déploiement. Cela permet une meilleure évolutivité du système.

Pour démontrer et valider la plateforme, une expérience sur un capteur qui effectue une analyse d'image a été effectuée, comme preuve de concept. Ce système se connecte à une caméra et exécute un algorithme de détection d'objet sur le FPGA. Il communique la position de l'objet.

Dans cette section, je vais me concentrer principalement sur la partie de nos travaux dédiés à l'outillage permettant de faire évoluer les FPGA distribués dans un système de systèmes.

4.4.2 Le Métamodèle OoRC

Pour décrire les circuits FPGA, nous avons défini un métamodèle qui s'appuie fortement sur le langage de description du matériel VHDL. Le langage est livré avec de nombreuses extensions et peut être utilisé pour réaliser des descriptions matérielles ou pour faire de la simulation. Notre objectif étant de modéliser des circuits FPGA pouvant être exécutés sur du matériel, le métamodèle est basé sur un sous-ensemble de VHDL dédié aux systèmes synthétisables : le standard de synthèse RTL IEEE 1076.66 (VHDL-87) [IEEE 2004].

Les structures du standard sont appelées métadescriptions dans notre métamodèle. Une métadescription contient des informations structurelles et comportementales sur un élément d'un circuit (données, fonctionnement, contrôle ou autre sous-circuit). Ainsi, pour chaque élément individuel, nous savons comment il est organisé, comment il se connecte à d'autres éléments et comment il est exécuté lorsqu'il est activé. Un diagramme de classe simplifié du métamodèle se trouve en Figure 4.6.

Notre métamodèle permet de construire des modèles directement exécutables dans lesquels le circuit est considéré comme un objet dans notre environnement de programmation dont les états internes peuvent être modifiés (entrée, sortie, valeur du signal, etc.) et peuvent être inspectés lors d'une phase de débogage. Il peut donc être manipulé et déboguer comme un objet normal. Le circuit peut être vérifié de manière transparente avec ou sans matériel. Enfin, pour prendre en charge la synthèse sur le matériel réel, nous avons développé l'export du métamodèle vers des descriptions VHDL pour une synthèse sur FPGA. OoRC dispose d'une chaîne d'outils qui automatise la synthèse des circuits à l'aide des outils du fournisseur.

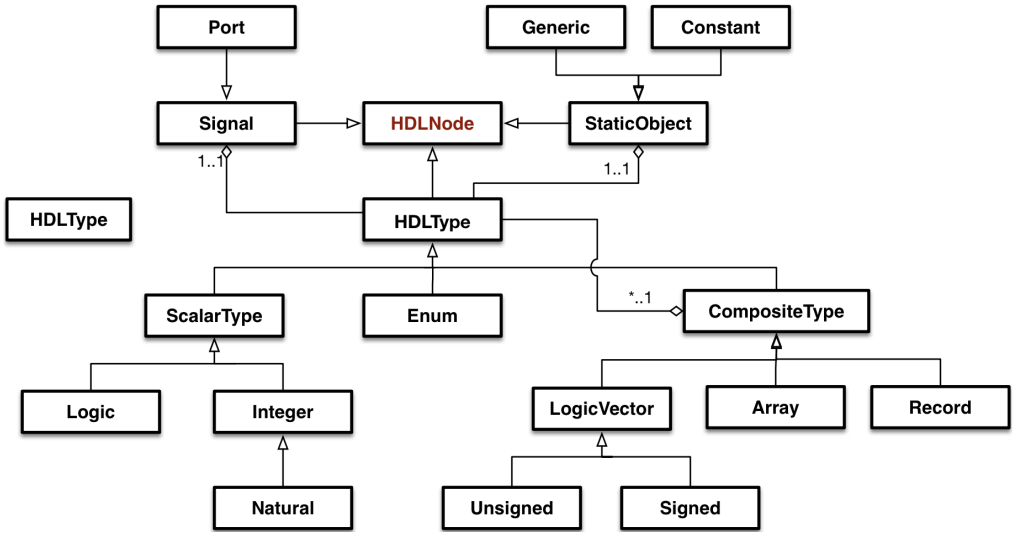

FIGURE 4.6 – Métamodèle OoRC

4.4.3 DSL simplifiée

OoRCScript est un DSL intégré dans l'environnement de développement et il est basé sur le métamodèle OoRC pour la description des circuits. Il est extensible, sans compilateur et hérite de toutes les caractéristiques d'un langage interprété ainsi que de l'environnement dynamique de Pharo [9]. Tout en étant capable d'exprimer des concepts matériels de bas niveau à l'aide de l'API. Par conséquent, il connaît la plupart des principes de conception orientés objet.

4.4.4 Réutilisation du code VHDL hérité

Un problème courant lors de la modélisation de systèmes matériels est de savoir comment réutiliser les conceptions VHDL existantes. OoRC fournit un importeur de VHDL hérité. C'est-à-dire que le métamodèle est capable d'instancier et de présenter des circuits VHDL qui peuvent ensuite être réutilisés. Pour prendre en charge la réutilisation de l'héritage VHDL, le système s'appuie sur un analyseur VHDL dédié pour importer et transformer un modèle HDL en modèle OoRC.

Une telle réutilisation peut enrichir la conception des modèles tout en réduisant les coûts de production en bénéficiant d'un ensemble existant de code VHDL tiers. Notre métamodèle prend en charge la réutilisation de presque toutes les conceptions VHDL conformes à la norme IEEE 1076.66 [IEEE 2004]. Le métamodèle est capable d'importer un fichier VHDL et d'en reconstruire un modèle de circuit sous forme de boîte noire. Son architecture interne est "inconnue" et nous ne pouvons donc pas la simuler dans Pharo.

Cependant, cela ouvre la voie à la co-simulation où la boîte noire fonctionne sur le vrai matériel et le reste est simulé. Nous prévoyons d'étendre notre solution pour prendre en charge cette fonctionnalité dans un futur travail. L'exécution matérielle des modèles à boîte noire est plus facile, car le code VHDL existe déjà et il est prêt pour la synthèse. Aucune génération de code n'est nécessaire. Le

9. https://pharo.org/

métamodèle n'a besoin que de générer l'interface de communication appropriée.

4.4.5 Environnement de co-conception

L'environnement de co-conception OoRCBridge est un ensemble d'outils et middleware dédiés pour l'intégration de FPGA à l'environnement de développement orienté objet Pharo [10]. Avec l'aide du métamodèle OoRC, l'approche repose sur un mécanisme de génération automatique de code.

Nous avons intégré la possibilité de débogage sur le matériel. Pour activer le débogage, nous avons intégré un concept de point d'arrêt matériel dynamique, présenté dans [49]. L'idée est que, lors de la génération du circuit, si la fonction de débogage est activée, un sous-circuit de débogage spécifique et une logique de commande de point d'arrêt sont injectés automatiquement. La Figure 4.7 illustre la conception et le sous-circuit généré.

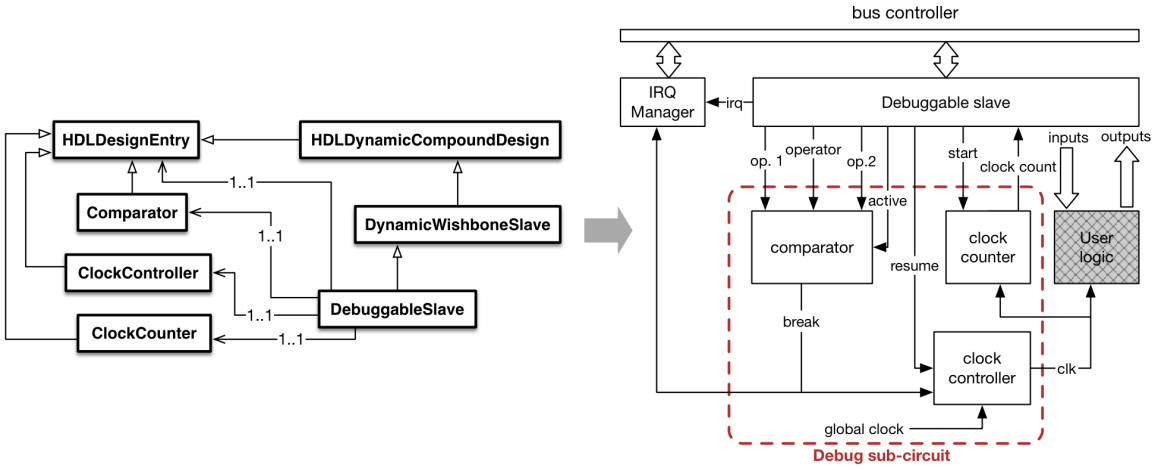

FIGURE 4.7 – Exemple d'un circuit généré

Le mécanisme permet d'arrêter le flux d'exécution en coupant une horloge spécifiquement intégrée au circuit. C'est-à-dire qu'au lieu d'alimenter le circuit avec l'horloge globale, le métamodèle génère une horloge reliée à l'horloge globale en entrée et émet un signal d'horloge contrôlable. Ce contrôleur peut désactiver l'horloge de sortie lorsqu'un point d'arrêt est défini ou satisfait à sa condition.

Lorsque l'horloge est coupée, l'exécution du circuit est ainsi interrompue et son état en cours est maintenu. Cet état peut facilement être lu dans l'environnement de développement et relié au modèle de circuit via l'interface de communication. Ceci nous permet d'inspecter et de modifier les valeurs transitant dans le circuit, comme sur un processus de débogage d'une application locale.

Dans la Figure 4.7, DebuggableSlave est utilisée. Lorsque le point d'arrêt est activé et que la condition sur le comparateur devient vraie, le comparateur déclenche le signal de coupure envoyé à l'horloge. Ce composant désactive l'horloge à sa sortie. La coupure de l'horloge entraîne l'arrêt de l'exécution du circuit cible tout en maintenant son état actuel. En plus du contrôleur d'horloge, un troisième composant, le compteur d'horloge, est ajouté pour mesurer le temps d'exécution du circuit

10. https://pharo.org/

cible en cycles d'horloge. Étant donné que le compteur utilise la même horloge que le circuit, il est également arrêté lorsque l'horloge est désactivée. De cette façon, le développeur peut inspecter exactement le temps d'exécution du circuit au moment de l'arrêt.

L'environnement peut reprendre l'exécution interrompue à tout moment en modifiant le registre de reprise qui se connecte au contrôleur d'horloge. Cette action permet au contrôleur d'horloge d'activer l'horloge de sortie et donc de relancer le circuit cible et le compteur d'horloge. Le flux d'exécution peut alors continuer.

4.4.6 Expérimentation

Nous avons utilisé les mêmes robots autonomes que dans la section précédente. Le robot utilise le middleware ROS. Nous avons expérimenté le traitement d'images pour le suivi des objets. Le traitement d'images est une tâche consommatrice de ressources. Le principe est le suivant : le FPGA capture les images de la caméra et utilise un algorithme filtrant les couleurs pour détecter la position de l'objet. Cette position est ensuite communiquée au contrôleur du robot via le middleware ROS.

Dans cette expérimentation, nous nous concentrons principalement sur l'implémentation de l'algorithme de traitement d'images - partie critique du nœud - sur FPGA. Le FPGA capture l'image de la caméra (pixel par pixel) et filtre chaque pixel à l'aide d'un filtre HSV. Les pixels filtrés sont ensuite utilisés pour calculer le barycentre de la région détectée qui fournit finalement la position de l'objet. La conception matérielle de ce processus peut être décrite à l'aide de notre métamodèle, mais ici nous utilisons un fichier VHDL existant d'un autre projet. Cela permet de démontrer la capacité de réutiliser des conceptions tierces existantes (rétrocompatibles) du métamodèle.

La première version de l'outil n'a fonctionné correctement qu'avec des petites images. Avec des formats d'images plus grands, la détection envoyait des résultats incorrects. Cela est probablement dû au fait que le filtre HSV prend plus de temps de traitement sur un pixel que le temps nécessaire pour qu'un nouveau pixel soit disponible.

Pour vérifier cette hypothèse, une solution a été d'utiliser le point d'arrêt matériel pour analyser le comportement du composant. Un modèle de compteur de pixels simple a été créé en utilisant l'unité de capture comme sous-circuit. L'option de débogage a été activée sur ce modèle. Après avoir déployé le compteur de pixels sur le FPGA, nous avons validé notre hypothèse : le traitement du filtre HSV prend du temps. Nous avons donc dupliqué ce filtre quatre fois et synthétisé ces 4 circuits en parallèle sur le FPGA.

4.4.7 Synthèse

Nous avons montré dans ces travaux qu'il était possible d'associer un environnement de développement orienté objet avec un environnement FPGA. Nous avons montré qu'il était possible de déboger à la volée un processus FPGA décentralisé.

Dans cette approche, nous avons montré qu'il est possible de coconstruire un logiciel réparti sur plusieurs environnements. Dans notre cas, nous avons utilisé l'environnement Pharo et un FPGA. Cette co-construction nous a conduits à proposer des outils adaptés : outre les outils standards de conception, nous avons proposé un environnement de simulation FPGA, associé à notre métamodèle exécutable, et nous avons proposé une méthode de débogage permettant de fixer un point d'arrêt directement sur le FPGA.

4.5 Bilan de l'axe et perspectives

Les CPS amènent de nombreux nouveaux problèmes. L'intégration de la partie matérielle, la prise en compte de l'environnement incertain et de communications denses entre les différents constituants sont quelques exemples que nous avons abordés dans ce chapitre. L'aspect dynamique de ces architectures nous encourage à repenser les outils que les développeurs utilisent pour vérifier et valider les applications.

Le principal problème que j'ai abordé dans cet axe est d'étendre les pratiques existantes en génie logiciel pour tenir compte de ces nouvelles dimensions. Dans une première partie, j'ai proposé une plateforme de simulation, permettant de tester différents algorithmes et les comparer. L'intérêt principal de cette plateforme est la répétabilité et la reproductibilité. Dans une deuxième partie, j'ai travaillé sur l'adaptation des tests logiciels aux environnements CPS. L'idée principale est de tester les capteurs avant les actionneurs, tout en intégrant le caractère incertain de l'environnement. Dans une troisième partie, j'ai travaillé sur un environnement de co-conception logiciel / FPGA intégrant un outil de débogage pour le FPGA.

