

Université Paris 1 Panthéon – Sorbonne

HABILITATION A DIRIGER DES RECHERCHES

Spécialité « Informatique » 27<sup>ème</sup> section CNU

par

Saïd ASSAR

Contributions à la multidisciplinarité de  
l'informatique et des SI

Présentée et soutenue publiquement le **20 janvier 2017** devant le jury composé de :

Bénédicte LE GRAND	Université Paris 1 Sorbonne	<i>Garant / Rapporteur</i>
Oscar PASTOR	Univ. Politècnica de Valencia	<i>Rapporteur</i>
Yves PIGNEUR	HEC Lausanne	<i>Rapporteur</i>
David AVISON	ESSEC Business School	<i>Examineur</i>
Jean-Bernard BAILLON	Université Paris 1 Sorbonne	<i>Examineur</i>
Corine CAUVET	Université Aix - Marseille	<i>Examineur</i>
Colette ROLLAND	Université Paris 1 Sorbonne	<i>Examineur</i>
Camille SALINESI	Université Paris 1 Sorbonne	<i>Examineur</i>



# Table des matières

<b>I. Chapitre I : Introduction générale</b>	<b>9</b>
<b>I.1 Préambule</b>	<b>10</b>
<b>I.2 Analyse ontologique : les sujets d'étude en informatique</b>	<b>11</b>
<i>I.2.1 Trois perspectives d'étude</i>	<i>11</i>
<i>I.2.2 Logiciel versus Système d'Information</i>	<i>12</i>
<i>I.2.3 Génie logiciel versus Ingénierie des SI</i>	<i>13</i>
<i>I.2.4 Conception et développement versus Usage et impact du SI</i>	<i>13</i>
<b>I.3 Analyse méthodologique : les méthodes de recherche</b>	<b>14</b>
<b>I.4 Positionnement de mes travaux de recherche</b>	<b>15</b>
<i>I.4.1 Thème 1 : modèles et méthodes pour l'ingénierie des SI</i>	<i>15</i>
<i>I.4.2 Thème 2 : usages et adoption des artefacts informatiques et sociotechniques</i>	<i>18</i>
<b>I.5 Organisation du mémoire</b>	<b>19</b>
<b>II. Chapitre II : A propos des méthodes en science et en ingénierie</b>	<b>21</b>
<b>II.1 Préambule</b>	<b>22</b>
<b>II.2 Méthode(s) dans les sciences du formel</b>	<b>22</b>
<i>II.2.1 Méthodes du formel et étude du logiciel et du SI</i>	<i>23</i>
<b>II.3 Méthode(s) dans les sciences des faits</b>	<b>23</b>
<i>II.3.1 L'expérimentation et la méthode hypothético-déductive</i>	<i>24</i>
<i>II.3.2 Qualitatif versus quantitatif</i>	<i>24</i>
<i>II.3.3 Inductif versus déductif</i>	<i>25</i>
<i>II.3.4 Récapitulatif : la dualité du « fait » empirique</i>	<i>25</i>
<b>II.4 Méthode(s) de conception</b>	<b>26</b>
<i>II.4.1 Processus de conception, modélisation et exploration</i>	<i>26</i>
<i>II.4.2 Le "design scientifique" et la "science du design"</i>	<i>27</i>
<b>II.5 Le "Design Science" dans le domaine des SI</b>	<b>28</b>
<b>II.6 Mes contributions aux questions méthodologiques en informatique et SI</b>	<b>30</b>
<i>II.6.1 Ingénierie des méthodes : synthèse sur l'adaptabilité et l'adaptation</i>	<i>30</i>
<i>II.6.2 IDM et ingénierie des besoins : synthèse des travaux d'un Workshop</i>	<i>32</i>
<i>II.6.3 Créativité et ingénierie des besoins : une revue de la littérature</i>	<i>33</i>
<i>II.6.4 Synthèse sur les méthodes empiriques en ingénierie des SI</i>	<i>33</i>
<i>II.6.5 Synthèse sur les méthodes de revue de la littérature</i>	<i>35</i>
<i>II.6.6 Réplication d'une étude en génie logiciel empirique</i>	<i>35</i>
<i>II.6.7 Théorisation dans la recherche empirique sur les ERP : études exploratoires</i>	<i>36</i>
<b>II.7 Conclusion du chapitre</b>	<b>38</b>
<b>III. Chapitre 3 : L'exécution de modèles conceptuels</b>	<b>39</b>
<b>III.1 Préambule</b>	<b>40</b>
<b>III.2 Problématique</b>	<b>41</b>
<i>III.2.1 Définir un langage</i>	<i>41</i>
<i>III.2.2 Définir sa sémantique</i>	<i>42</i>
<i>III.2.3 Exécution d'un langage vs. "process enactment"</i>	<i>43</i>
<i>III.2.4 Conclusion et positionnement</i>	<i>43</i>
<b>III.3 Bref retour sur l'exécutabilité du modèle REMORA</b>	<b>44</b>

<b>III.4 Une approche ad-hoc pour interpréter les cartes MAP</b>	<b>45</b>
III.4.1 Contexte : modélisation intentionnelle et carte MAP	45
III.4.2 La sémantique d'une carte Map	46
III.4.3 Tentative de formalisation de la sémantique d'un MAP	47
III.4.4 Une architecture de base de données pour exécuter les cartes	48
III.4.5 Conclusion sur l'approche ad-hoc	49
<b>III.5 Vers une approche générique pour l'expression de l'exécutabilité</b>	<b>49</b>
III.5.1 Une approche de type IDM pour définir l'architecture d'exécution	49
III.5.2 Modélisation événementielle de la sémantique	50
III.5.3 Exploitation de la sémantique opérationnelle	51
III.5.4 Application sur le modèle de carte MAP	53
<b>III.6 Conclusion et perspectives de recherche</b>	<b>54</b>
<b>III.7 Annexe 1 : Diagramme de classe du moteur d'exécution d'un modèle de carte (niveau instances)</b>	<b>56</b>
<b>IV. Chapitre IV : Exploitation des rapports d'erreurs logicielles pour la prédiction du temps de résolution</b>	<b>57</b>
<b>IV.1 Préambule</b>	<b>58</b>
<b>IV.2 Gestion et traitement des erreurs logicielles</b>	<b>59</b>
<b>IV.3 Bref état de l'art sur la prédiction du temps de résolution des erreurs logicielles</b>	<b>60</b>
IV.3.1 Apprentissage supervisé avec la technique $\alpha$ -kNN	60
IV.3.2 Apprentissage non supervisé avec la technique du partitionnement	61
<b>IV.4 Problématique</b>	<b>62</b>
IV.4.1 Brève présentation de l'étude initiale	62
IV.4.2 Estimation de la répliquabilité de l'étude initiale	63
IV.4.3 Techniques de simulation pour le test d'hypothèse	64
<b>IV.5 Partie 1 : réplification de l'étude Raja (2013)</b>	<b>64</b>
IV.5.1 Description des jeux de données	65
IV.5.2 Le protocole expérimental de la réplification	66
IV.5.3 Les résultats de la réplification	66
IV.5.4 Conclusion sur la partie 1	68
<b>IV.6 Partie 2 : test opérationnel de l'hypothèse émise par Raja (2013)</b>	<b>69</b>
IV.6.1 Le protocole expérimental du scénario de simulation	69
IV.6.2 Premiers résultats	72
IV.6.3 Modification de la taille de l'ensemble de test	74
IV.6.4 Augmentation du nombre de partitions ( $K = 6, 8$ et $10$ )	75
IV.6.5 Conclusion sur la partie 2	76
<b>IV.7 Conclusion et perspectives de recherche</b>	<b>78</b>
<b>V. Chapitre V : Enseignement en ligne et prise en compte des styles d'apprentissage</b>	<b>79</b>
<b>V.1 Préambule</b>	<b>80</b>
<b>V.2 Problématique</b>	<b>80</b>
V.2.1 Caractéristiques des apprenants, styles cognitifs et styles d'apprentissage	81
V.2.2 Constats empiriques de l'effet des styles d'apprentissage	82
V.2.3 Personnalisation de l'enseignement et systèmes adaptatifs	82
<b>V.3 Intégration des styles d'apprentissage dans un dispositif d'enseignement</b>	<b>83</b>
V.3.1 Le modèle de Felder & Silverman	83
V.3.2 Proposition d'une grille d'adaptation des contenus multimédia aux styles d'apprentissage	84
V.3.3 Evaluation empirique de la grille d'adaptation	86
V.3.4 Conclusion	86