Ce travail ne présente qu'une partie des problématiques d'outillage pour l'évolution des CPS. Nous avons identifié plusieurs verrous aux approches proposées :

— Pour le banc de simulation, nous devons envisager de déployer l'approche sur les systèmes CPS en général. Ce déploiement devrait être accompagné de la définition de scénarios plus diversifiés que la découverte de zone. Cela nécessitera une extension du fichier de configuration, plus adaptée aux systèmes CPS.

— L'environnement de tests a montré son efficacité. Certains travaux à court terme permettraient de valoriser cet environnement. D'une part, la répétabilité et l'automatisation des tests ne sont pas complètes. Actuellement, certains tests nécessitent une intervention humaine pour la configuration ou pour vérifier la conformité du résultat aux exigences. Idéalement, les environnements de tests devraient être instrumentés, de manière similaire à une chaîne de production automatisée. D'autre part, nous envisageons le développement de modélisation spécifique aux tests du système.

— Dans l'environnement de co-conception, il est possible de reconfigurer dynamiquement un logiciel sur un constituant, mais la reconfiguration matérielle à distance reconfigure et réinitialise l'ensemble du FPGA, même si le changement n'affecte qu'une petite partie du circuit. Cela provoque une interruption complète du FPGA, comprenant également les parties indépendantes. Une solution à ce problème consiste à ajouter la prise en charge de la reconfiguration partielle FPGA.

— L'environnement de co-conception a pour objectif de déployer des applications distribuées sur un réseau IoT. Différentes stratégies peuvent être imaginées et testées pour optimiser la répartition des ressources de calcul sur le réseau. Concrètement, la distribution d'une application sur un réseau IoT doit être planifiée en fonction des ressources – telles que la mémoire, le processeur ou l'état de la batterie – disponibles sur chaque nœud. Cela peut conduire au développement d'un ordonnanceur permettant de distribuer la charge sur chaque nœud.

4.6 Synthèse des contributions

Chacune des trois problématiques a fait l'objet de plusieurs publications.

La section 4.2 a été présentée dans :

— [63] : présentation technique de l'environnement de simulation multirobot utilisant le middleware ROS et l'environnement de simulation Morse.

— [62] : présentation des premières métriques pour le benchmarking de performance d'exploration multirobots.

— [33] : comparaison de différents algorithmes d'exploration multirobot.

— [9] : proposition du benchmarking de performance pour l'exploration multirobots.

— [31] : proposition des métriques de performance pour l'exploration multirobot.

— [3] : présentation de l'ensemble outillé de benchmarks et de l'analyse de deux algorithmes de fusion de cartes pour l'exploration multirobot.

La section 4.3 a été présentée dans :

— [64] : identification des verrous pour les tests automatisés des robots mobiles, et proposition de pistes de solutions.

— [52] : présentation d'une première architecture de tests pour les robots mobiles.

— [34] : présentation de la méthodologie de tests pour les robots.

La section 4.4 a été présentée dans :

— [51] : présentation de la méthodologie de co-conception entre les environnements Smalltalk et FPGA.

— [49] : présentation du métamodèle permettant d'exécuter du code dans les environnements Smalltalk et FPGA.

— [66] : présentation du framework global au GDR SocSiP.

— [26] : présentation de l'environnement de développement permettant la co-conception Smalltalk-FPGA.

CHAPITRE 5

Bilan et Perspectives

Contents

5.1 Bilan **80**
5.2 Transfert **83**
5.3 Perspectives **85**

La montagne nous offre le décor ... À nous d'inventer l'histoire qui va avec.
[Nicolas HELMBACHER]

En quelques mots...

Ce chapitre conclut ce manuscrit. Après une synthèse de mes travaux dans les trois axes présentés, j'aborde les perspectives en étendant chacun de ces axes. Je valorise chaque dimension par la description de liens forts entre chacun des axes.

Cette conclusion présente un bilan des contributions que j'ai abordées dans ce mémoire. Après cette synthèse, j'ouvre les perspectives de recherche et je détaille ma projection à moyen terme.

5.1 Bilan

Après avoir réalisé une thèse et un postdoctorat en Génie Logiciel, sur la maintenance du code source et d'architectures logicielles, mon parcours a évolué à deux reprises. Une première fois, dans le domaine des systèmes d'information distribués pour la robotique. Cette thématique, abordée durant les trois années effectuées à l'Unité de Recherche Informatique et Automatique de l'école des Mines de Douai, m'a permis d'étudier les systèmes de systèmes dans un environnement distribué. Durant cette période, j'ai étudié les outils logiciels pour la maintenance de systèmes cyberphysiques, particulièrement appliquée à la robotique de service.

Dans un deuxième temps, j'ai intégré le laboratoire DISP à l'Université Lumière Lyon 2, où je travaille depuis septembre 2015. Dans ce laboratoire, j'ai élargi mon domaine de compétences pour aborder l'agilité des systèmes d'information. J'ai ajouté à la partie compréhension de l'architecture et du fonctionnement des systèmes de systèmes, la considération de la sémantique des flux d'informations.

L'ensemble de mes travaux dans les deux laboratoires ont fait l'objet de suivis de thèses, d'un postdoctorat, de stages de master, de projets de recherche et un contrat de collaboration avec une entreprise.

Cette Habilitation à Diriger des Recherches a permis de retracer mon parcours dans l'agilité des systèmes d'information distribués. La problématique a été abordée selon le cycle de vie du système d'information, que je décompose en trois axes : la conception, l'exploitation et la maintenance de ces systèmes.

J'ai mis en oeuvre différents domaines de recherche, visible dans la Figure 5.1, présentant chacun des verrous scientifiques en lien avec le cycle de vie des systèmes d'information.

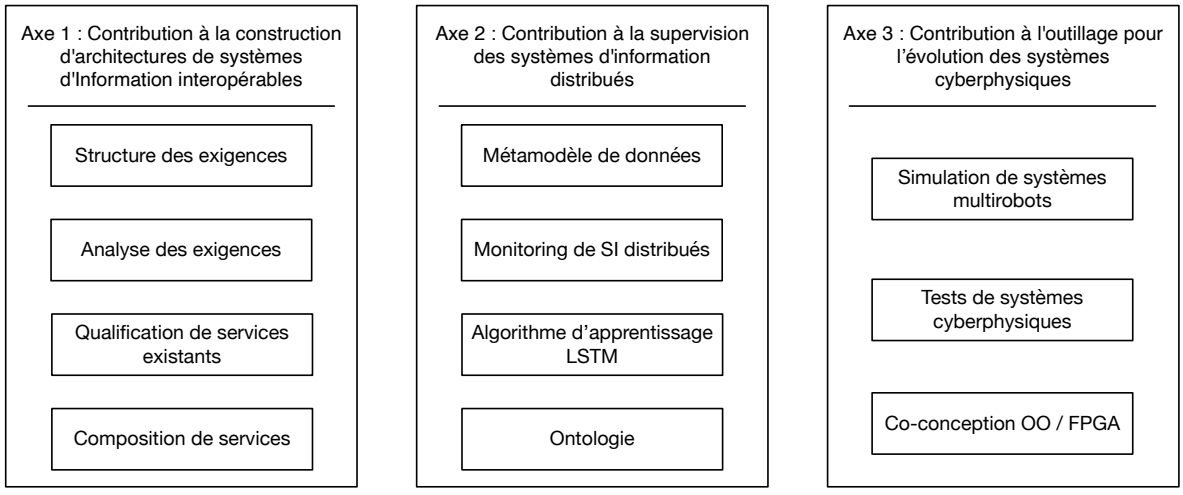

FIGURE 5.1 – Travaux réalisés sur l'ensemble du cycle de vie du système d'information.

L'introduction présente le contexte, la problématique et mon positionnement. Je définis le terme "agilité" dans le contexte des systèmes d'information. Cette définition est issue des articles rédigés avec Vincent Cheutet et la supervision d'un étudiant de Master [2, 28]. Mon positionnement dans le domaine des systèmes de systèmes est expliqué selon deux points de vue, d'une part selon mon parcours, d'autre part selon les thématiques abordées.

Ensuite, je présente mes travaux selon trois axes représentant un découpage du cycle de vie des systèmes d'information.

L'axe 1, "Contribution à la construction d'architectures de systèmes d'information interopérables", propose la création de systèmes d'information distribués à partir d'applications existantes. J'explique le besoin des entreprises de faire de la réutilisation d'applications. À partir de la formalisation des exigences, les travaux effectués nous ont amenés à créer des architectures servicisées. Les deux thèses, l'une soutenue en décembre 2019, la deuxième devant se terminer courant 2020, montrent la possibilité de définir des exigences et de les combiner avec des fonctionnalités de manière à construire ou adapter les systèmes d'information existants.

Dans ces travaux, nous avons identifié des verrous que nous pourrions aborder à court terme :

— L'identification des solutions émergentes au moment de la composition des services est une voie à explorer pour répondre aux exigences non résolues, ou pour identifier des futures fonctionnalités.

— Nos approches ne prennent pas en compte les écarts de propriétés non fonctionnelles du système de systèmes et de ses constituants. Les systèmes hérités par exemple ne sont pas adaptés à être exposés. Lorsqu'on les expose via des services, les exigences non fonctionnelles de ce système peuvent générer des interférences avec le système de systèmes dans lequel ils sont intégrés. Un travail est à réaliser sur l'impact de la composition de ces éléments sur les propriétés non fonctionnelles.

— Dans les deux approches présentées, chaque exigence est considérée séparément. Cette manière de faire n'est pas optimale. Nous pourrions développer une approche globale de manière à voir le système comme un tout et dérouler les analyses en considérant l'ensemble des exigences du système.

L'axe 2, "Contribution à la supervision des systèmes d'information distribués", propose de considérer le comportement du système d'information durant son exploitation. J'explique le besoin des entreprises de superviser les systèmes d'information distribués, et particulièrement les systèmes cyberphysiques. Les contraintes matérielles s'ajoutant à celles logicielles nous placent devant des verrous de l'ordre de l'instabilité du système, des risques de déconnexions d'une partie du système, et de la sécurité. Ces aspects ont tous trois des impacts sur la fiabilité. La supervision du système est abordée dans le cadre d'une collaboration avec l'entreprise Berger-Levrault et le recrutement de trois étudiants de Master. Cette collaboration se poursuit avec la contractualisation d'une thèse CIFRE.

La partie sécurité est abordée d'une manière originale, en considérant l'extraction des données à partir de sources d'information fiables. La thèse à ce sujet aborde la récupération et le traitement des informations venant de texte libre. L'approche propose le traitement de langage naturel avec la mise en place d'une architecture LSTM pour interroger la source d'information et extraire les données utiles pour ensuite pouvoir agir sur son propre système.

Dans ces travaux, nous avons identifié des verrous que nous pourrions aborder à court terme :

— Les deux approches étant complémentaires : l'une considère les évènements internes au système, l'autre une analyse des informations externes, nous devrions intégrer les deux approches pour compléter la supervision d'un système d'information distribué.

— Dans l'état actuel des travaux, seuls les évènements individuels sont considérés. La recherche des causes nécessite une analyse approfondie. Nous envisageons d'implémenter des méthodes analytiques telles que le traitement d'évènements complexes pour analyser les corrélations entre les données d'évènement.

L'axe 3, "Contribution à l'outillage pour l'évolution des systèmes cyberphysiques", propose de considérer l'intégration de la partie matérielle et de l'environnement incertain. Nous montrons que les outils de développement méritent d'être améliorés de manière à prendre en considération les nouvelles contraintes de ces systèmes. J'ai travaillé sur trois approches permettant de faciliter l'évolution des systèmes cyberphysiques. La première approche est un banc de simulations. L'idée ici est de mettre en avant la réutilisabilité et la reproductibilité de la simulation pour pouvoir comparer différents algorithmes. Ces travaux ont été réalisés dans le cadre de l'accompagnement d'un postdoctorat. La deuxième approche est l'adaptation des tests logiciels aux environnements cyberphysiques, en explicitant les contraintes de ces nouveaux systèmes. Ces travaux ont été réalisés dans le cadre du projet Roboshop. La troisième approche est un environnement de co-conception logiciel / FPGA, permettant d'améliorer les performances du logiciel par l'ajout de matériel en gardant le même environnement de travail. Ces travaux sont le résultat d'une thèse en codirection avec l'université de Brest.

Dans ces travaux, nous avons identifié des verrous que nous pourrions aborder à court terme :

— Le banc de simulation a montré le besoin dans des environnements multirobots de pouvoir utiliser des configurations explicites de manière à pouvoir comparer les algorithmes. D'une manière plus générale, nous pourrions étendre ce banc de simulation à des environnements cyberphysiques, en considérant les exigences spécifiques à ces systèmes.

— Le framework de tests nous a montré le besoin de structuration des tests logiciels pour les systèmes distribués. Nous avons réalisé ces tests sur des robots mobiles autonomes. Nous pourrions étendre le framework pour tester d'une part, des systèmes multi-robots, en ajoutant de nouvelles dimensions liées à l'interaction des différents constituants. D'autre part, la méthode devrait être étendue aux systèmes cyberphysiques en général.

— Les travaux réalisés dans le cadre de la co-conception logiciel / FPGA montrent qu'il est possible d'associer les paradigmes logiciel et matériel. La prochaine évolution est d'identifier les parties de code ayant des temps d'exécution importants par des techniques de profiling, d'isoler ces parties de code et de les projeter sur un matériel. Ceci pourrait être envisageable en cours d'exécution.

5.2 Transfert

Dans cette section, je montre l'intérêt de mes thématiques de recherche pour l'environnement dans lequel j'évolue. Je donne des éléments de développements, pour certains déjà engagés, selon trois axes : le transfert aux étudiants, le transfert aux entreprises et les projets nationaux et internationaux.

5.2.1 Transfert aux étudiants

Je développe des innovations pédagogiques. Elles sont valorisées dans des projets de recherches tels qu'Apache et Shyfte. Dans ces innovations, nous avons noté plusieurs verrous scientifiques à développer à court et moyen termes :

— mise en place de l'industrie du futur : La stratégie européenne est de valoriser la transformation digitale dans les industries. Cette transformation nécessite des investissements importants et implique des transformations en profondeurs du mode de fonctionnement des industries. Elle oblige également à repenser les métiers, par conséquent nos formations et les compétences à développer.

— les compétences à développer : Alors que la formation est en période de transition (réforme du Bac, réforme des DUT), la mise en évidence des compétences devient essentielle. Les approches par compétences, les formations organisées par blocs de compétences, l'évaluation des compétences sont autant de sujets qui remettent en question nos objectifs d'enseignement et nos manières d'enseigner. Des compétences en informatiques sont aujourd'hui largement recherchées dans toutes les formations. Les compétences comportementales, aujourd'hui recherchées en entreprise, doivent également être explicitées.

— la traçabilité : l'utilisation d'approches projet dans nos enseignements, notamment par l'utilisation d'ALPES, nous invite, comme dans tout projet, à développer des indicateurs de suivi. Dans le cadre de l'enseignement, identifier les moyens de traçabilité de la compréhension d'un cours par les étudiants reste un verrou à lever. L'utilisation des objets connectés pourrait être une solution à apporter dans le cadre des approches que nous développons déjà.

Actuellement, les projets ALPES, SoArm, APACHES, Shyfte, DIH encouragent le transfert de nos innovations pédagogiques et de nos résultats de recherche sur l'industrie du futur auprès des étudiants. La consolidation de ces travaux sera, à court terme, concrétisée par des articles de recherches (ALPES notamment), mais aussi par la création d'outils pédagogiques permettant de répondre aux verrous énoncés.

Par ailleurs, nous développons actuellement avec la métropole de Lyon et l'ensemble des acteurs de la formation de l'agglomération un démonstrateur de notre savoir-faire, qui devrait être présenté au salon Global Industrie 2021.

5.2.2 Transfert aux entreprises

Je travaille actuellement en collaboration avec l'entreprise Berger-Levrault. Nous abordons les thématiques de traçabilité de la donnée dans le cadre de l'analyse de l'interopérabilité de leurs solutions. Ces travaux de recherche ont par ailleurs été présentés dans l'axe 2 de ce manuscrit. Cette collaboration a mené à différents résultats et notamment au démarrage d'une thèse dans les mois à venir. Nous avons décidé de démarrer une deuxième thèse à propos de l'autoconfiguration à partir de l'analyse sémantique des données pour compléter nos travaux actuels.

En termes de transfert aux entreprises, notre expérience avec Berger-Levrault m'encourage à développer ce type de contrat et valoriser les éléments suivants :

— l'accompagnement des entreprises dans le processus de valorisation de la recherche. Parce que l'IUT dans lequel j'évolue professionnellement offre des formations uniquement en alternance, je suis en relation permanente avec des entreprises de la région. Cette proximité me permet de valoriser mes travaux de recherche et de les mettre en perspective. Je concrétise cet accompagnement par des conventions industrielles de formation par la recherche (CIFRE). Avec Berger-Levrault, avant la mise en place de la CIFRE, nous avons signé un contrat de collaboration de recherche. Je compte et suis en train de développer d'autres projets de ce type.
— le développement d'outillage utilisable. À de nombreuses reprises, ma recherche a conduit à la production de logiciels open source. Aller au-delà de la preuve de concept et proposer des outils utilisables, accompagner les entreprises et les développeurs dans le déploiement de ces outils est une des caractéristiques de la recherche en informatique appliquée. Je continuerai à promouvoir cette démarche pragmatique de la recherche.
— la rédaction d'articles de recherche avec les entreprises. Les entreprises nous apportent des problématiques complexes et des études de cas réelles. Je vais continuer, comme je le fais depuis ma thèse, à travailler sur les problématiques des entreprises, élevées en abstraction pour répondre à des verrous scientifiques et diffusées par des articles.

Le développement futur du transfert aux entreprises passera notamment par le "Digital Innovative Hub" (DIH). Ce projet nous permet d'assurer une visibilité régionale, nationale, internationale. Il permet de promouvoir nos compétences auprès des entreprises.

5.2.3 Projets nationaux et internationaux

Un chercheur doit aussi participer et porter des projets de recherche. Dans le cadre de mes thématiques, j'ai déjà participé à différents projets : européen (VFOS), erasmus (Shyfte), régionaux.

Par ailleurs, durant ces 5 dernières années, j'ai travaillé à la soumission de projets ANR JCJC et à la coordination d'une proposition de projet européen. Ces projets ont été soumis, évalués, mais pas retenus. Ils ne sont donc pas valorisés dans ce manuscrit. Pour autant, l'expérience acquise dans chacun d'eux est importante pour améliorer et réussir le dépôt des projets suivants.

La participation à ces projets est un moment de partage particulièrement efficace en termes de développement de la recherche. A terme, je continuerai le développement de projets :

— européens : la commission européenne ouvre, chaque année, des appels à proposition ICT. En juillet 2019, lors de la présentation des projets 2019/2020, parmi les quatre défis technologiques, l'intelligence artificielle pour l'industrie du futur est un axe fort dans lequel mes travaux actuels et futurs prennent place.
— ANR : actuellement, je participe au dépôt d'un projet ANR sur la mesure dynamique de l'alignement des systèmes d'information (en lien avec les perspectives proposées ci-dessous (page 87). Ce projet est porté par Virginie Goepp de l'INSA Strasbourg.
— nationaux : je compte continuer à développer mon réseau pour établir des projets inter-laboratoire sur des thématiques spécifiques. Notamment, j'envisage de développer des partenariats avec des laboratoires de recherche en sécurité sur le sujet du monitoring pour la sécurité proposé dans l'axe 2.

5.3 Perspectives

J'ai présenté mes travaux selon les 3 axes présentés en Figure 5.1. Les perspectives pour ma recherche sont nombreuses de par les différentes thématiques déjà abordées, mais également par les verrous existants et émergents dans le domaine des systèmes de systèmes. D'après l'article [Cadavid 2020a], les phases d'évaluation, d'implémentation et de maintenance des systèmes de systèmes sont des sujets insuffisamment abordés dans la littérature. Il est de même pour la flexibilité et l'adaptabilité. L'originalité de mes travaux réside dans la considération de ces aspects.

Je présente mes perspectives selon trois volets : (i) l'auto-construction des systèmes d'information interopérables, (ii) la surveillance de la dynamique des systèmes cyberphysiques, (iii) l'aide à l'évolution des systèmes d'information flexibles et ouverts. Ces trois axes sont présentés dans la Figure 5.2. Cette figure reprend la Figure 5.1 que j'étends de deux manières : l'approfondissement et l'élargissement. Le travail d"approfondissement est une mise en perspective de mes travaux dans le même axe du cycle de vie, alors que l'élargissement est une ouverture des thématiques vers les autres axes du cycle de vie du système, et par conséquent un rapprochement de ces axes.

Pour le premier volet, je mène une réflexion sur la formalisation de capacités logicielles. Cette réflexion nous amènera à approfondir l'aspect sémantique pour répondre au mieux aux exigences du système visé et de contribuer à l'auto-construction du système d'information à partir de la définition fonctionnelle de chaque constituant. J'étends ces travaux vers l'axe 2 en proposant d'aligner les données d'utilisation des constituants avec les besoins des utilisateurs.

Pour le deuxième volet, je mets en avant les résultats de la supervision des systèmes, pour les étendre vers l'axe 3 et alimenter une maintenance prédictive du système cyberphysique. L'objectif est d'anticiper les réactions du système en fonction du comportement des constituants.

Pour le troisième volet, je valorise les travaux de la validation et la vérification pour les étendre vers l'axe 1 et les intégrer tout au long du cycle de vie en appui à l'agilité du système. Ainsi, nous pourrons valoriser l'auto-construction et l'auto-adaptation du système d'information grâce à des constituants vérifiés et validés.

Par ailleurs, durant toutes nos expérimentations, nous avons constaté qu'il est particulièrement difficile de comparer les résultats et de réutiliser les autres approches. Je compte structurer ma recherche de manière à apporter des réponses à ces problématiques. De manière à valoriser l'applicabilité de mes travaux, je compte m'atteler à l'application de bonnes pratiques, notamment, le partage d'expérimentation avec la communauté de chercheurs par la diffusion de plateformes de tests communes, des Gold standards, ou encore des algorithmes réutilisables facilement via des API, ou documentation d'installation.

5.3.1 Perspectives 1 : vers l'auto-construction des systèmes d'information interopérables

Au vu de mes précédentes réalisations de l'axe 1 "Contribution à la construction d'architectures de systèmes d'information interopérables", dans lesquelles nous avons créé des architectures servicisées à partir des besoins, il serait possible d'automatiser la configuration du SI à partir des mêmes données d'entrée. Ainsi, nous pourrions envisager des systèmes d'information plus flexibles et répondants plus exactement aux exigences des entreprises.

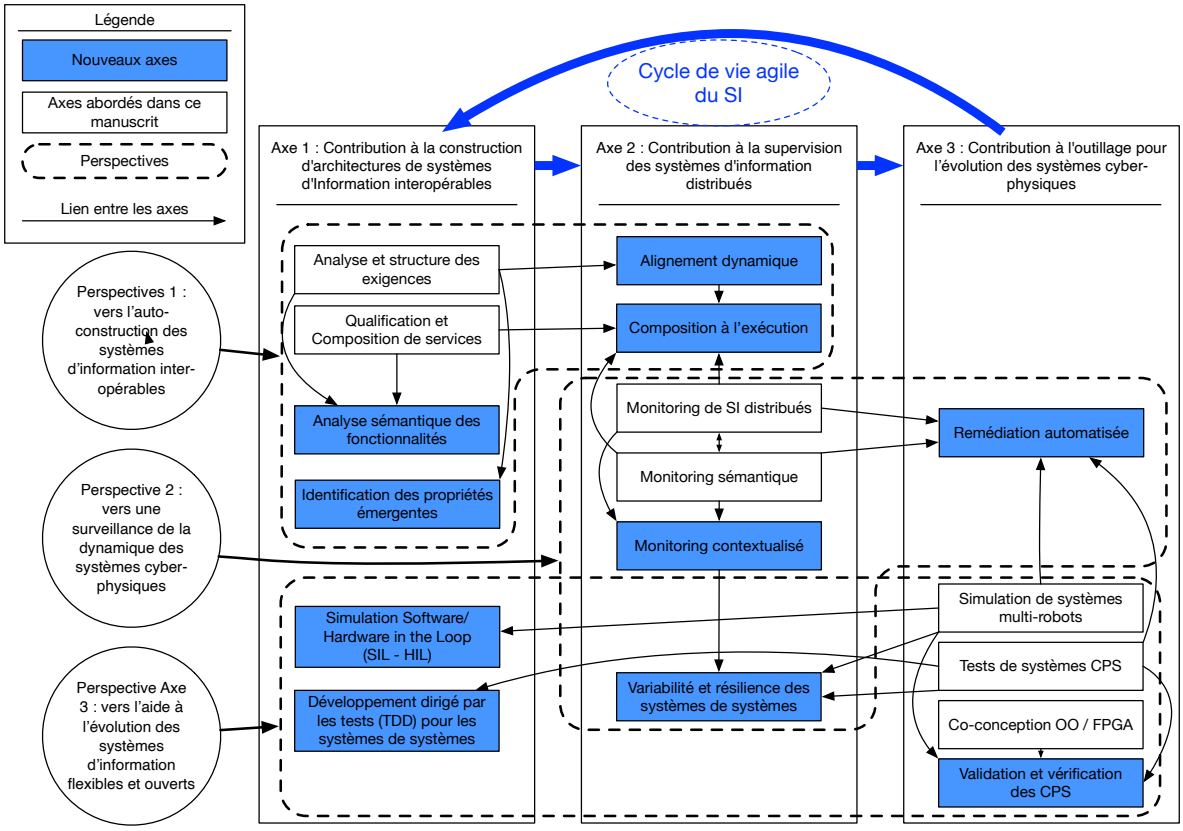

FIGURE 5.2 – Perspectives de recherche

Dans la littérature scientifique, de nombreux verrous persistent pour assurer la construction automatique des systèmes d'information interopérables. Des travaux utilisent des approches GQM (Goal Question Metrics) [Chaabane 2019] pour décrire des architectures de système de systèmes. Les verrous énoncés sont le passage à l'échelle, ou encore la découverte et la prise en charge des propriétés émergentes du système de systèmes [Raman 2019].

Une possibilité d'approfondissement est d'aller plus en avant dans l'analyse sémantique des exigences et des capacités des logiciels. L'extension de l'analyse sémantique des logiciels est de considérer les exigences comme un ensemble. Dans cette analyse, il est également nécessaire d'intégrer les exigences non fonctionnelles. Ainsi, nous pourrions identifier dès la conception des propriétés émergentes.

Une extension vers la supervision du système est de réaliser l'analyse des exigences de manière dynamique. En associant mes travaux sur l'analyse des exigences de l'axe 1 avec les résultats de la supervision de l'axe 2, nous pouvons aller vers l'alignement dynamique du système d'information et la composition à chaud des systèmes de systèmes, en utilisant par exemple l'auto-orchestration des microservices. Des travaux similaires sont déjà réalisés dans le domaine, utilisant les évènements apparaissant à l'exploitation du système [Monteiro 2018]. Dans ces travaux, les verrous énoncés sont l'autoadaptation, en plus de la flexibilité et de la fiabilité du système construit.

Pour l'alignement des exigences avec les fonctionnalités du système, j'ai déjà initié un travail sur

l'alignement dynamique des systèmes d'information, lors de l'encadrement d'un étudiant de Master, un article accepté dans la conférence ILS 2020 [10], et la soumission d'un projet ANR en 2020 ayant obtenu un avis favorable à l'étape 1.

À plus long terme, mon objectif est d'utiliser des algorithmes d'apprentissage automatique pour améliorer le processus de conception des systèmes de systèmes. En effet, lors de la conception, de nombreuses informations métiers sont explicitées et peu exploitées par la suite. La complexité des systèmes de systèmes rend difficile la prise en compte de l'ensemble des exigences (métiers, fonctionnelles et non fonctionnelles). Utiliser l'apprentissage automatique comme outils d'aide à la décision permettrait aux architectes de répondre plus précisément aux exigences.