<b>V.4 Impact des styles d'apprentissage dans l'adoption de l'apprentissage mobile</b>	<b>87</b>
V.4.1 <i>Le Mobile Learning</i>	87
V.4.2 <i>Les modèles d'acceptation de la technologie</i>	87
V.4.3 <i>Le modèle des styles d'apprentissage de Kolb</i>	88
V.4.4 <i>Proposition de nouveaux modèles théoriques</i>	90
V.4.5 <i>Enquête quantitative – phase 1</i>	90
V.4.6 <i>Enquête quantitative – phase 2</i>	91
V.4.7 <i>Analyse des données et modélisation PLS</i>	92
<b>V.5 Conclusion et perspectives de recherche</b>	<b>94</b>
<b>VI. Chapitre VI : Conclusion générale</b>	<b>95</b>
<b>VII. Production et rayonnement scientifique</b>	<b>97</b>
<b>VII.1 Synthèse bibliométrique</b>	<b>97</b>
<b>VII.2 Publications</b>	<b>97</b>
VII.2.1 <i>Revue avec comités de lecture</i>	97
VII.2.2 <i>Co-édition d'ouvrage</i>	98
VII.2.3 <i>Chapitres d'ouvrage</i>	98
VII.2.4 <i>Conférences et ateliers avec comité de programme</i>	99
VII.2.5 <i>Autres publications</i>	101
<b>VII.3 Financement de projets de recherche</b>	<b>101</b>
<b>VII.4 Animation de la recherche et travaux d'expertise</b>	<b>101</b>
VII.4.1 <i>Membre de comités éditoriaux</i>	101
VII.4.2 <i>Coordination de numéros spéciaux de revues</i>	101
VII.4.3 <i>Comité de programme de conférences et atelier de recherche</i>	102
VII.4.4 <i>Organisation et animation de la recherche</i>	102
VII.4.5 <i>Comités scientifiques</i>	102
<b>VII.5 Direction de recherche</b>	<b>102</b>
VII.5.1 <i>Encadrement de thèse</i>	102
VII.5.2 <i>Appartenance à des jurys de thèse</i>	102
<b>VIII. Références bibliographiques</b>	<b>103</b>

## Index des figures

Fig. I-1. Analyse ontologique : trois perspectives des sciences informatiques .....	12
Fig. I-2. Cadre général d'analyse de la recherche en SI utilisé par K. Lyytinen dans [152] .....	14
Fig. I-3. Aperçu volumétrique de ma production scientifique .....	15
Fig. I-4. Synopsis général de l'activité de conception d'artéfact (d'après [226]), et les deux phases de l'ingénierie du SI (d'après [196]) .....	16
Fig. II-1. Le cycle de la recherche scientifique (d'après M. Bunge dans [34], p.9) .....	24
Fig. II-2. Synopsis d'une expérimentation .....	24
Fig. II-3. Synopsis d'un processus générique de conception (gauche), et droite, un exemple de l'exploration de l'espace des solutions pour la conception d'un abri de jardin facilement transportable (Source : [226], p.15 et p.46).....	27
Fig. II-4. Cadre de référence de Hevner et al. ([103], p. 16) .....	28
Fig. II-5. Cadre de référence de Wieringa et distinction entre problème pratique et problème de connaissance ([243]).....	29
Fig. II-6. Principe des approches adaptables (g), et principes des méthodes adaptatives (d) .....	31
Fig. II-7. Typologie des techniques de créativité analysées dans [publi#01] .....	33
Fig. III-1. La modélisation conceptuelle dans l'ingénierie des SI (d'après Rolland et Prakash (2000), [196]) .....	40
Fig. III-2. Modèle transformationnel de l'analyse, la conception et l'implémentation d'un SI ([237], p.63) ; ou encore, le mapping à l'aide de scripts du monde réel vers le monde de la machine ([235], p.220) .....	41
Fig. III-3. Représentation implicite et explicite de la sémantique d'un langage (d'après [153], p.57) .....	42
Fig. III-4. Architecture de l'atelier Rubis .....	44
Fig. III-5. Cadre général proposé dans ma thèse de doctorat pour l'expression du processus de transformation de modèles ([7], p.9) .....	45
Fig. III-6. Deux exemples de carte MAP.....	46
Fig. III-7. Illustration des sections candidates dans une carte Map.....	47
Fig. III-8. Exemple de spécification d'opérations ([66], p.211).....	47
Fig. III-9. Modèle conceptuel des données pour le référentiel du moteur d'exécution des modèles de carte MAP ([66], p.130) .....	48
Fig. III-10. Modèle conceptuel des données de la trace d'exécution des cartes Map ([66], p.131).....	48
Fig. III-11. Synopsis de l'approche .....	50
Fig. III-12. Structure générale du moteur d'exécution .....	51
Fig. III-13. Transcription de l'événement interne en une spécification orientée objet .....	52
Fig. III-14. Modèle structurel à deux niveaux (concepts-types et instances) du Map .....	53
Fig. III-15. Spécification événementielle de la sémantique d'exécution du Map .....	54
Fig. IV-1. Cycle de vie d'un rapport d'erreur (adapté de (Zeller 2009) [255]).....	59
Fig. IV-2. Synopsis de la technique de prédiction par apprentissage supervisé développé dans Weiss et al. 2007 .....	60
Fig. IV-3. Synopsis de l'étude expérimentale. ....	66