5.3.2 Perspective 2 : vers une surveillance de la dynamique des systèmes cyberphysiques

Dans les réalisations présentées dans l'axe 2 de ce manuscrit, j'ai montré que la supervision des systèmes distribués est indispensable pour contrôler et assurer le fonctionnement du système dans son ensemble. Nous avons montré aussi que les analyses sémantiques permettaient de rendre plus efficace la prise en charge des problèmes potentiels d'un système complexe.

Dans la littérature scientifique, de nombreux verrous persistent pour assurer la surveillance de la dynamique des systèmes cyberphysiques. Ces systèmes ont la particularité d'être hétérogènes, de comporter des composantes matérielles et logicielles, incluant des composants dits "plug-and-play". Les systèmes cyberphysiques permettent de mettre en relation de nombreux capteurs, échangeant de grandes quantités de données tout en nécessitant une faible latence [Buyya 2016]. La difficulté aujourd'hui est d'être certains que toutes les données produites seront consommées dans un laps de temps raisonnable. Plusieurs solutions existent à différents niveaux : embarquer des algorithmes de contrôle dans les capteurs, adapter le dimensionnement du réseau, utiliser des logiciels d'orchestration de données. Ces systèmes nécessitent des mécanismes de reconfiguration dynamique et de surveillance adaptés [Vora 2003, Dávid 2018, Mohsin 2019].

L'augmentation des volumes de données échangées implique la prise en compte de la variabilité de la fréquence de production des données. Cela nécessite que les architectures distribuées (à la fois en infrastructure et en flux) soient adaptables ou autoadaptables [Gascon-Samson 2015] afin de renforcer la résistance du système aux dysfonctionnements en garantissant des interactions fiables tout en évitant les phénomènes de congestion.

La disponibilité des ressources utilisées dans les architectures d'échange doit être garantie pour assurer la fiabilité du système. La variabilité des types et des fréquences de données échangées peut être gérée en appliquant le principe d'élasticité [Al-Dhuraibi 2017]. Ce principe permet d'automatiser l'adaptation du système en fonction des constituants disponibles. Cela nécessite la définition d'un ensemble de configurations d'architectures d'échange (optimisées, stables, dégradées, etc.) et de mécanismes permettant de passer d'une configuration à une autre, afin de permettre la résilience du système d'échange. La capacité de résilience de ces systèmes permet d'assurer le fonctionnement, mais peut cacher une complexité des problèmes bien souvent difficile à maitriser pour revenir à un état stable.

La suite de mes travaux, dans cet axe, aborde les problématiques liées à la surveillance contextualisée et intelligente et à l'adaptation dynamique de l'architecture.

Nous avons commencé à faire de l'analyse sémantique dans nos travaux sur la cybersécurité.

Par ailleurs, LSTM s'est révélé être une approche intéressante pour l'analyse de texte libre dans le domaine spécifique de la cybersécurité. Nous allons mener une étude pour appliquer cette approche dans le but d'identifier les symptômes et les causes potentielles d'un problème en présence d'un grand nombre de données.

J'aborde cet axe avec l'entreprise Berger-Levrault. Nous démarrons une thèse CIFRE dans le domaine de la supervision de l'architecture du système d'information distribué. Une deuxième thèse CIFRE est en cours de rédaction pour consolider les analyses sémantiques du système et la reconfiguration automatique.

Par ailleurs, nous allons étendre ces travaux vers l'axe maintenance du cycle de vie pour assurer l'évolution des systèmes cyberphysiques. L'objectif étant d'avoir des systèmes sûrs, il faudrait aborder la sûreté de fonctionnement, l'analyse de fiabilité en allant jusqu'à la maintenance. Ainsi, pour compléter les activités de maintenance et d'évolution des systèmes cyberphysiques, nous allons utiliser les activités de monitoring pour étendre la maintenance corrective et évolutive vers la maintenance prédictive, et envisager le développement de pronostics de défaillance ("Prognostics and Health Management" – PHM) du système et de ses constituants. Dans le domaine de la mécanique [Atamuradov 2017], le pronostic de défaillance transforme un ensemble de données recueillies sur l'équipement surveillé en indicateurs de santé du système. Ces données sont ensuite extrapolées dans le temps pour aider à la maintenance du système avant défaut. J'envisage de développer ce type de solutions prévisionnelles de maintenance sur les systèmes cyberphysiques. Pour cela, je compte étendre mes travaux sur la supervision des systèmes d'information en les couplant avec des approches de pronostics de défaillance pour avoir une gestion fine du cycle de vie des composants du système.

À long terme, mon objectif est de pouvoir rendre explicable et maîtrisable l'évolution des systèmes cyberphysiques. De même que pour les travaux présentés dans ce mémoire, l'originalité de tels travaux réside dans l'explicitation de propriétés et la rationalisation des comportements de systèmes variables. Plus précisément, il s'agit de mettre en oeuvre des outils de supervision, des algorithmes de reconnaissance de patrons de comportements et d'adapter le système de manière à le maintenir fiable.

Ces différentes étapes seront à terme supportées par des algorithmes d'apprentissage automatique ou d'autres algorithmes d'intelligence artificielle. Aujourd'hui, ces technologies ne sont pas suffisamment exploitées dans le domaine, les utiliser permettrait d'améliorer les processus de supervision. Par ailleurs, ces technologies méritent d'être comprises pour que le système de décision reste traçable et compréhensible.

5.3.3 Perspective Axe 3 : vers l'aide à l'évolution des systèmes d'information flexibles et ouverts

Les réalisations présentées dans l'axe 3 de ce manuscrit s'attachent à développer des outils permettant de comprendre et de fiabiliser les systèmes cyberphysiques. Nous avons développé des tests permettant de valider le comportement d'un robot. Nous avons réalisé un environnement de simulation pour les systèmes multirobots et nous avons proposer un outillage de débogage dans le cadre d'un environnement de co-conception logiciel/matériel.

Dans la littérature scientifique, de nombreux verrous persistent pour assurer l'aide à l'évolution des systèmes d'information flexibles et ouverts. Les verrous des systèmes de systèmes ont été référencés dans [Dahmann 2014, Guariniello 2019]. Ces verrous posent les questions d'intégration des constituants, de leurs collaborations, du rôle de chaque constituant, de leurs capacités à répondre

aux besoins, de leurs processus de validation... Chaque verrou est bien souvent traité séparément des autres, car chacun est un problème complexe. Par ailleurs, la résolution des dysfonctionnements se fait bien souvent de manière ad hoc. Outre le coût de développement et de correction que cela implique, cela ne répond pas aux exigences de réactivité de la technologie.

Mes perspectives dans cet axe sont d'assurer l'évolution des systèmes cyberphysiques sans rupture de service notamment en utilisant des stratégies de validation et de vérification dynamiques. L'objectif est de rendre l'évolution des systèmes de systèmes explicable et fiable, notamment grâce à l'historisation, aux outils de vérification et à la validation de l'architecture de manière dynamique.

Dans le cas des tests, nous devons étendre l'approche aux systèmes de systèmes et considérer des formes multiconstituants. Il sera intéressant d'aborder les problématiques de tests en présence de constituants hétérogènes. Pour la simulation, nous devons étendre nos solutions vers les verrous liés à l'intégration de matériel dans la boucle ("hardware in the loop"). Le principal problème est de trouver des approches permettant la simulation et les tests pour vérifier et valider le comportement du système en cours de fonctionnement.

Il est pertinent de considérer que la validation et la vérification doivent être présentes à chaque étape du cycle de vie du logiciel. Ainsi, nous devons aller vers des frameworks complets intégrant la validation et la vérification dès la conception. Pour cela, nous allons chercher à intégrer les propriétés non fonctionnelles comme des exigences explicites.

L'originalité de la démarche est d'apporter une approche outillée et intégrant, dès la conception, les tests et la simulation. L'intégration des tests et de la simulation dès la conception a déjà fait ses preuves dans le génie logiciel, notamment par le développement dirigé par les tests (TDD) [Beck 2003]. Nous allons développer ces approches sur les systèmes cyberphysiques. La simulation sera complémentaire aux tests dans l'explicitation des exigences non fonctionnelles, où nous pouvons envisager d'étendre nos précédents travaux, par exemple en intégrant de nouvelles données dans le fichier de configuration du banc de simulation.

À plus long terme, nous exploiterons mes travaux de co-conception et les étendrons aux problématiques des systèmes distribués et des nouvelles architectures. Aujourd'hui, la migration vers les microservices est une activité importante dans les entreprises [Di Francesco 2017, Plakidas 2018, Cojocaru 2019]. Il est clair que certaines parties des systèmes méritent de migrer vers les microservices et ont besoin de plus d'outillage de validation et de vérification.

Par ailleurs, l'arrivée de nouvelles technologies, comme la blockchain, qui permet d'améliorer la sécurisation des données, posent les mêmes questions sur l'intérêt de la migration, comment migrer, et quels éléments migrer [Zheng 2017, Porru 2017]. L'utilisation de la blockchain est en forte croissance depuis 3 ans dans de nombreux secteurs d'activités. Ces développements sont coûteux et complexes, principalement dus au manque de maturité de la technologie. Selon Moody's, la blockchain deviendra un standard en 2021. La standardisation permettra de gagner du temps et des coûts sur l'adoption de la technologie et ensuite sur la disponibilité des données. Elle améliorera également l'interopérabilité entre blockchains et systèmes hérités [Moody's 2019]. Nous devons accompagner les entreprises vers la migration sur ces technologies.

Mon objectif est de réutiliser les différentes approches que j'ai développées ces dernières années dans la co-conception, la simulation, la supervision, la modélisation pour offrir un système d'aide à la décision dans l'utilisation des nouvelles architectures et la migration des systèmes d'information.

Table des figures

1.1 Positionnement d'un système de systèmes 5
1.2 Évolution chronologique de mes travaux de recherche 6
1.3 Structure du manuscrit 8

2.1 Le framework EACP 16
2.2 Le Métamodèle EACP 17
2.3 Le processus d'exploitation EACP 20
2.4 Modèle du framework 23
2.5 Modèle d'analyse des fonctions 24
2.6 Profil de Capacité de l'ISO 16100 25

3.1 Le framework Pulse 35
3.2 Processus de gestion de version avec Pulse 36
3.3 Le Métamodèle Pulse 37
3.4 Processus de récupération des informations avec Pulse 42
3.5 Nombre de messages par seconde sur une file d'attente 43
3.6 Structure du système analysé 43
3.7 Exemple de description de vulnérabilité 45
3.8 Principe de la reconnaissance d'entités nommées et de l'extraction de relations . . . 46
3.9 Framework d'identification des vulnérabilités 47
3.10 Ontologie utilisée pour identifier les vulnérabilités 48
3.11 Fonctionnement de l'architecture LSTM-CRF 49

4.1 Architecture du banc de simulations 59
4.2 Exemple de fichier de configuration du banc de simulation. 61
4.3 Architecture logicielle d'un robot 62
4.4 Les cinq niveaux de tests de la méthodologie 66
4.5 Testing the Robot Sensors in a Static Known Environment 68
4.6 Métamodèle OoRC 73
4.7 Exemple d'un circuit généré 74

5.1 Travaux réalisés sur l'ensemble du cycle de vie du système d'information. 80

5.2 Perspectives de recherche 86

Liste des tableaux

3.1 Exemples d'exigences d'interopérabilité, problèmes d'interopérabilité, indicateurs et causes potentielles 41
3.2 Performance pour l'ensemble des entités 50
3.3 Détails des performances des modèles LSTM-CRF et CRF 50
3.4 Paramètres du modèle LSTM-CRF 50
3.5 Résultats des variations des paramètres du modèle 51
3.6 Performance des trois modèles d'extraction des relations 51

Production scientifique

Journaux internationaux avec comité de lecture

[1] Houssem Gasmi, Jannik Laval and Abdelaziz Bouras. *Information Extraction of Cybersecurity Concepts : An LSTM Approach*. Applied Sciences, vol. 9, no. 19, 2019.

[2] Jannik Laval, Chantal Cherifi and Vincent Cheutet. *Towards the measurement of Enterprise Information Systems agility to support EIS improving projects*. International Journal of Agile Systems and Management, vol. 11, no. 3, pages 222–246, 2018.

[3] Zhi Yan, Luc Fabresse, Jannik Laval and Noury Bouraqadi. *Building a ROS-Based Testbed for Realistic Multi-Robot Simulation : Taking the Exploration as an Example*. Robotics, vol. 6, no. 3, page 21, September 2017.

[4] Jannik Laval and Stéphane Ducasse. *Resolving cyclic dependencies between packages with enriched dependency structural matrix*. Software : Practice and Experience, vol. 44, no. 2, pages 235–257, February 2014.

[5] Jannik Laval, Nicolas Anquetil, Usman Bhatti and Stéphane Ducasse. *oZone : Layer identification in the presence of cyclic dependencies*. Science of Computer Programming, vol. 78, no. 8, pages 1055 – 1072, 2013. Special section on software evolution, adaptability, and maintenance & Special section on the Brazilian Symposium on Programming Languages.

[6] Karine Mordal, Nicolas Anquetil, Jannik Laval, Alexander Serebrenik, Bogdan Vasilescu and Stéphane Ducasse. *Software quality metrics aggregation in industry*. Journal of Software : Evolution and Process, vol. 25, no. 10, pages 1117–1135, October 2013.

[7] Jannik Laval, Jean-Rémy Falleri, Philippe Vismara and Stéphane Ducasse. *Efficient Retrieval and Ranking of Undesired Package Cycles in Large Software Systems*. The Journal of Object Technology, vol. 11, no. 1, page N/A, April 2012.

[8] Jannik Laval, Simon Denier, Stéphane Ducasse and Jean-Rémy Falleri. *Supporting simultaneous versions for software evolution assessment*. Science of Computer Programming, vol. 76, no. 12, pages 1177 – 1193, 2011. Special Issue on Software Evolution, Adaptability and Variability.

*

Journaux nationaux avec comité de lecture

[9] Zhi Yan, Luc Fabresse, Jannik Laval, Noury Bouraqadi. Benchmarking de performance pour l'exploration multi-robots. In *Journal RIA, numéro spécial de la Revue d'Intelligence Artificielle consacré à une version étendue des meilleurs articles des JFSMA 2014 et 2015*. no 1-2/2016, 211-236.

Conférences internationales avec comité de lecture

[10] Hamza Darii, Jannik Laval, Valérie Botta-Genoulaz and Virginie Goepp. *Measurement of the Alignment of Information Systems*. In the 8th International Conference on Information Systems, Logistics and Supply Chain (ILS), Austin, TX, US, April 22-24, 2020.

[11] Jannik Laval, Nawel Amokrane, Nejib Moalla and Mustapha Derras. *Analysis of Data Exchange among Heterogeneous IoT Systems*. In 10th International Conference on Interoperability for Enterprise Systems And Applications (I-ESA 2020), Tarbes, France, March 24-27, 2020.

[12] Houssem Gasmi, Jannik Laval and Abdelaziz Bouras. *Cold-start cybersecurity ontology population using information extraction with LSTM*. In 2019 International Conference on Cyber Security for Emerging Technologies (CSET).

[13] Mathieu Vermeulen, Abir Karami, Anthony Fleury, François Bouchet, Nadine Mandran, Jannik Laval and Jean-Marc Labat. *APACHES : Human-Centered and Project-Based Methods in Higher Education*. In Maren Scheffel, Julien Broisin, Viktoria Pammer-Schindler, Andri Ioannou and Jan Schneider, editeurs, Transforming Learning with Meaningful Technologies, pages 683–687, Cham, 2019. Springer International Publishing.

[14] Abir M'baya, Jannik Laval and Nejib Moalla. *An Assessment Conceptual Framework for the Modernization of Legacy Systems*. In Keith Popplewell, Klaus-Dieter Thoben, Thomas Knothe and Raúl Poler, editeurs, Enterprise Interoperability VIII, pages 347–357, Cham, 2019. Springer International Publishing.

[15] Hamza Lahbabi, Aicha Sekhari, Omar El Beqqali and Jannik Laval. *MBaaS-based Authentication service for handicraft products traceability system*. In 12th International Conference on Software, Knowledge, Information Management Applications (SKIMA), pages 1–5, Dec 2018.

[16] Houssem Gasmi, Jannik Laval, Abdelaziz Bouras. LSTM Recurrent Neural Networks for Cybersecurity Named Entity Recognition. In *the Thirteenth International Conference on Software Engineering Advances ICSEA 2018*, Nice, France, October 14, 2018 to October 18, 2018.

[17] Abdelhadi Belfadel, Emna Amdouni, Jannik Laval , Chantal Cherifi and Néjib Moalla. *Ontology-based Software Capability Container for RESTful APIs*. In 2018 International Conference on Intelligent Systems (IS), pages 466–473, Sep. 2018.

[18] Rami Torkhani, Jannik Laval , Hedi Malek and Néjib Moalla. *Intelligent Framework For Business Process Automation And Re-engineering*. In 2018 International Conference on Intelligent Systems (IS), pages 624–629, Sep. 2018.

[19] Nawel Amokrane, Jannik Laval , Philippe Lanco, Mustapha Derras and Néjib Moala. *Analysis of Data Exchanges, Contribution to Data Interoperability Assessment*. In 2018 International Conference on Intelligent Systems (IS), pages 199–208, Sep. 2018.

[20] Abdelhadi Belfadel, Jannik Laval, Chantal Cherifi and Néjib Moalla. *Towards service orchestration through software capability profile*. In 9th International Conference on Interoperability for Enterprise Systems And Applications (I-ESA 2018), Berlin, Germany, March 2018.

[21] Rami Torkhani, Jannik Laval, Hedi Mallek and Néjib Moalla. *Collaborative Information System Toward A Robotic Process Automation Solution*. In 9th International Conference on Interoperability for Enterprise Systems And Applications (I-ESA 2018), Berlin, Germany, March 2018.

[22] Abir M'baya, Jannik Laval and Néjib Moalla. *An assessment conceptual framework for the modernization of legacy systems*. In 2017 11th International Conference on Software, Knowledge, Information Management and Applications (SKIMA), pages 1–11, Dec 2017.

[23] Abdehadi Belfadel, Jannik Laval , Chantal Cherifi and Néjib Moalla. *Capability profile for enterprise application integration*. In 2017 International Conference on Engineering, Technology and Innovation (ICE/ITMC), pages 1333–1337, June 2017.

[24] Rami Torkhani, Jannik Laval, Mustapha Derras and Néjib Moalla. *Conceptual interoperability framework for software development and maintenance*. In 23rd IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC 2017), pages 1327–1332, Funchal, Portugal, June 2017. IEEE.

[25] Mathieu Vermeulen, Jannik Laval, Xavier Serpaggi and Rémy Pinot. *Soyez agiles dans les A.L.P.E.S.! Une pedagogie en mode agile.* In 9ème Colloque Questions de Pédagogie dans l'Enseignement Supérieur (QPES 2017), Grenoble, France, June 2017.

[26] Xuan Sang Le, Jean-Christophe Le Lann, Loïc Lagadec, Luc Fabresse, Noury Bouraqadi and Jannik Laval . *CaRDIN : An Agile Environment for Edge Computing on Reconfigurable Sensor Networks*. In 2016 International Conference on Computational Science and Computational Intelligence (CSCI), pages 168–173, Dec 2016.

[27] Abir M'Baya, Jannik Laval, Néjib Moalla, Yacine Ouzrout, and Abdelaziz A Bouras. Ontology based multi criteria recommended system to guide internship assignment process. In *CSCI 2016 - International Conference on Computational Science and Computational Intelligence*, Las Vegas, United States, December 2016.

[28] Fatima Zohra Tanane, Jannik Laval and Vincent Cheutet. *Towards assessment of information system agility*. In 2016 10th International Conference on Software, Knowledge, Information Management Applications (SKIMA), pages 348–353, Dec 2016.

[29] Abir M'Baya, Jannik Laval, Néjib Moalla, Yacine Ouzrout and Abdelaziz Bouras. *Ontology Based System to Guide Internship Assignment Process*. In 2016 12th International Conference on Signal-Image Technology Internet-Based Systems (SITIS), pages 589–596, Nov 2016.

[30] Jannik Laval and Mathieu Vermeulen. *LES A.L.P.E.S. : APPROCHES AGILES POUR L'ENSEIGNEMENT SUPERIEUR*. In 29ème Congrès de l'Association Internationale de Pédagogie Universitaire (AIPU 2016), Lausanne, Switzerland, June 2016. AIPU.

[31] Zhi Yan, Luc Fabresse, Jannik Laval and Noury Bouraqadi. *Metrics for performance benchmarking of multi-robot exploration*. In 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 3407–3414, Sep. 2015.

[32] Mathieu Vermeulen, Anthony Fleury, Kathy Fronton, Jannik Laval. LES A.L.P.E.S. : Approches agiLes Pour l'Enseignement Supérieur. In *Conférence. Questions de Pédagogies dans l'Enseignement Supérieur (QPES 2015)*, 17-19 juin 2015, Brest, France. Page 243-248.

[33] Zhi Yan, Luc Fabresse, Jannik Laval and Noury Bouraqadi. *Team Size Optimization for Multi-robot Exploration*. In Davide Brugali, Jan F. Broenink, Torsten Kroeger and Bruce A. MacDonald, editeurs, Simulation, Modeling, and Programming for Autonomous Robots, pages 438–449, Cham, 2014. Springer International Publishing.

[34] Jannik Laval, Luc Fabresse and Noury Bouraqadi. *A methodology for testing mobile autonomous robots*. In 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 1842–1847, Nov 2013.

[35] Jean-Rémy Falleri, Simon Denier, Jannik Laval, Philippe Vismara and Stéphane Ducasse. *Efficient Retrieval and Ranking of Undesired Package Cycles in Large Software Systems*. In Judith Bishop and Antonio Vallecillo, editeurs, Objects, Models, Components, Patterns, pages 260–275, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[36] Nicolas Anquetil and Jannik Laval. *Legacy Software Restructuring : Analyzing a Concrete Case*. In 2011 15th European Conference on Software Maintenance and Reengineering, pages 279–286, March 2011.

[37] Karine Mordal-Manet, Jannik Laval, Stéphane Ducasse, Nicolas Anquetil, Françoise Balmas, Fabrice Bellingard, Laurent Bouhier, Philippe Vaillergues and Thomas J. McCabe. *An Empirical Model for Continuous and Weighted Metric Aggregation*. In 2011 15th European Conference on Software Maintenance and Reengineering, pages 141–150, March 2011.

[38] Karine Mordal-Manet, Françoise Balmas, Simon Denier, Stéphane Ducasse, Harald Wertz, Jannik Laval, Fabrice Bellingard and Philippe Vaillergues. *The squale model — A practice-based industrial quality model*. In 2009 IEEE International Conference on Software Maintenance, pages 531–534, Sep. 2009.

[39] Jannik Laval, Simon Denier, Stéphane Ducasse and Alexandre Bergel. *Identifying Cycle Causes with Enriched Dependency Structural Matrix*. In 2009 16th Working Conference on Reverse Engineering, pages 113–122, Oct 2009.

Editeur invité de revue

[40] Anne Etien and Jannik Laval. *Advances in Dynamic Languages*. Science of Computer Programming, vol. 161, pages 1–171, 2018. Advances in Dynamic Languages.

[41] Jannik Laval and Andy Kellens. *Advances in Dynamic Languages*. Science of Computer Programming, vol. 98, pages 317–438, 2015. Special Issue on Advances in Dynamic Languages.

Livre

[42] Alexandre Bergel, Damien Cassou, Stéphane Ducasse and Jannik Laval. Deep into Pharo. Square Bracket Associates, September 2013.

Chapitres de livre

[43] Nawel Amokrane, Jannik Laval, Néjib Moalla, Mustapha Derras. *Analysis of Data Exchanges, Towards a Tooled Approach for Data Interoperability Assessment.*. In Studies Comp.Intelligence, Ricardo Jardim-Goncalves et al : Intelligent Systems : theory, research and innovation in applications, vol 864, 978-3-030-38703-7.

[44] Jannik Laval, Simon Denier and Stéphane Ducasse. *Visualisations pour la remodularisation à large échelle des systèmes à objets*. In Hermès, editeur, Évolution et rénovation des systèmes logiciels. Hermès, November 2011.

[45] Karine Mordal-Manet, Jannik Laval and Stéphane Ducasse. *Modèles de mesure de la qualité des logiciels*. In Hermès, editeur, Évolution et rénovation des systèmes logiciels. Hermès, November 2011.

Workshops internationaux avec comité de lecture

[46] Nawel Amokrane, Jannik Laval, Philippe Lanco, Mustapha Derras and Nejib Moalla. *Analysis of Information Exchanges among Distributed IS, an Industrial Case* In IWST 18, Cagliari, Italy, September 10-14th, 2018.

[47] Nicolas Anquetil, Muhammad Usman Bhatti, Stéphane Ducasse, André Hora and Jannik Laval. *The Case for Non-Cohesive Packages*. In SQAMIA 2017 - 6th workshop on Software Quality Analysis, Monitoring, Improvement, and Applications, Belgrade, Serbia, September 2017.

[48] Abir M'Baya, Jannik Laval, Néjib Moalla, Yacine Ouzrout and Abdelaziz Bouras. *Ontology based multi criteria recommendation system to guide internship assignment process*. In 9th IEEE International Workshop on Knowledge Acquisition Reuse & Evaluation (KARE 2016), Naples, Italy, November 2016.

[49] Xuan Sang Le, Loïc Lagadec, Luc Fabresse, Jannik Laval and Noury Bouraqadi. *A meta-model supporting both hardware and smalltalk-based execution of fpga circuits*. In Proceedings of the International Workshop on Smalltalk Technologies, IWST '15, pages 6 :1–6 :14, New York, NY, USA, 2015. ACM.

[50] Pablo Estefo, Miguel Campusano, Luc Fabresse, Johan Fabry, Jannik Laval and Noury Bouraqadi. *Towards Live Programming in ROS with PhaROS and LRP*. CoRR, vol. abs/1412.4629, 2014.

[51] Xuan Sang Le, Loïc Lagadec, Noury Bouraqadi, Luc Fabresse, and Jannik Laval. *From Smalltalk to Silicon : Towards a methodology to turn Smalltalk code into FPGA*. In IWST 14, Cambridge, United Kingdom, August 2014.

[52] Jannik Laval, Luc Fabresse and Noury Bouraqadi. *Towards Test-Driven Development for Mobile Robots*. In Proceedings of the eighth full-day Workshop on Software Development and Integration in Robotics (SDIR VIII), in conjunction with the IEEE International Conference on Robotics and Automation (ICRA), pages 12–13, Karlsruhe, Germany, May 2013.

[53] Jannik Laval, Stéphane Ducasse, Nicolas Anquetil. OZONE : Package Layered Structure Identification in presence of Cycles. In *Proceedings of the 9th BElgian-NEtherlands software eVOLution seminar (BENEVOL 2010)*, 2010.

[54] Alexandre Bergel, Mariano Abel Coca, Gabriela Arevalo, Dale Henrichs, Jannik Laval. Memory Profiling Blueprint. In *Proceedings of the 4th Argentinian Smalltalk Conference 2010 (Smalltalks10)*, 2010

[55] Tristan Bourgois, Jannik Laval, Stéphane Ducasse and Damien Pollet. *BLOC : a Trait-Based Collections Library – a Preliminary Experience Report*. In International Workshop on Smalltalk Technologies, Barcelona, Spain, September 2010.

[56] Jannik Laval, Simon Denier, Stéphane Ducasse. Identifying cycle causes with CycleTable. In *FAMOOSr 2009 : 3rd Workshop on FAMIX and MOOSE in Software Reengineering, conjointement à WCRE 2009*, 2009.

[57] Jannik Laval, Simon Denier, Stéphane Ducasse and Andy Kellens. *Supporting Incremental Change in Large System Models*. In IWST, Brest, France, August 2009.

[58] Jannik Laval, Alexandre Bergel, Stéphane Ducasse. Assessing the Quality of your Software with MoQam. In *FAMOOSr, 2nd Workshop on FAMIX and Moose in Reengineering*, conjointement à WCRE 2008, 2008.

[59] Alexandre Bergel, Stéphane Ducasse, Jannik Laval, Romain Peirs. Enhanced Dependency Structure Matrix for Moose. In *FAMOOSr, 2nd Workshop on FAMIX and Moose in Reengineering*, conjointement à WCRE 2008, 2008.