Fig. IV-4. Distribution des TRE pour les trois jeux de données (temps de résolution exprimé en jours) .....	67
Fig. IV-5. Les moyennes des TRE dans les quatre partitions pour chaque jeu de données .....	67
Fig. IV-6. Des tranches de données de taille croissante pour le scénario de simulation .....	69
Fig. IV-7. Synopsis de la partie 2 test de l'étude (cf. section §IV.5.2 pour la partie préparation des données) .....	70
Fig. IV-8. Définition de la portion de rapports d'erreur ouverts utilisé pour la simulation dans une tranche de données SLi .....	70
Fig. IV-9. Synopsis du scénario de test pour la prédiction du TRE .....	71
Fig. IV-10. Résultat du test post-hoc de Games-Howell pour les partitions pour chaque jeu de données.....	73
Fig. IV-11. La précision de la prédiction du TRE pour les trois jeux de données, seuil d'erreur $\alpha = 25\%$ .....	73
Fig. IV-12. La précision de la prédiction du TRE pour les trois jeux de données, seuil d'erreur $\alpha = 50\%$ .....	74
Fig. IV-13. La précision de prédiction du TRE pour les trois jeux de données avec des valeurs différentes du paramètre Taille du Facteur de Simulation (0.1, 0.3 et 0.5), et avec un seuil d'erreur de 25% et 50% .....	75
Fig. IV-14. Résultats du test post-hoc Games-Howell pour les tranches de données de Company A avec $K=6$ et $K=8$ .....	76
Fig. IV-15. La précision de prédiction du TRE pour des valeurs différentes du nombre K de partitions, en utilisant deux valeurs du paramètre Taille du Facteur de Simulation (0.3 et 0.5), et avec un seuil d'erreur de 25% et 50% .....	77
Fig. V-1. Cadre d'analyse pour la recherche sur l'apprentissage médiatisé par les TIC ([3], p.5) .....	80
Fig. V-2. Intégration des stratégies d'enseignement avec les styles d'apprentissage et les e-média .....	83
Fig. V-3. Correspondance entre les styles d'apprentissage et les stratégies d'enseignement ([82], p.108).....	85
Fig. V-4. Correspondance entre les styles d'apprentissage et les médias électroniques ([82], p.109) .....	86
Fig. V-5. Le modèle théorique UTAUT sur lequel s'est basée cette recherche (extrait de [228], p.477) .....	88
Fig. V-6. Le modèle descriptif des styles d'apprentissage de Kolb (d'après [44], p.62). .....	89
Fig. V-7. Le premier modèle théorique proposé pour la compréhension du ML (1ère phase) .....	90
Fig. V-8. Le second modèle théorique proposé pour la compréhension du ML (2nde phase).....	91
Fig. V-9. Modèle structurel pour la phase 1 .....	92
Fig. V-10. Le modèle obtenu après analyse PLS, cas général (uniquement les liens significatifs sont montrés).....	93
Fig. V-11. Le modèle obtenu après analyse PLS, cas des styles AE/RO (les liens en gras indique une modération positive).....	93

## Index des tableaux

Tableau I-1. Résumé des principaux travaux de recherche menés en collaboration avec le CRI .....	17
Tableau I-2. Résumé des principaux travaux de recherche relatifs aux questions d'usage des artefacts technologiques .....	18
Tableau II-1. Principales méthodes empiriques en ingénierie des SI ([publi#03], p.21) .....	34
Tableau IV-1. Les analyses statistiques appliquées dans l'étude initiale Raja (2013).....	62
Tableau IV-2. Evaluation de la réplicabilité de l'étude Raja (2013) .....	63
Tableau IV-3. Comparaison entre l'expérience initiale dans Raja (2013) et la réplique que nous avons réalisée .....	65
Tableau IV-4. Aperçu des jeux de données utilisées dans cette étude .....	65
Tableau IV-5. Analyse de la variance pour les quatre partitions dans les trois jeux de données .....	68
Tableau IV-6. Les valeurs-p pour le test post-hoc Games-Howell (les valeurs en gras ne sont pas significatives à la limite $\alpha=0.05$ ).....	68
Tableau IV-7. Résultats du test ANOVA pour les partitions obtenues dans chaque tranche de données du scénario de simulation (les valeurs en gras ne sont pas significatives à la limite $\alpha=0.05$ ) .....	72
Tableau V-1. Les quatre dimensions du style d'apprentissage de Felder & Silverman ([75]) .....	84
Tableau V-2. Les neuf stratégies d'enseignement identifiées dans la littérature ([82], p.104) .....	84
Tableau V-3. Liste des 29 e-média recensés dans la littérature et sur le web ([82], p. 46–49).....	85
Tableau V-4. Aperçu de l'échantillon utilisé dans l'évaluation empirique .....	86
Tableau VI-1. Questionnements soulevés dans l'introduction de ce manuscrit .....	96

# I. Chapitre I : Introduction générale

## Sommaire

<u>I.1</u>	<u>Préambule</u>	10
<u>I.2</u>	<u>Analyse ontologique : les sujets d'étude en informatique</u>	11
<u>I.2.1</u>	<u>Trois perspectives d'étude</u>	11
<u>I.2.2</u>	<u>Logiciel versus Système d'Information</u>	12
<u>I.2.3</u>	<u>Génie logiciel versus Ingénierie des SI</u>	13
<u>I.2.4</u>	<u>Conception et développement versus Usage et impact du SI</u>	13
<u>I.3</u>	<u>Analyse méthodologique : les méthodes de recherche</u>	14
<u>I.4</u>	<u>Positionnement de mes travaux de recherche</u>	15
<u>I.4.1</u>	<u>Thème 1 : modèles et méthodes pour l'ingénierie des SI</u>	15
<u>I.4.2</u>	<u>Thème 2 : usages et adoption des artefacts informatiques et sociotechniques</u>	18
<u>I.5</u>	<u>Organisation du mémoire</u>	19

## I.1 Préambule

L'informatique renvoie à des significations multiples. Dans un ouvrage à ce propos de la série *Que-Sais-Je* [2], les auteurs considèrent que l'informatique est d'abord **un ensemble d'outils technologiques** pour programmer des logiciels et réaliser des calculs complexes. C'est ensuite **une fonction** dans une entreprise qui vise à exploiter de manière optimale ces outils pour aboutir à un Système d'Information adéquat au service d'une organisation et de sa stratégie. Avec l'essor d'Internet, des terminaux mobiles et des réseaux sociaux, j'ajouterais que l'informatique est devenue **un médium numérique ubiquitaire** qui supporte et transforme des pans entiers de l'activité humaine et sociale telles que l'éducation ou l'administration publique [42]. Enfin, et c'est l'aspect le plus important au niveau de cette introduction, l'informatique est **une discipline de recherche scientifique** avec des spécialisations multiples et des savoirs d'envergure encyclopédique<sup>1</sup> [1].

Dans mon parcours de chercheur en informatique et en systèmes d'information, j'ai été confronté à des thématiques de recherche et des problématiques diverses ainsi que des pratiques méthodologiques variées. Les travaux de recherche que j'ai menés m'ont conduit à développer un regard **épistémologique** sur mes travaux, à explorer les fondements de la recherche scientifique, et à interroger le positionnement spécifique de l'informatique et des systèmes d'information (SI) en tant que champs d'investigation multidisciplinaires. Des interrogations similaires sont parues dès les années 80 (p. ex. [87], [11]). Elles restent d'actualité, et sont posées – sous différentes formes – par d'éminents chercheurs tels que P. Dening de l'ACM<sup>2</sup>[58], J. Livari [112] et R. Watson [239] de l' AIS<sup>3</sup>, ou encore G. Dowek de l'INRIA [61]. Par ailleurs, ce type d'interrogations est important pour le développement de la pensée scientifique ; par exemple, le projet de recherche du CNRS en 2002 mentionne explicitement « *l'impératif de renforcer systématiquement une certaine pratique de l'autoréflexivité scientifique* » ([43], p.14).

Dans ce mémoire, j'adopte une vision élargie des sciences de l'informatique pour présenter mes différents travaux consécutifs à la collaboration avec des équipes de recherche différentes. Un des objectifs de cette HDR est de montrer comment cette vision concourt à clarifier mon parcours de chercheur, à situer mes contributions scientifiques, ainsi qu'à expliciter une **compréhension holistique et multidisciplinaire** des approches de recherche autour de l'artéfact informatique et sociotechnique. Cette vision élargie s'appuie sur une analyse épistémologique des sciences de l'informatique à travers deux principaux questionnements :

- Le premier questionnement, que je qualifierais d'**ontologique**, considère la nature différente des problématiques et questions de recherche que soulève l'informatique. Un artéfact informatique (p. ex. un programme ou un logiciel) soulève des questions multiples telles que : est-il exempt d'erreurs ? est-il valide par rapport au problème qu'il est censé résoudre ? est-il utile du point de vue de l'utilisateur final ? comment le construire de manière à garantir l'absence d'erreurs, la validité par rapport aux besoins et l'utilité ?

- Le second questionnement est **méthodologique**, il considère la méthode de recherche qui est mobilisée pour mener une investigation et trouver des réponses aux questions posées. Démontrer qu'un programme informatique est exempt d'erreurs ferait ainsi appel à une technique de test ou à l'élaboration d'une preuve formelle, l'étude de l'utilité d'un logiciel nécessiterait une enquête auprès d'un échantillon d'utilisateurs et d'un raisonnement statistique, alors que sa construction ferait appel à une méthode d'ingénierie avec un modèle du cycle de vie et des notations pour modéliser le logiciel. La nature des connaissances obtenues dans chacun des cas (démonstration formelle, indices statistiques, méthode d'ingénierie) est différente ; en particulier, la *validité* et la *véracité* de ces connaissances renvoient à des postures épistémologiques assez distinctes. Dans ma réflexion, le questionnement méthodologique complète la perspective ontologique et constitue l'autre facette de mon analyse de la multidisciplinarité des sciences informatiques.

<sup>1</sup> Dix neuf sections et 170 articles dans *Encyclopédie de l'Informatique et des SI*, Akoka & Comyn-Wattiau, Vuibert, 2006.

<sup>2</sup> ACM : Association for Computing Machinery

<sup>3</sup> AIS : Association for Information Systems

Dans cette introduction, je vais développer et expliciter le premier questionnement (le questionnement ontologique) pour ensuite présenter une problématique générale dans laquelle se situent mes travaux. Ce faisant, j'expose ma vision de ce qu'est la multidisciplinarité. Le second questionnement (les méthodologies de recherche) fera l'objet d'un chapitre distinct dans la suite du mémoire (Chap. II)

## I.2 Analyse ontologique : les sujets d'étude en informatique

Dans la classification disciplinaire des sciences, la perspective ontologique permet d'organiser les champs de recherche et les connaissances qui s'y rattachent selon l'objet – ou le phénomène – étudié et les questions qui sont posées ([34], p.25). Dans le cadre de ce mémoire, la perspective ontologique est la première facette de ma vision de la multidisciplinarité des sciences informatiques.

### I.2.1 Trois perspectives d'étude

L'informatique puise ses sources premières dans une branche des mathématiques qui raisonne sur les calculs réalisés par des automates imaginaires (telle que la machine de Turing en 1936). A ses début donc, l'informatique est naturellement considérée comme une **science du formel**, c.-à-d. une science qui traite d'idées et d'abstractions ([34], p.21). Ses sujets de prédilection sont algorithmes, complexité et preuves de programmes [56]. Néanmoins, les termes "**logiciel**" et "**génie logiciel**" émergent rapidement pour désigner un objet et des problématiques qui dépassent les périmètres restreints du programme et de la programmation. Le génie logiciel met en exergue une activité d'ingénierie où les sciences informatiques sont *appliquées* pour résoudre des problèmes et pour répondre à des besoins concrets liées à la vie humaine. La question n'est plus la construction de programmes en soi, mais la conception d'un artefact – le logiciel – selon un cahier des charges qui se réfère à un domaine d'activité humaine. Dans cette science appliquée, *l'utilité* et *l'ingéniosité* sont des enjeux de premier plan.

**Les Sciences de l'Artificiel**, définis par H.A. Simon en 1969 [216], introduisent un cadrage théorique important à cette problématique. Le champ des sciences de l'artificiel se définit à l'intersection entre l'artefact technologique, qui obéit aux lois régissant sa structure interne, et les objectifs et contraintes de l'environnement externe auquel il est destiné ([216], p.113). La raison d'être d'un logiciel est de résoudre un problème donné, de satisfaire les besoins d'un ensemble d'utilisateurs et de contribuer à leurs objectifs. Le **génie logiciel** (GL) se définit ainsi comme une **science de conception** qui exploite les lois formelles qui gouvernent les artefacts informatiques de manière à résoudre le problème posé, à garantir l'utilité, à maximiser la satisfaction des besoins des utilisateurs et à minimiser les propriétés et effets indésirables. Il soulève des problèmes de nature **mal-structurée** : la formulation d'un problème et de sa solution sont intrinsèquement liés [217]. La meilleure solution étant très difficile ou impossible à trouver, on ne peut que converger vers un compromis suffisamment utile et satisfaisant ([216], p.120).

Enfin, l'informatique peut aussi être étudiée en tant que **science des faits**, c.-à-d. une science qui réfère à des événements et des phénomènes factuels et observables, et qui vise à les comprendre, les expliquer, et ultimement, les prédire ([34], p.21). La conception, la production et l'utilisation d'un artefact informatique dans un contexte organisationnel, social et humain sont des phénomènes qui donnent lieu à des faits observables et mesurables. L'analyse de ces faits permet de comprendre et d'expliquer une multitude de questions relatives aux succès et échecs des méthodes et projets d'ingénierie, aux usages réels des artefacts et à la perception des utilisateurs de l'utilité de ceux-ci, ou encore aux interactions et effets de ces artefacts sur le contexte et l'environnement d'usage.

L'informatique en tant que sujet d'étude définit ainsi trois perspectives complémentaires (cf. Fig. I-1). Ces trois perspectives correspondent généralement à des organisations disciplinaires distinctes et à des communautés différentes de chercheurs. Plusieurs exemples permettent d'illustrer la complémentarité de ces trois perspectives:

- Les **bases de données** modernes apparaissent avec le modèle relationnel et son algèbre formelle définis en 1970 par E.F. Codd. La conception des premiers prototypes (tels que System R [9]) aboutit au développement de logiciels puissants et robustes. L'utilisation d'un SGBD est par la suite étudiée en tant que phénomène organisationnel qui soulève des questions telles que l'adoption (p.ex. [93]).