Conférences nationales avec comité de lecture

[60] Nawel Amokrane, Jannik Laval, Philippe Lanco, Mustapha Derras and Nejib Moalla. Data Interoperability Assessment Towards the Resilience of Data Exchange Systems In *CAL 2019, Conférence Francophone sur les Architectures Logicielles*, Hammamet, Tunisie, 29-30 octobre 2019.

[61] Jannik Laval and Mathieu Vermeulen. LES A.L.P.E.S. : APPROCHES AGILES POUR L'ENSEIGNEMENT SUPERIEUR. In *Inforsid 2016 - INFormatique des ORganisation et Systèmes d'Information et de Décision*, Grenoble, France, May 2016.

[62] Zhi Yan, Luc Fabresse, Jannik Laval, Noury Bouraqadi. *Métriques pour le benchmarking de performance d'exploration multi-robots : Environnements socio-techniques*. In 23es Journées Francophones sur les Systèmes Multi-Agents (JFSMA'15), Rennes, France. pp.9-18, Jun 2015.

[63] Zhi Yan, Luc Fabresse, Jannik Laval, and Noury Bouraqadi. *Simulating multi-robot exploration using ros and morse.*. In Proceedings of 9th National Conference on "Control Architecture of Robots" Control Architectures of Robots (CAR'14), Paris, France, June 2014.

[64] Luc Fabresse, Jannik Laval and Noury Bouraqadi. *Towards Quality Assurance Tests for Mobile Robots*. In Proceedings of 8th National Conference on "Control Architecture of Robots" Control Architectures of Robots (CAR'13), 2013.

[65] Jannik Laval, Alexandre Bergel, Stéphane Ducasse and Romain Piers. *Matrice de dependances enrichie*. In LMO, Nancy, France, March 2009.

Communication GDR

[66] Xuan Sang Le, Luc Fabresse, Jannik Laval, Jean-Christophe Le Lann, Loïc Lagadec and Noury Bouraqadi. *Dynamic Distributed Programming on Reconfigurable IP-Based Smart Sensor Networks*. Poster presented at 11eme Colloque du GDR SoC-SiP, France, 2016.

[67] Madeleine El-Zaher, Taha Elhariri, Baudouin Dafflon et Jannik Laval. *Réseaux CPS pour des Systèmes auto-adaptables dédiés à la production.* Poster presented at 3ème Journée sur les Architectures Logicielles pour la Robotique Autonome, les Systèmes Cyber-Physiques et les Systèmes Auto-Adaptables (GdR GPL et Robotique), 2015.

[68] Pablo Estefó, Luc Fabresse, Jannik Laval, and Noury Bouraqadi. *Développement interactif d'applications robotiques avec ROS.* Journée sur les Architectures Logicielles pour la Robotique Autonome, les Systèmes Cyber-Physiques et les Systèmes Auto-Adaptables (GdR GPL et Robotique), 2014.

[69] Jannik Laval, Simon Denier, Stéphane Ducasse, Jean-Rémy Falleri. *Supporting simultaneous versions for software evolution assessment.* Journées du GDR-GPL, 2012.

[70] Jannik Laval, Usman Bhatti, Nicolas Anquetil, Stéphane Ducasse. *Software Maintenance Analysis and Understanding of the Software Structure.* Journées du GDR-GPL, 2011.

[71] Jannik Laval, Alexandre Bergel, Stéphane Ducasse, Romain Piers. *Matrice de dépendances enrichie.* Journées du GDR-GPL, 2010.

[72] Nicolas Anquetil, Simon Denier, Stéphane Ducasse, Jannik Laval, Damien Pollet, Roland Ducournau, Rodolphe Giroudeau, Marianne Huchard, Jean-Claude Konig, Abdelhak-Jamel Seriai. *Software (re)modularization : Fight against the structure erosion and migration preparation.* Journées du GDR-GPL, 2010.

Diffusion de l'information scientifique

[73] Jannik Laval, Stéphane Ducasse. Package et gestion de versions en Pharo. GNU/Linux Magazine, novembre 2010.

[74] Jannik Laval, Stéphane Ducasse. Pharo : un nouveau Smalltalk open source. GNU/Linux Magazine, septembre 2010.

[75] Jannik Laval, Stéphane Ducasse. Optimisation d'applications en Pharo. GNU/Linux Magazine, janvier 2010.

Rapports de recherche

[76] Jannik Laval. JetStorm - A communication protocol between Pharo and Lego Mindstorms. 2014.

[77] Noury Bouraqadi, Luc Fabresse, Jannik Laval, Santiago Bragagnolo. A Robotic Infrastructure for Map Construction and Navigation - Deliverable 2 of the RoboShop Project. 2013.

[78] Noury Bouraqadi, Luc Fabresse, Jannik Laval, Santiago Bragagnolo. PhaROS : Concepts and Architecture - Deliverable 1 of the RoboShop Project. 2013.

[79] Stéphane Ducasse, Nicolas Anquetil, Usman Bhatti, Andre Cavalcante Hora, Jannik Laval, Tudor Girba. MSE and FAMIX 3.0 : an Interexchange Format and Source Code Model Family. 2011.

[80] Jannik Laval, Simon Denier, Stéphane Ducasse. Cycles Assessment with CycleTable. 2011.

[81] Simon Denier, Jannik Laval, Stéphane Ducasse, Fabrice Bellingard. Technical and Economical Model (Squale Deliverable 2.2). Disponible sur www.squale.org, 2010.

[82] Simon Denier, Jannik Laval, Stéphane Ducasse, Fabrice Bellingard. Technical and Economical Model (Squale Deliverable 2.1). Disponible sur www.squale.org, 2010.

[83] Françoise Balmas, Fabrice Bellingard, Simon Denier, Stéphane Ducasse, Bertrand Franchet, Jannik Laval, Karine Mordal-Manet and Philippe Vaillergues. *Practices in the Squale Quality Model (Squale Deliverable 1.3)*. Research report. Disponible sur www.squale.org, March 2010.

[84] Stéphane Ducasse, Simon Denier, Françoise Balmas, Alexandre Bergel, Jannik Laval, Karine Mordal-Manet, Fabrice Bellingard. *Visualization of Practices and Metrics (Squale Deliverable 1.2)*. Research report. Disponible sur www.squale.org, 2010.

[85] Françoise Balmas, Alexandre Bergel, Simon Denier, Stéphane Ducasse, Jannik Laval, Karine Mordal-Manet, Hani Abdeen and Fabrice Bellingard. *Software metric for Java and C++ practices (Workpackage 1.1)*. Research report. Disponible sur www.squale.org, March 2010.

[86] Alexandre Bergel, Simon Denier, Stéphane Ducasse, Jannik Laval, Fabrice Bellingard, Philippe Vaillergues, Françoise Balmas, Karine Mordal-Manet. SQUALE – Software QUALity Enhancement. European Projects Track, in 13th European Conference on Software Maintenance and Reengineering (CSMR), 2009.

[87] Karine Mordal-Manet, Françoise Balmas, Simon Denier, Stéphane Ducasse, Harald Wertz, Jannik Laval, Fabrice Bellingard, Philippe Vaillergues. The Squale Model – A Practice-based Industrial Quality Model. Disponible sur www.squale.org, 2009.

Articles en soumission

[88] Editeur invité : Anne Etien, Jannik Laval and Loic Lagadec. *Advances in Dynamic Languages*. Science of Computer Programming. Advances in Dynamic Languages, 3rd edition.

[89] Abdelhadi Belfadel, Emna Amdouni, Jannik Laval, Chantal Cherifi and Nejib Moalla. *Towards Software Reuse Through an Enterprise Architecture-based Software Capability Profile*. Enterprise Information Systems, 2020.

[90] Abdelhadi Belfadel, Jannik Laval, Chantal Cherifi and Nejib Moalla. *Enterprise Architecture-based Software Capability Profile*. Journal : Science of Computer Programming, 2020.

[91] Abdelhadi Belfadel, Jannik Laval, Chantal Cherifi and Nejib Moalla. *Requirements Engineering and Enterprise Architecture-based Software Discovery and Reuse*. Journal of Universal Computer Science (J.UCS).

Bibliographie

[Al-Dhuraibi 2017] Yahya Al-Dhuraibi, Fawaz Paraiso, Nabil Djarallah and Philippe Merle. *Elasticity in cloud computing : state of the art and research challenges*. IEEE Transactions on Services Computing, vol. 11, no. 2, pages 430–447, 2017.

[Al-Mamary 2014] Yaser Hasan Al-Mamary, Alina Shamsuddin, Abdul Hamid and Nor Aziati. *The role of different types of information systems in business organizations : A review*. International Journal of Research (IJR), vol. 1, no. 7, 2014.

[Al-Masri 2007] Eyhab Al-Masri and Qusay H Mahmoud. *Qos-based discovery and ranking of web services*. In 2007 16th international conference on computer communications and networks, pages 529–534. IEEE, 2007.

[Alebrahim 2017] Azadeh Alebrahim and Maritta Heisel. Bridging the gap between requirements engineering and software architecture. Springer, 2017.

[Atamuradov 2017] Vepa Atamuradov, Kamal Medjaher, Pierre Dersin, Benjamin Lamoureux and Noureddine Zerhouni. *Prognostics and health management for maintenance practitioners-review, implementation and tools evaluation*. International Journal of Prognostics and Health Management, vol. 8, no. 060, pages 1–31, 2017.

[Avritzer 2018] Alberto Avritzer, Vincenzo Ferme, Andrea Janes, Barbara Russo, Henning Schulz and André van Hoorn. *A Quantitative Approach for the Assessment of Microservice Architecture Deployment Alternatives by Automated Performance Testing*. In European Conference on Software Architecture, pages 159–174. Springer, 2018.

[Baskerville 2011] R. Baskerville, J. Pries-Heje and S. Madsen. *Post-Agility : What Follows a Decade of Agility ?* Information and Software Technology, vol. 53, no. 5, pages 543–555, 2011.

[Beck 2003] Kent Beck. Test-driven development : by example. Addison-Wesley Professional, 2003.

[Belfadel 2019] Abdelhadi Belfadel. *Enterprise Architecture Capability Profile for Software Reuse.* PhD thesis, University of Lyon, France, 2019.

[Biggs 2010] G. Biggs. *Applying regression testing to software for robot hardware interaction*. In Robotics and Automation (ICRA), 2010 IEEE International Conference on, pages 4621–4626, 2010.

[Bouraqadi 2009] Noury Bouraqadi, Serge Stinckwich, Victor Moraru, Arnaud Doniec*et al. Making networked robots connectivity-aware*. In 2009 IEEE International conference on robotics and automation, pages 3502–3507. IEEE, 2009.

[Brand 2019] T. Brand and H. Giese. *Generic Adaptive Monitoring Based on Executed Architecture Runtime Model Queries and Events*. In 2019 IEEE 13th International Conference on Self-Adaptive and Self-Organizing Systems (SASO), pages 17–22, June 2019.

[Bridges 2013] Robert A Bridges, Corinne L Jones, Michael D Iannacone, Kelly M Testa and John R Goodall. *Automatic labeling for entity extraction in cyber security*. arXiv preprint arXiv :1308.4941, 2013.

[Buyya 2016] Rajkumar Buyya and Amir Vahid Dastjerdi. Internet of things : Principles and paradigms. Elsevier, 2016.

[Cadavid 2020a] Héctor Cadavid, Vasilios Andrikopoulos and Paris Avgeriou. *Architecting systems of systems : A tertiary study*. Information and Software Technology, vol. 118, page 106202, 2020.

[Cadavid 2020b] Héctor Cadavid, Vasilios Andrikopoulos and Paris Avgeriou. *Architecting systems of systems : A tertiary study*. Information and Software Technology, vol. 118, page 106202, 2020.

[Ceccarelli 2016] Andrea Ceccarelli, Andrea Bondavalli, Bernhard Froemel, Oliver Hoeftberger and Hermann Kopetz. Basic concepts on systems of systems, pages 1–39. Springer International Publishing, Cham, 2016.

[Ceusters 2012] Werner Ceusters. *An information artifact ontology perspective on data collections and associated representational artifacts.* In MIE, pages 68–72, 2012.

[Chaabane 2019] Mariam Chaabane, Ismael Bouassida Rodriguez, Ricardo Colomo-Palacios, Walid Gaaloul and Mohamed Jmaiel. *A modeling approach for Systems-of-Systems by adapting ISO/IEC/IEEE 42010 Standard evaluated by Goal-Question-Metric*. Science of Computer Programming, vol. 184, page 102305, 2019.

[Chen 2006] David Chen, Nicolas Daclin*et al. Framework for enterprise interoperability*. In Proc. of IFAC Workshop EI2N, pages 77–88, 2006.

[Chen 2008] David Chen, Guy Doumeingts and François Vernadat. *Architectures for enterprise integration and interoperability : Past, present and future*. Computers in industry, vol. 59, no. 7, pages 647–659, 2008.

[Chen 2011] Ian Yen-Hung Chen, Bruce A. MacDonald and Burkhard C. Wünsche. *A Flexible Mixed Reality Simulation Framework for Software Development in Robotics*. Journal of Software Engineering for Robotics, vol. 2, no. 1, pages 40–54, September 2011.

[Chiu 2016] Jason PC Chiu and Eric Nichols. *Named entity recognition with bidirectional LSTM-CNNs*. Transactions of the Association for Computational Linguistics, vol. 4, pages 357–370, 2016.

[Chung 2007] Yun Koo Chung and Sun-Myung Hwang. *Software testing for intelligent robots*. In International Conference on Control, Automation and Systems 2007, pages 2344–2349. IEEE, 2007.

[Clark 1999] Thea Clark and Richard Jones. *Organisational interoperability maturity model for C2*. In Proceedings of the 1999 Command and Control Research and Technology Symposium. Citeseer, 1999.

[Cojocaru 2019] M. Cojocaru, A. Uta and A. Oprescu. *Attributes Assessing the Quality of Microservices Automatically Decomposed from Monolithic Applications*. In 2019 18th International Symposium on Parallel and Distributed Computing (ISPDC), pages 84–93, June 2019.