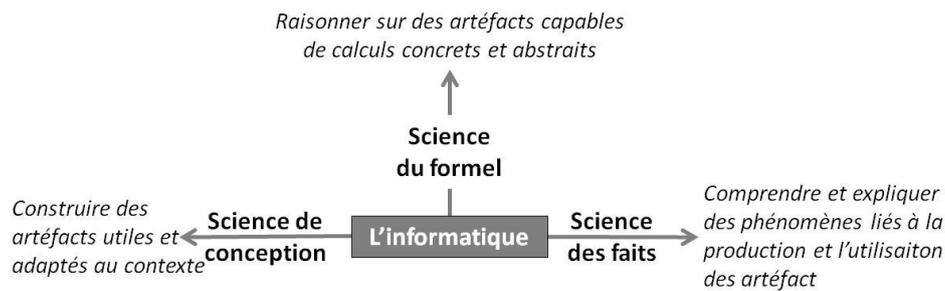


Fig. I-1. Analyse ontologique : trois perspectives des sciences informatiques

– Le **langage de modélisation** orienté objet **UML** a émergé de la pratique vers la fin des années 90. Les fondements formels seront explorés à posteriori (p.ex. [94]), alors que l'analyse empirique des usages (p.ex. [60], [33]) et de l'utilité (p.ex. [40]) sont des questionnements qui restent d'actualité.

– A contrario pour les **langages de programmation**, alors que la définition formelle d'un langage est une problématique très explorée en informatique, assez peu d'études finalement s'intéressent aux usages et aux facteurs d'adoption (p.ex. [158]).

– Pour les **progiciels intégrés** (ERP), les conditions d'usages et de succès, ainsi que les méthodes de configuration vis-à-vis des exigences métiers, sont abondamment étudiées (p.ex. [209], [54]). Cependant, à notre connaissance, aucun fondement formel n'a été envisagé pour ces systèmes logiciels complexes.

## I.2.2 Logiciel versus Système d'Information

L'utilisation des logiciels dans les organisations a rapidement soulevé de nouveaux questionnements et nécessitait de nouvelles notions pour l'appréhender. L'émergence du concept de **Système d'Information** (SI) s'appuie sur deux idées fondamentales :

– La première est la distinction déterminante entre **donnée**, stockée et manipulée par un ordinateur, et **information**, comprise et exploitée par un acteur humain pour mener une activité à but administratif ou managérial, et ayant un certain impact sur son environnement. Cette distinction est explicitée pour la première fois en 1968 par Börje Långfors avec les niveaux *Datalogical* et *Infological* [138]. Elle sera codifiée par la suite dans le célèbre schéma AN-SI/X3/SPARC pour bases de données avec les niveaux physique, logique et conceptuel.

– La seconde idée reconnaît l'immense potentiel de l'artéfact logiciel à devenir un organe **opérationnel**, de **contrôle** et de **pilotage** au service de l'organisation et de ses acteurs selon **une vision systémique** issue de la cybernétique et des systèmes complexes [140].

Aussi, dès la fin des années 70 [106], le SI est perçu comme un **artéfact sociotechnique**, c.-à-d. un système technique avec des conséquences sociales, ou encore, *un système social implémenté grâce aux TIC* [107]. Au sein de ce système, la technologie informatique est intrinsèquement liée aux contextes d'usage et aux objectifs de toutes les parties prenantes. Le concept de SI révèle en conséquence **la multiplicité des rôles** que joue l'artéfact logiciel :

– En tant qu'*automate* régi par des algorithmes et soumis à des contraintes technologiques (p.ex. le stockage d'une date), il modifie les processus de travail, impose des restrictions et crée de nouvelles opportunités d'affaires [4].

– En tant qu'*artéfact pensé et conçu pour servir l'organisation*, il est façonné par les besoins des utilisateurs et défini selon des cahiers de charge issus des contextes et finalités d'usage.

– Enfin, c'est le vecteur d'une *force interne* à l'organisation qui émerge de l'action, des choix opérationnels et stratégiques des acteurs, et de l'appropriation des individus. Cette force impacte et transforme les contextes et les finalités, générant de nouveaux besoins (p.ex. l'interopérabilité des systèmes) ainsi que de nouveaux problèmes (p.ex. la confidentialité des données) [172].

Bref, la conception, la mise en place et l'exploitation du SI soulèvent des questions et des problématiques qui dépassent largement la construction (ou l'acquisition) et l'installation d'un système informatique. Pour autant, l'étude du SI et de l'artéfact logiciel sur lequel il repose relève-t-elle des sciences des faits ou des sciences de conception ? Dans

quelle mesure l'étude de l'un éclaire-t-elle l'étude de l'autre ? Dans quelle mesure est-il possible d'étudier l'un (p.ex. le SI) en ignorant l'autre (p.ex. le logiciel) ? Est-il nécessaire et/ou pertinent de combiner l'étude de l'un et de l'autre, et si oui, comment faire et quel est l'impact sur les méthodes de recherche ? Comment se distinguent les critères d'évaluation des contributions scientifiques lorsque l'on étudie l'un ou l'autre ?

Ces questionnements ont constitué la toile de fond de ma réflexion depuis une décennie, ils ont émergé au fil des travaux et collaborations réalisés avec des équipes de recherche appartenant à des champs disciplinaires différents en informatique et en SI. Au-delà de la présentation *per se* de mes travaux, le second objectif de cet HDR est d'apporter des éclairages et des éléments de réponses à ces questions.

### 1.2.3 Génie logiciel versus Ingénierie des SI

En tant que science de conception, le génie logiciel vise la construction et la maintenance de logiciels qui satisfont les besoins des utilisateurs selon des approches systématiques et rigoureuses basées sur des connaissances scientifiques et théoriques de l'artéfact (le logiciel). L'adjonction du vocable « génie » entend bien de faire de la construction des logiciels un domaine d'activité similaire au génie civil ou le génie électrique, et d'obtenir de ce fait, les mêmes critères de fiabilité, de sécurité et de facilité d'entretien que les grandes œuvres d'ingénierie [190]. Le génie logiciel impose donc des connaissances et des pratiques qui sont distinctes de ce qu'on appelle « *Computer Science* » en anglais, et qu'on peut traduire ici par « informatique formelle ». Cette distinction, induite par le rattachement du génie logiciel aux sciences de conception, a mis du temps à s'imposer [177]. Ce n'est par exemple qu'en 2001 que deux cursus<sup>4</sup> d'enseignement différents ont été promulgués par les associations IEEE et ACM [190].

A l'aune de la distinction faite entre le logiciel et le SI, l'ingénierie des SI se distinguerait du génie logiciel par **les dimensions organisationnelles et managériales** inhérentes aux projets de construction de SI [117]. En réalité, beaucoup de problématiques et d'approches techniques et théoriques sont communes aux deux<sup>5</sup> : on peut citer l'ingénierie des besoins, l'ingénierie dirigée par les modèles, la modélisation des données et des systèmes (i.e. UML), ou encore, les environnements de développement et les outils de type CASE.

On peut à ce niveau s'interroger de nouveau : est-il possible de faire de l'ingénierie des SI sans faire – serait-ce un peu – de génie logiciel, et vice-versa ? Dans quelle mesure les recherches et contributions scientifiques en ingénierie des SI sont-elles distinctes et s'évaluent-elles différemment de celles en génie logiciel, et vice-versa ? Autrement dit, y aurait-il des différences significatives dans les fondements épistémologiques de la recherche en ingénierie des SI et celle en génie logiciel ? Ces questionnements complètent ceux de la rubrique précédente, et l'objectif de ce mémoire d'HDR est de contribuer à y apporter quelques éléments de réponse.