[Conboy 2004] K. Conboy and B. Fitzgerald. *Toward a Conceptual Framework of Agile Methods : A Study of Agility in Different Disciplines*. In 2004 ACM workshop on Interdisciplinary software engineering research, Newport Beach, CA, USA, pages 37–44, 2004.

[Couceiro 2014] Micael S Couceiro, Patricia A Vargas, Rui P Rocha and Nuno MF Ferreira. *Benchmark of swarm robotics distributed techniques in a search task*. Robotics and Autonomous Systems, vol. 62, no. 2, pages 200–213, 2014.

[CYBER 2019] Pôle d'excellence CYBER. *Référentiel cyber v5.0*. Rapport technique, Pôle d'excellence CYBER, nov. 2019.

[Cynet 2020] Cynet. *State of Breach Protection 2020*. Rapport technique, Cynet, 2020.

[Da Xu 2011] Li Da Xu. *Enterprise systems : state-of-the-art and future trends*. IEEE Transactions on Industrial Informatics, vol. 7, no. 4, pages 630–640, 2011.

[Daclin 2008] Nicolas Daclin, David Chen and Bruno Vallespir. *Methodology for enterprise interoperability*. IFAC Proceedings Volumes, vol. 41, no. 2, pages 12873–12878, 2008.

[Dadeau-FEMTO-ST 2014] Frédéric Dadeau-FEMTO-ST and Hélène Waeselynck-LAAS. *Les défis du Test Logiciel-Bilan et Perspectives*. Groupement De Recherche CNRS du Génie de la Programmation et du Logiciel, page 177, 2014.

[Dahmann 2014] Judith Dahmann. *1.4. 3 system of systems pain points*. In INCOSE International Symposium, volume 24, pages 108–121. Wiley Online Library, 2014.

[Dahmann 2015] J. Dahmann. *The state of systems of systems engineering knowledge sources*. In 2015 10th System of Systems Engineering Conference (SoSE), pages 475–479, May 2015.

[Dávid 2018] István Dávid, István Ráth and Dániel Varró. *Foundations for streaming model transformations by complex event processing*. Software & Systems Modeling, vol. 17, no. 1, pages 135–162, 2018.

[Davis 1995] Alan Mark Davis. 201 principles of software development. McGraw-Hill, 1995.

[de Soria 2009] Iker Martinez de Soria, Juncal Alonso, Leire Orue-Echevarria and Mikel Vergara. *Developing an enterprise collaboration maturity model : research challenges and future directions*. In Technology Management Conference (ICE), 2009 IEEE International, pages 1–8. IEEE, 2009.

[Demeyer 2001] Serge Demeyer, Sander Tichelaar and Stéphane Ducasse. *FAMIX 2.1 — The FAMOOS Information Exchange Model*. Rapport technique, University of Bern, 2001.

[Di Francesco 2017] Paolo Di Francesco, Ivano Malavolta and Patricia Lago. *Research on architecting microservices : Trends, focus, and potential for industrial adoption*. In 2017 IEEE International Conference on Software Architecture (ICSA), pages 21–30. IEEE, 2017.

[Dong 2009] Xin Luna Dong and Felix Naumann. *Data fusion : resolving data conflicts for integration*. Proceedings of the VLDB Endowment, vol. 2, no. 2, pages 1654–1655, 2009.

[Dossot 2014] David Dossot. Rabbitmq essentials. Packt Publishing Ltd, 2014.

[Ducasse 2009] Stéphane Ducasse, Tudor Gîrba, Adrian Kuhn and Lukas Renggli. *Meta-Environment and Executable Meta-Language using Smalltalk : an Experience Report*. Journal of Software and Systems Modeling (SOSYM), vol. 8, no. 1, pages 5–19, February 2009.

[Dynatrace 2020] Dynatrace. *CIOs are turning to AI to bridge the gap between constrained resources & cloud complexity*. Rapport technique, Dynatrace, 2020.

[Echeverria 2012] Gilberto Echeverria, Séverin Lemaignan, Arnaud Degroote, Simon Lacroix, Michael Karg, Pierrick Koch, Charles Lesire and Serge Stinckwich. *Simulating complex robotic scenarios with MORSE*. In International Conference on Simulation, Modeling, and Programming for Autonomous Robots, pages 197–208. Springer, 2012.

[Erlikh 2000] Len Erlikh. *Leveraging Legacy System Dollars for E-Business*. IT Professional, vol. 2, no. 3, pages 17–23, 2000.

[Faigl 2015] Jan Faigl and Miroslav Kulich. *On benchmarking of frontier-based multi-robot exploration strategies*. In 2015 european conference on mobile robots (ECMR), pages 1–8. IEEE, 2015.

[Finkel 2005] Jenny Rose Finkel, Trond Grenager and Christopher Manning. *Incorporating non-local information into information extraction systems by gibbs sampling*. In Proceedings of the 43rd annual meeting on association for computational linguistics, pages 363–370. Association for Computational Linguistics, 2005.

[Ford 2007] Thomas Ford, John Colombi, Scott Graham and David Jacques. *The interoperability score*. Rapport technique, AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH, 2007.

[Freese 2010] Marc Freese, Surya Singh, Fumio Ozaki and Nobuto Matsuhira. *Virtual robot experimentation platform v-rep : a versatile 3d robot simulator*. In International Conference on Simulation, Modeling, and Programming for Autonomous Robots, pages 51–62. Springer, 2010.

[Fülöp 2010] Lajos Jenő Fülöp, Gabriella Tóth, Róbert Rácz, János Pánczél, Tamás Gergely, Arpád Beszédes and Lóránt Farkas. *Survey on complex event processing and predictive analytics*. In Proceedings of the Fifth Balkan Conference in Informatics, pages 26–31. Citeseer, 2010.

[Garg 2013] Nishant Garg. Apache kafka. Packt Publishing Ltd, 2013.

[Gascon-Samson 2015] Julien Gascon-Samson, Franz-Philippe Garcia, Bettina Kemme and Jörg Kienzle. *Dynamoth : A scalable pub/sub middleware for latency-constrained applications in the cloud*. In 2015 IEEE 35th International Conference on Distributed Computing Systems, pages 486–496. IEEE, 2015.

[Geraci 1991] Anne Geraci, Freny Katki, Louise McMonegal, Bennett Meyer, John Lane, Paul Wilson, Jane Radatz, Mary Yee, Hugh Porteous and Fredrick Springsteel. Ieee standard computer dictionary : Compilation of ieee standard computer glossaries. IEEE Press, 1991.

[Gerber 2010] Aurona Gerber, Paula Kotzé and Alta Van der Merwe. *Towards the formalisation of the TOGAF Content Metamodel using ontologies*. In 12th International Conference on Enterprise Information Systems (ICEIS), 2010.

[Gerkey 2003] Brian Gerkey, Richard T Vaughan and Andrew Howard. *The player/stage project : Tools for multi-robot and distributed sensor systems*. In Proceedings of the 11th international conference on advanced robotics, volume 1, pages 317–323, 2003.

[Gers 1999] Felix A Gers, Jürgen Schmidhuber and Fred Cummins. *Learning to forget : Continual prediction with LSTM*. In 9th International Conference on Artificial Neural Networks : ICANN '99, pages 850–855. IET, 1999.

[Group 1998] C4ISR Architecture Working Group*et al.* *Levels of information systems interoperability (LISI)*. US DoD, 1998.

[Guariniello 2019] C. Guariniello, A. K. Raz, L. Mockus and D. DeLaurentis. *Experiences in Evolving System-of-Systems Engineering Methodology to Address Pain Points*. In 2019 14th Annual Conference System of Systems Engineering (SoSE), pages 189–194, May 2019.

[Guédria 2009] Wided Guédria, David Chen and Yannick Naudet. *A maturity model for enterprise interoperability*. In OTM Confederated International Conferences" On the Move to Meaningful Internet Systems", pages 216–225. Springer, 2009.

[Harman 2007] Mark Harman. *The current state and future of search based software engineering*. In 2007 Future of Software Engineering, pages 342–357. IEEE Computer Society, 2007.

[Hauge 2010] Øyvind Hauge, Daniela Soares Cruzes, Reidar Conradi, Ketil Sandanger Velle and Tron André Skarpenes. *Risks and risk mitigation in open source software adoption : bridging the gap between literature and practice*. In IFIP International Conference on Open Source Systems, pages 105–118. Springer, 2010.

[Hohpe 2004] Gregor Hohpe and Bobby Woolf. Enterprise integration patterns : Designing, building, and deploying messaging solutions. Addison-Wesley Professional, 2004.

[Huang 2015] Zhiheng Huang, Wei Xu and Kai Yu. *Bidirectional LSTM-CRF models for sequence tagging*. arXiv preprint arXiv :1508.01991, 2015.

[Hussain 2017] Sayed Muqtada Hussain, Shahid Nazir Bhatti and M Faizan Ur Rasool. *Legacy system and ways of its evolution*. In 2017 International Conference on Communication Technologies (ComTech), pages 56–59. IEEE, 2017.

[Iacob 2009] Maria-Eugenia Iacob and Henk Jonkers. *A model-driven perspective on the rule-based specification and analysis of service-based applications*. Enterprise information systems, vol. 3, no. 3, pages 279–298, 2009.

[IEEE 2004] IEEE. *IEEE Standard for VHDL Register Transfer Level (RTL) Synthesis*. IEEE Std 1076.6-2004 (Revision of IEEE Std 1076.6-1999), pages 1–118, Oct 2004.

[Int'l 2015] Standish Group Int'l. *Chaos Report*. Tech. report, Standish Group Int'l, 2015.

[ISO Central Secretary 2011] ISO Central Secretary. *Ingénierie des systèmes et du logiciel — Exigences de qualité et évaluation des systèmes et du logiciel (SQuaRE) — Modèles de qualité du système et du logiciel*. Standard ISO/IEC 25010 :2011, International Organization for Standardization, Geneva, CH, 2011.

[ISO Central Secretary 2015] ISO Central Secretary. *Ingénierie des systèmes et du logiciel — Processus du cycle de vie du système*. Standard ISO/IEC/IEEE 15288 :2015, International Organization for Standardization, Geneva, CH, May 2015.

[ISO Central Secretary 2018] ISO Central Secretary. *Industrial automation systems and integration — Manufacturing software capability profiling for interoperability — Part 6 : Interface services and protocols for matching profiles based on multiple capability class structures*. Standard ISO 16100-6 :2018, International Organization for Standardization, Geneva, CH, 2018.

[Jinhui 2009] Zhu Jinhui, Zheng Qilun, Liang Yingju, Liang Mingjie and Min Huaqing. *A component-based hybrid testbed for multi-mobile robots*. In 2009 International Conference on Information and Automation, pages 905–909. IEEE, 2009.

[Joshi 2013] Arnav Joshi, Ravendar Lal, Tim Finin and Anupam Joshi. *Extracting cybersecurity related linked data from text*. In 2013 IEEE Seventh International Conference on Semantic Computing, pages 252–259. IEEE, 2013.

[Kakarontzas 2010] George Kakarontzas, Panagiotis Katsaros and Ioannis Stamelos. *Component Certification as a Prerequisite forWidespread OSS Reuse*. Electronic Communications of the EASST, vol. 33, 2010.

[Kamble 2019] Sachin S Kamble, Angappa Gunasekaran, Harsh Parekh and Sudhanshu Joshi. *Modeling the internet of things adoption barriers in food retail supply chains*. Journal of Retailing and Consumer Services, vol. 48, pages 154–168, 2019.

[Kingston 2005] Gina Kingston, Suzanne Fewell and Warren Richer. *An organisational interoperability agility model*. Rapport technique, Defence Science and Technology Organisation Canberra (Australia), 2005.

[Koenig 2004] Nathan Koenig and Andrew Howard. *Design and use paradigms for gazebo, an open-source multi-robot simulator*. In 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566), volume 3, pages 2149–2154. IEEE, 2004.

[Kudelski 2013] Michal Kudelski, Luca M Gambardella and Gianni A Di Caro. *RoboNetSim : An integrated framework for multi-robot and network simulation*. Robotics and Autonomous Systems, vol. 61, no. 5, pages 483–496, 2013.

[Laboratoire LIUPPA 2014] Université de Pau et Pays de l'Adour Laboratoire LIUPPA and Université de Bretagne-Sud Archware Laboratoire IRISA. *Défis dans l'ingénierie logiciel des Systèmes de Systèmes*. Rapport technique, GDR GPL, 2014.

[Lample 2016] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami and Chris Dyer. *Neural architectures for named entity recognition*. arXiv preprint arXiv :1603.01360, 2016.

[Lankhorst 2009] Marc Lankhorst*et al.* Enterprise architecture at work, volume 352. Springer, 2009.

[Leal 2018] António Leal, Fátima Oliveira and Purificação Silvano. *Path scales : Directed-motion verbs, prepositions and telicity in European Portuguese*. Studies in Language Companion Series, vol. 197, no. 1, pages 335–355, 2018.

[Lehman 1996] Manny Lehman. *Laws of Software Evolution Revisited*. In European Workshop on Software Process Technology, pages 108–124, Berlin, 1996. Springer.

[Li 2015] Shancang Li, Li Da Xu and Shanshan Zhao. *The internet of things : a survey*. Information Systems Frontiers, vol. 17, no. 2, pages 243–259, 2015.

[Liao 2016] Xiaojing Liao, Kan Yuan, XiaoFeng Wang, Zhou Li, Luyi Xing and Raheem Beyah. *Acing the ioc game : Toward automatic discovery and analysis of open-source cyber threat intelligence*. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, pages 755–766, 2016.

[Lim 2010] Jae-Hee Lim, Suk-Hoon Song, Jung-Rye Son, Tae-Yong Kuc, Hong-Seong Park and Hong-Seak Kim. *An Automated Test Method for Robot Platform and Its Components*. International Journal of Software Engineering and Its Applications, vol. 4, no. 3, pages 9–18, July 2010.

[Lou 1995] H-L Lou. *Implementing the Viterbi algorithm*. IEEE Signal processing magazine, vol. 12, no. 5, pages 42–52, 1995.

[Lowry 2016] Paul Benjamin Lowry and David Wilson. *Creating agile organizations through IT : The influence of internal IT service perceptions on IT service quality and IT agility*. The Journal of Strategic Information Systems, vol. 25, no. 3, pages 211–226, 2016.