### 1.2.4 Conception et développement versus Usage et impact du SI

Dès que le SI a été reconnu comme sujet d'étude en informatique, des orientations assez divergentes se manifestent pour l'étudier. Ces divergences se sont cristallisées lors de la création par l'IFIP en 1975 du comité technique TC8 et des trois groupes<sup>6</sup> de travail [230] : les Working Group TC8.1, TC8.2 et TC8.3 consacrés respectivement à **la conception et l'évaluation des SI**, à **l'interaction du SI avec l'organisation** et aux **systèmes de support à la prise de décision**. Le premier groupe a été fortement influencé à ses débuts par le génie logiciel, mais il a contribué par la suite à l'émergence de l'ingénierie des SI en tant que domaine d'étude spécifique. Le second a été dominé par des visions et des analyses issues des sciences de l'organisation, des sciences de gestion et des sciences humaines [167] ; il a abouti à la création de l'association AIS et de la communauté de recherche en MIS (*Management Information Systems*).

<sup>4</sup> Le *SWEBOKK* pour le génie logiciel et le *Computing curricula CC2001* pour l'informatique formelle (d'après [190], p. 1054)

<sup>5</sup> Notons que dans *l'Encyclopédie de l'Informatique et des SI* [1], la section consacrée au génie logiciel apparaît dans la partie de l'ouvrage dédiée à la dimension *technologique* du SI, alors que celle de l'ingénierie des SI apparaît dans la partie dédiée à la dimension *organisationnelle*.

<sup>6</sup> Le TC8 sera par la suite élargi ; il compte actuellement 8 groupes [source : <http://www.ifip.org/bulletin/bulltcs/memtc08.htm>]

Toutefois, et malgré ces orientations différentes, certains chercheurs donnèrent dès les années 80 une analyse et une compréhension globale du domaine des SI. C'est le cas notamment de Kalle Lyytinen qui, dans une étude exhaustive de la littérature en 1987, discute l'ensemble des problématiques de recherche en SI [152]. Son analyse s'appuie sur un cadre théorique qui organise la recherche en SI selon trois processus : l'usage, le développement et l'exploitation (cf. Fig. I-2). De son analyse, il ressort clairement que les problématiques relatives au développement et aux usages, ainsi que les perspectives futures de la recherche pour y répondre, ne sont pas totalement indépendantes. Ainsi, « les progrès dans la modélisation du SI » auraient un impact significatif autant sur le processus de développement que sur les processus d'utilisation (Ibid., p.35). De même, « les avancées dans les théories » qui décrivent et expliquent l'usage et l'impact du SI auraient un impact significatif sur certaines problématiques du processus de développement (telle que les objectifs de ce processus). D'ailleurs, l'auteur indique dès le début de son étude que les problèmes rencontrés lors du développement impactent fortement le système développé et son utilisation, et qu'on ne peut pas totalement séparer ces deux processus d'analyse et de compréhension (Ibid., p.10).

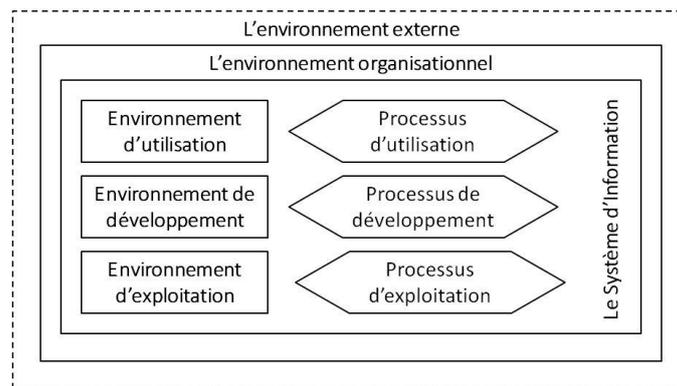


Fig. I-2. Cadre général d'analyse de la recherche en SI utilisé par K. Lyytinen dans [152]

On peut de nouveau s'interroger : en quoi l'étude du développement du SI peut-elle renseigner l'étude des usages (et vice-versa) ? Ces deux domaines d'étude sont-ils disjoints, ou y a-t-il des intersections et des points de vue communs qui pourraient être exploités ? Quelles seraient les différences épistémologiques et méthodologiques entre les deux sujets d'étude ? On tentera d'apporter des éléments de réponse à ces questions complexes.

### I.3 Analyse méthodologique : les méthodes de recherche

Dans la vision holistique de l'informatique et des SI que je développe, le second grand questionnement concerne la méthode de recherche, c.-à-d. la démarche intellectuelle qu'adopte le chercheur pour élaborer des réponses aux questions qu'il pose. L'analyse du questionnement méthodologique ramène vers les grandes idées qui ont agité le développement de la pensée scientifique au 19<sup>e</sup> et 20<sup>e</sup> siècles, telles que l'approche cartésienne et le rationalisme, le positivisme, l'expérimentation et la falsifiabilité des énoncés scientifiques, l'empirisme et l'induction, la cybernétique et la systémique, etc. (p.ex. [141], [12]). **Un chapitre de ce mémoire** (chapitre II) est consacré à une discussion synthétique des questions méthodologiques dans les sciences informatiques ; chapitre dans lequel je présenterais aussi plusieurs contributions récentes relatives à ces questions.

Pour en faire un bref résumé ici, je dirais que l'objectif des sciences des faits est de **comprendre et expliquer** les phénomènes du monde réel (p.ex. la photosynthèse), alors que celui des sciences de conception est de **le changer et de l'améliorer** (p.ex. augmenter le rendement des terres agricoles à l'aide d'engrais). La connaissance produite dans le premier cas est *descriptive* et *explicative*, sa véracité doit pouvoir être scientifiquement démontrée [34].

Dans les sciences de conception, il y a d'abord production d'un artéfact, ou d'une représentation de celui-ci (plan, modèle, maquette, etc.) ; s'il y a production de connaissance, elle est *prescriptive* et prend la forme de procédés opératoires, de méthodes et d'outils conceptuels et technologiques pour la construction d'artéfacts. L'impératif de **vérité scientifique** des connaissances produites dans les sciences de conception glisse subtilement pour concerner la *nouveauté*, *l'originalité*, *l'utilité* de l'artéfact et la *faisabilité* de sa réalisation. La validité et la pertinence des procédés,

méthodes et outils, peuvent alors être traitées comme des phénomènes qu'on peut questionner et chercher à comprendre et à expliquer selon les démarches méthodologiques des sciences des faits (cf. chapitre II).

## I.4 Positionnement de mes travaux de recherche

Mes travaux portent autant sur l'étude de l'**artéfact logiciel** que de l'**artéfact sociotechnique**. Ils concernent en premier lieu **la conception et le développement** de tels artéfacts, mais aussi **l'utilisation et l'adoption** de ces artéfacts dans des contextes d'usage spécifiques. Certains de ces travaux abordent l'informatique en tant que **science de conception** et relèveraient de l'ingénierie des SI ou du génie logiciel, d'autres abordent l'informatique en tant que **science des faits** et relèveraient du génie logiciel empirique ou des sciences de l'éducation et de gestion.

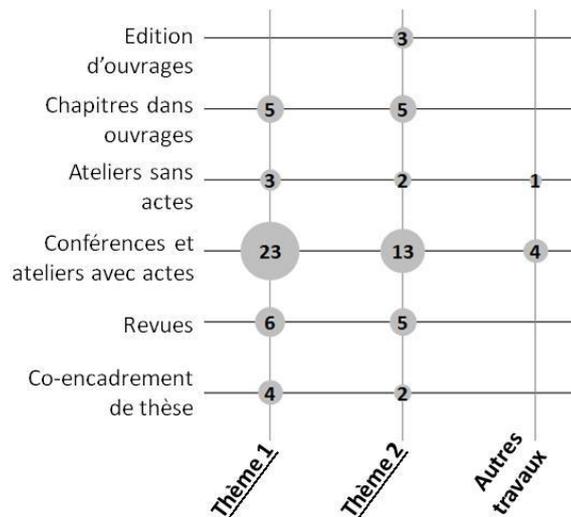


Fig. I-3. Aperçu volumétrique de ma production scientifique

Pour organiser la présentation de mes travaux, je fais le choix délibéré de les regrouper en deux thématiques selon le cadre d'analyse utilisé par K. Lyytinen dans [152] (cf. Fig. I-2). La Fig. I-3 donne un aperçu volumétrique de ma production scientifique en termes de publications et de co-encadrement de thèses pour chacun de ces thèmes.

### I.4.1 Thème 1 : modèles et méthodes pour l'ingénierie des SI

La démarche de conception dans l'ingénierie des SI emprunte celle, générale, de l'activité de conception (cf. Fig. I-4g). Dans cette démarche, la modélisation est le support de l'action et de la réflexion du concepteur ; elle lui permet de construire des représentations de l'artéfact en cours de conception, de raisonner sur ses propriétés éventuelles, d'envisager des alternatives et d'explorer l'espace des solutions<sup>7</sup>.

Contrairement aux activités de conception en architecture civile ou en design industriel, les artéfacts sociotechniques conçus dans l'ingénierie des SI englobent et codifient une connaissance substantielle sur le domaine d'activité et le contexte d'usage de ces artéfacts. En ingénierie des SI, la modélisation conceptuelle est le point focal d'interaction entre l'**ingénierie des besoins** dont le focus est sur l'univers du discours (cf. Fig. I-4d), et l'**ingénierie des systèmes** dont l'objectif est la conception de l'artéfact technique [196]. La portée de la modélisation conceptuelle penche souvent d'un côté, c.-à-d. vers le domaine d'activité (avec p.ex. acteur, processus métier et objectif stratégique, etc.), ou de l'autre, c.-à-d. vers le logiciel informatique (avec p.ex. structures de données, algorithmes, temps de réponse, etc.). Un challenge important dans l'ingénierie des SI est de réussir à **construire un continuum entre la modélisation des besoins et la modélisation des systèmes**, à prendre en compte autant le fonctionnel que le côté non-fonctionnel. La réalité est que l'ingénierie des SI fait appel à une « farandole » de modèles [144], [145] ; la maîtrise et la mise en cohérence de ces modèles sont une facette majeure de la complexité de l'ingénierie des SI [86].

<sup>7</sup> J'évoquerais ces questions avec plus de détails dans le chapitre II.

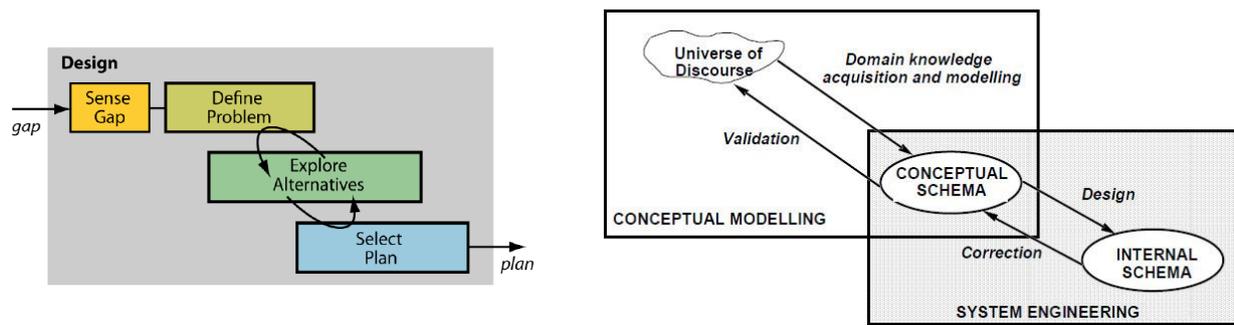


Fig. I-4. Synopsis général de l'activité de conception d'artefact (d'après [226]), et les deux phases de l'ingénierie du SI (d'après [196])

Les travaux de recherche que j'ai réalisés, ou auxquels j'ai participé, et qui relèvent de cette problématique de modélisation et son usage dans l'ingénierie des systèmes, ont été réalisés en collaboration avec des chercheurs et des doctorants du laboratoire CRI à Paris 1 dirigé par Prof. C. Rolland. Les travaux de recherche du CRI reposent en grande partie sur l'usage extensif des modèles, pour proposer des méthodes de conception des SI (méthodes d'ingénierie basées sur les modèles), pour raisonner sur les modèles (la méta-modélisation), et pour raisonner sur les méthodes elles-mêmes (ingénierie des méthodes). Deux langages de modélisation sont au cœur des travaux du CRI :

- La **modélisation à base d'événement** et méthode Remora : publié dans les années 80 [198], le modèle Remora considère l'événement, en tant que changement d'état d'un objet du système (et qui inclut l'arrivée de messages et les événements temporels), comme un concept de première classe.
- La **modélisation intentionnelle** et l'approche MAP : inspirée de l'approche de modélisation des processus orientée contexte [92], et de la modélisation des besoins à base de buts et de scénarios [199], la modélisation intentionnelle MAP se positionne à un haut niveau d'abstraction pour capturer les finalités stratégiques des acteurs et des systèmes (les intentions), ainsi que les différentes manières de les réaliser (les stratégies et tactiques de réalisation). Définies initialement en 1998 [197], les cartes MAP ont depuis été utilisées pour modéliser et raisonner sur de multiples problématiques en ingénierie des besoins, en ingénierie des systèmes et en ingénierie des méthodes.

Le Tableau I-1 donne un aperçu des principaux projets de recherche auxquels j'ai participé au CRI et qui, concrètement, correspondent à des activités de co-encadrement de thèse. Parmi ceux-ci, ma participation à l'encadrement fut à mi-parcours de thèse pour H. El Ghazi et K. Aljoumaa et à l'intégralité du parcours pour les deux autres.

Un **point commun** entre ces quatre activités est le recours à la **modélisation intentionnelle** et aux **cartes MAP**. Dans les travaux sur la gestion de la traçabilité (thèse de H. El Ghazi) et sur la publication et recherche de services intentionnels (thèse de K. Aljoumaa), les cartes MAP servent à décrire les processus d'ingénierie qui accompagnent les démarches proposées. La carte MAP permet de capturer les différentes *alternatives* de navigation dans un processus (c.-à-d. avancer vers quelle intention et selon quelle stratégie), et d'associer des *directives de guidage* pour aider l'ingénieur dans son choix ([184], p.107).