[Ma 2016] Xuezhe Ma and Eduard Hovy. *End-to-end sequence labeling via bi-directional lstm-cnns-crf*. arXiv preprint arXiv :1603.01354, 2016.

[Maggi 2018] Federico Maggi, Rainer Vosseler and Davide Quarta. *The fragility of industrial IoT's data backbone*. Rapport technique, Trend Micro Research, 2018.

[Maier 1998] Mark W. Maier. *Architecting principles for systems-of-systems*. Systems Engineering, vol. 1, no. 4, pages 267–284, 1998.

[Mallek 2010] Sihem Mallek, Nicolas Daclin and Vincent Chapurlat. *Towards a conceptualisation of interoperability requirements*. In Enterprise Interoperability IV, pages 439–448. Springer, 2010.

[Mallek 2012] Sihem Mallek, Nicolas Daclin and Vincent Chapurlat. *The application of interoperability requirement specification and verification to collaborative processes in industry*. Computers in industry, vol. 63, no. 7, pages 643–658, 2012.

[Martin 2004] David Martin, Massimo Paolucci, Sheila McIlraith, Mark Burstein, Drew McDermott, Deborah McGuinness, Bijan Parsia, Terry Payne, Marta Sabou, Monika Solanki*et al.* *Bringing semantics to web services : The OWL-S approach*. In International Workshop on Semantic Web Services and Web Process Composition, pages 26–42. Springer, 2004.

[Mateos 2004] Montserrat Mateos, Juan Carlos Álvarez and María N Moreno. *PROCESS FOR REENGINEERING OF LEGACY SYSTEMS*. e-Society 2004, page 1033, 2004.

[McAteer 2017] Ian Noel McAteer, Muhammad Imran Malik, Zubair Baig and Peter Hannay. *Security vulnerabilities and cyber threat analysis of the AMQP protocol for the internet of things*. The Proceedings of 15th Australian Information Security Management Conference, pages 70–80, December 2017.

[Michael 2008] Nathan Michael, Jonathan Fink and Vijay Kumar. *Experimental testbed for large multirobot teams*. IEEE robotics & automation magazine, vol. 15, no. 1, pages 53–61, 2008.

[Michel 2004] O Michel. *WebotsTM : Professional mobile robot simulation*. arXiv preprint cs, vol. 412052, 2004.

[Miwa 2016] Makoto Miwa and Mohit Bansal. *End-to-end relation extraction using lstms on sequences and tree structures*. arXiv preprint arXiv :1601.00770, 2016.

[Moalla 2012] Nejib Moalla, Hervé Panetto, Xavier Boucher, N. Moalla, H. Panetto, X. Boucher and Moalla al. *Interopérabilité et partage de connaissances*. Ingénierie des Systèmes d'Information, vol. 17, pages 7–16, 08 2012.

[Mohsin 2019] Ahmad Mohsin, Naeem Khalid Janjua, Syed MS Islam and Valdemar Vicente Graciano Neto. *Modeling Approaches for System-of-Systems Dynamic Architecture : Overview, Taxonomy and Future Prospects*. architecture, vol. 13, page 15, 2019.

[Monteiro 2018] Davi Monteiro, Rômulo Gadelha, Paulo Henrique M Maia, Lincoln S Rocha and Nabor C Mendonça. *Beethoven : an event-driven lightweight platform for microservice orchestration*. In European Conference on Software Architecture, pages 191–199. Springer, 2018.

[Moody's 2019] Moody's. *Blockchain standardisation will amplify benefits for securitisations*. Rapport technique, Moody's, 2019.

[More 2012] Sumit More, Mary Matthews, Anupam Joshi and Tim Finin. *A knowledge-based approach to intrusion detection modeling*. In 2012 IEEE Symposium on Security and Privacy Workshops, pages 75–81. IEEE, 2012.

[Mulwad 2011] Varish Mulwad, Wenjia Li, Anupam Joshi, Tim Finin and Krishnamurthy Viswanathan. *Extracting information about security vulnerabilities from web text*. In 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology, volume 3, pages 257–260. IEEE, 2011.

[Natschläger 2011] Christine Natschläger. *Towards a BPMN 2.0 ontology*. In International Workshop on Business Process Modeling Notation, pages 1–15. Springer, 2011.

[O'Brien 2006] James A O'Brien, George M Marakas*et al.* Management information systems, volume 6. McGraw-Hill Irwin, 2006.

[Ostadzadeh 2015] S Shervin Ostadzadeh, Fereidoon Shams and Kambiz Badie. *An architectural model framework to improve digital ecosystems interoperability*. In New trends in networking, computing, e-learning, systems sciences, and engineering, pages 513–520. Springer, 2015.

[Ouni 2015] Ali Ouni, Marouane Kessentini, Katsuro Inoue and Mel O Cinnéide. *Search-based web service antipatterns detection*. IEEE Transactions on Services Computing, vol. 10, no. 4, pages 603–617, 2015.

[Panetto 2016] Hervé Panetto, Milan Zdravkovic, Ricardo Jardim-Goncalves, David Romero, Joe Cecil and István Mezgár. *New perspectives for the future interoperable enterprise systems*. Computers in Industry, vol. 79, pages 47–63, 2016.

[Paschali 2017] Maria-Eleni Paschali, Apostolos Ampatzoglou, Stamatia Bibi, Alexander Chatzigeorgiou and Ioannis Stamelos. *Reusability of open source software across domains : A case study*. Journal of Systems and Software, vol. 134, pages 211–227, 2017.

[Patterns 2009] Microsoft Patterns. Microsoft application architecture guide. Microsoft Press, 2009.

[Petters 2008] S. Petters, D. Thomas, M. Friedmann and O. Von Stryk. *Multilevel testing of control software for teams of autonomous mobile robots*. Simulation, Modeling, and Programming for Autonomous Robots, pages 183–194, 2008.

[Piskorski 2013] Jakub Piskorski and Roman Yangarber. *Information extraction : Past, present and future*. In Multi-source, multilingual information extraction and summarization, pages 23–49. Springer, 2013.

[Plakidas 2018] Konstantinos Plakidas, Daniel Schall and Uwe Zdun. *Software Migration and Architecture Evolution with Industrial Platforms : A Multi-case Study*. In Carlos E. Cuesta, David Garlan and Jennifer Pérez, editeurs, Software Architecture, pages 336–343, Cham, 2018. Springer International Publishing.

[Porru 2017] Simone Porru, Andrea Pinna, Michele Marchesi and Roberto Tonelli. *Blockchain-oriented software engineering : challenges and new directions*. In 2017 IEEE/ACM 39th International Conference on Software Engineering Companion (ICSE-C), pages 169–171. IEEE, 2017.

[Purushothaman 2005] R. Purushothaman and D.E. Perry. *Toward understanding the rhetoric of small source code changes*. Software Engineering, IEEE Transactions on, vol. 31, no. 6, pages 511 – 526, June 2005.

[Quigley 2009] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler and Andrew Y Ng. *ROS : an open-source Robot Operating System*. In ICRA workshop on open source software, volume 3, page 5. Kobe, Japan, 2009.

[Raman 2019] Ramakrishnan Raman and Meenakshi D'Souza. *Decision learning framework for architecture design decisions of complex systems and system-of-systems*. Systems Engineering, vol. 22, no. 6, pages 538–560, 2019.

[Rasouli 2017] M Rasouli, H Ghazanfari and Rik Eshuis. *A process aware information system to support agility in relief operations*. In 13th International Conference on Industrial Engineering (IIEC 2017), 2017.

[Retaillé 2011] Jean-Philippe Retaillé. Refactoring des applications java/j2ee. Editions Eyrolles, 2011.

[Robertson 2012] Suzanne Robertson and James Robertson. Mastering the requirements process : Getting requirements right. Addison-wesley, 2012.

[Sang 2000] Erik F. Tjong Kim Sang and Sabine Buchholz. *Introduction to the CoNLL-2000 Shared Task : Chunking*. CoRR, vol. cs.CL/0009008, 2000.

[Seong 2012] Hong Seong and Jeong Seok. *SITAF : Simulation-based Interface Testing Automation Framework for Robot Software Component*. In Florian Kongoli, editeur, Automation. InTech, July 2012.

[Shackelford 2006] Russell Shackelford, Andrew McGettrick, Robert Sloan, Heikki Topi, Gordon Davies, Reza Kamali, James Cross, John Impagliazzo, Richard LeBlanc and Barry Lunt. *Computing curricula 2005 : The overview report*. In ACM SIGCSE Bulletin, volume 38, pages 456–457. ACM, 2006.

[Shelby 2014] Zach Shelby, Klaus Hartke and Carsten Bormann. *The Constrained Application Protocol (CoAP)*. RFC 7252, RFC Editor, June 2014.

[Sommerville 2011] Ian Sommerville. *Software engineering 9th Edition*. ISBN-10137035152, 2011.

[Son 2011] Jung-Rye Son, Tae-Yong Kuc, Jong-Koo Park and Hong-Seok Kim. *Simulation Based Functional and Performance Evaluation of Robot Components and Modules*. In Information Science and Applications (ICISA), 2011 International Conference on, pages 1–7. IEEE, April 2011.

[Standard 2014] OASIS Standard. *MQTT version 3.1. 1*. URL http ://docs. oasis-open. org/mqtt/mqtt/v3, vol. 1, 2014.

[Tarafdar 2017] Monideepa Tarafdar and Sufian Qrunfleh. *Agile supply chain strategy and supply chain performance : complementary roles of supply chain practices and information systems capability for agility*. International Journal of Production Research, vol. 55, no. 4, pages 925–938, 2017.

[The Open Group 2018] The Open Group. *The TOGAF® Standard, version 9.2*. Standard C182, The Open Group, 2018.

[Tolk 2003] Andreas Tolk and James A Muguira. *The levels of conceptual interoperability model*. In Proceedings of the 2003 fall simulation interoperability workshop, volume 7, pages 1–11. Citeseer, 2003.

[van der Aalst 2018] Wil MP van der Aalst, Martin Bichler and Armin Heinzl. *Robotic process automation*, 2018.

[Verdecho 2018] María-José Verdecho, Juan-José Alfaro-Saiz and Raúl Rodriguez. *Integrating business process interoperability into an inter-enterprise performance management system*. Proceedings of the 9th International Conference on Interoperability for Enterprise Systems and Applications (I-ESA) Berlin, Germany, 2018.

[Vinoski 2006] Steve Vinoski. *Advanced message queuing protocol*. IEEE Internet Computing, vol. 10, no. 6, pages 87–89, 2006.

[Vora 2003] Ojas Vora, Pankaj Vora and Urjaswala Vora. *Predictive Modeling for Infrastructure System Engineering*. Journal of Professional Issues in Engineering Education and Practice, 2003.

[Xu 2015] Yan Xu, Lili Mou, Ge Li, Yunchuan Chen, Hao Peng and Zhi Jin. *Classifying relations via long short term memory networks along shortest dependency paths*. In Proceedings of the 2015 conference on empirical methods in natural language processing, pages 1785–1794, 2015.

[Zheng 2017] Zibin Zheng, Shaoan Xie, Hongning Dai, Xiangping Chen and Huaimin Wang. *An overview of blockchain technology : Architecture, consensus, and future trends*. In 2017 IEEE international congress on big data (BigData congress), pages 557–564. IEEE, 2017.

[Zijden 2019] Stefan Van Der Zijden and Dennis Gaughan. *How to Prepare for the Future of Applications*. Tech. report, Gartner Research, March 2019.

Contribution à l'agilité des systèmes d'information

Ce mémoire d'Habilitation à Diriger des Recherches a pour objectif de présenter une synthèse de mes activités d'enseignement et de mes travaux de recherches depuis ma soutenance de thèse en 2011. Alors qu'en thèse, je me suis consacré à la maintenance logicielle, la suite de mes travaux a concerné les systèmes de systèmes, représentés par des robots mobiles ou encore des systèmes d'information distribués.

Mes réflexions m'ont permis de mener ma recherche selon un fil conducteur malgré deux changements d'équipe de recherche. J'ai d'abord intégré l'Unité de Recherche Informatique et Automatique de l'école des Mines de Douai en 2012. Dans cette équipe de recherche, j'ai travaillé sur le génie logiciel appliqué à la robotique mobile. J'ai proposé d'adapter des approches du génie logiciel à ce domaine. Ensuite, j'ai intégré le laboratoire DISP à Lyon en 2015. Dans ce laboratoire, je travaille sur l'agilité des systèmes d'information.

La problématique principale à laquelle je me suis consacré est l'agilité des systèmes d'information, tout au long de leur cycle de vie. La complexité des systèmes d'information aujourd'hui nécessite de les voir comme des systèmes de systèmes, et de les analyser d'une manière globale, tout en considérant l'incertitude de la stabilité des constituants et de leurs communications.

Ce mémoire est organisé en deux parties.

La première partie présente l'ensemble de mes activités et travaux d'enseignant-chercheur sous la forme d'un curriculum vitae détaillé. J'y détaille mes activités pédagogiques, de recherche, et d'animation scientifiques. Je liste également mes publications et mes projets de recherche industriels et universitaires ainsi que mes productions de logiciels.

La seconde partie constitue le cœur du document. J'y développe un bilan et les perspectives de mes activités de recherche selon trois principaux axes qui traitent de l'agilité du système d'information tout au long de son cycle de vie. J'ai choisi de décrire mes travaux de recherche, ainsi que les différents projets et encadrements suivant une articulation qui permet d'expliquer non seulement leur développement et leur évolution, mais également leur cohérence. J'ai donc choisi de les présenter selon un cycle de vie en trois axes :

— Axe 1 : Contribution à la construction d'architectures système d'information interopérables.

— Axe 2 : Contribution à la supervision des systèmes d'information distribués.

— Axe 3 : Contribution à l'outillage pour l'évolution des systèmes cyberphysiques.

Elle se termine par un bilan et une présentation de plusieurs perspectives pour la suite de ma carrière d'enseignant-chercheur.

Mots clés : Cycle de vie des systèmes de systèmes, système cyberphysique, architecture des systèmes d'information, modélisation des systèmes d'information, architecture orientée service, interopérabilité, agilité, monitoring, simulation, Vérification & Validation, analyse de vulnérabilité, LSTM.