Dans les travaux de thèses de M-H. Edme et S. Mallouli, les cartes MAP sont utilisées autrement. Dans le premier cas, le travail de thèse a abouti à la spécification complète d'un moteur d'exécution de cartes MAP. Cette spécification combine un aspect déclaratif et formel, et un aspect concret sous la forme d'une architecture de bases de données. Dans le second cas, la thèse propose une approche qui recourt à la modélisation événementielle pour spécifier la sémantique d'un langage. Le méta-modèle obtenu est ensuite transformé en une architecture d'outil logiciel pour exécuter des instances du langage (c.-à-d. un moteur d'exécution). Cette approche est appliquée sur les cartes MAP et constitue une approche complémentaire et générique par rapport au travail de thèse de M-H. Edme.

Contexte et période	Problématique	Approche de solution et contributions
Co-encadrement de la thèse de Marc-Henri Edme (2002-2005)	<ul style="list-style-type: none"> <li>▪ Modéliser les processus d'usage et la variabilité sous-jacente</li> <li>▪ Quels mécanismes de contrôle pour guider l'utilisateur</li> <li>▪ Quelle architecture d'exécution pour supporter ces mécanismes</li> </ul>	<ul style="list-style-type: none"> <li>▪ Modélisation intentionnelle du processus d'usage avec une carte MAP</li> <li>▪ Définition d'une logique de navigation et d'exécution d'une carte MAP</li> <li>▪ Formalisation partielle de la logique d'exécution avec une notation mathématique</li> <li>▪ Spécification et implémentation d'une architecture d'exécution</li> </ul>
Co-encadrement de la thèse CIFRE de Hamid El Ghazi (2007-2009)	<ul style="list-style-type: none"> <li>▪ Comment intégrer la traçabilité dans les projets d'ingénierie</li> <li>▪ Comment identifier et modéliser les liens de traçabilité selon les besoins d'un projet</li> <li>▪ Comment guider l'utilisateur dans la gestion de la traçabilité dans un projet</li> </ul>	<ul style="list-style-type: none"> <li>▪ Proposition de l'approche MV-TMM avec un méta-modèle général de traçabilité</li> <li>▪ Intégration d'une technique d'optimisation pour sélectionner le modèle adéquat de traçabilité</li> <li>▪ Modélisation de la démarche d'utilisation</li> <li>▪ Application sur un cas d'entreprise</li> </ul>
Co-encadrement de la thèse de Kadan Aljoumaa (2009-2011)	<ul style="list-style-type: none"> <li>▪ Dans la continuité de travaux antérieurs sur les services web intentionnels (i-service et iSAO), quelle architecture technique pour les décrire ?</li> <li>▪ Quels processus de publication et de recherche pour garantir une utilisation optimale des i-services ?</li> </ul>	<ul style="list-style-type: none"> <li>▪ Définition d'un descripteur étendu d'un service intentionnel (adopté des travaux sur YASA4WSDL)</li> <li>▪ Modélisation avec MAP et guidage du processus de publication d'un i-service</li> <li>▪ Modélisation avec MAP et guidage du processus de formulation de requête</li> </ul>
Co-encadrement de la thèse de Sana Mallouli (2008-2014)	<ul style="list-style-type: none"> <li>▪ Les techniques de méta-modélisation concernent la structure statique d'un modèle, comment prendre en compte sa sémantique opérationnelle ?</li> <li>▪ Comment dériver un outil d'exécution à partir de cette spécification ?</li> </ul>	<ul style="list-style-type: none"> <li>▪ Proposition d'une approche de méta-modélisation événementielle pour capturer la sémantique d'un modèle</li> <li>▪ Définition d'une démarche de type IDM pour générer l'architecture d'un moteur d'exécution</li> <li>▪ Illustration de l'approche avec le modèle MAP</li> </ul>

Tableau I-1. Résumé des principaux travaux de recherche menés en collaboration avec le CRI

Ces deux travaux de thèse (M-H. Edme et S. Mallouli) font suite à mon intérêt particulier pour les problématiques de définition de langages et de spécification d'outils logiciels associés. Ils constituent une extension et une continuité de mon travail de thèse. En effet, ma thèse de doctorat portait sur la méthode Remora et la modélisation événementielle, et j'avais contribué à la réalisation d'un environnement de spécification et de prototypage basé sur cette méthode [7]. De plus, dès les débuts d'utilisation des cartes MAP, j'ai contribué à la définition précise, quoi que informelle, de la sémantique d'exécution du MAP [publi#62]. Enfin, j'ai aussi encadré plusieurs mémoires de Master recherche sur des sujets connexes à ces questions [221], [253], [100].

Pour ces différentes raisons, je vais consacrer l'intégralité d'un chapitre de ce mémoire (cf. chapitre III) pour présenter une synthèse de ces travaux qui relèvent de la problématique générale d'exécution de modèles. Les deux autres travaux de co-encadrement de thèse ne seront pas, par contre, présentés en détail dans ce mémoire.

### Génie logiciel empirique et analyse des rapports d'erreurs logiciels

Pour diversifier et enrichir mes expériences de recherche, j'ai effectué un séjour sabbatique à l'université de Lund au sein du groupe SERG<sup>8</sup>. Ce groupe de recherche est reconnu internationalement pour son expertise dans la recherche empirique en génie logiciel. La collaboration avec cette équipe s'est concrétisée dans un travail qui porte sur une problématique de maintenance des systèmes logiciels : le traitement des erreurs et la prédiction du temps nécessaire pour la résolution d'une erreur logicielle. Le travail réalisé exploite les rapports textuels d'erreur, et fait appel aux techniques de fouille de texte ("text mining") et d'apprentissage machine non supervisé ("unsupervised machine learning") ; il a été publié dans une revue de référence du domaine du génie logiciel empirique [publi#02]. Il sera présenté en détail dans le chapitre IV de ce mémoire.

<sup>8</sup> SERG : Software Engineering Research Group, Lund Technical University, Sweden (<http://serg.cs.lth.se/>)

## I.4.2 Thème 2 : usages et adoption des artefacts informatiques et sociotechniques

L'utilisation de l'artefact technologique dans les organisations et dans la société est un phénomène complexe à multiples facettes [172]. Il est construit pour s'intégrer dans un contexte d'usage et selon les besoins des acteurs, mais aussi, il influence et modifie ce contexte et ces besoins, et devient le vecteur d'une force interne ayant le potentiel de remodelé l'organisation et la société (cf. §1.2.4). Les recherches que j'ai menées à ce propos ont été réalisées au sein de l'équipe de recherche en SI dans mon institution de rattachement, Télécom Ecole de Management (ex. INT Gestion). Elles ont porté sur deux domaines spécifiques d'usage : l'informatique dans le contexte de l'administration publique, et l'enseignement en ligne. Le Tableau I-2 présente les travaux de recherche que j'ai menés sur ce sujet.

Contexte et période	Problématique	Résultats et contributions
Projet ProAdmin et prolongement (2005-2014)	<ul style="list-style-type: none"> <li>▪ Quels sont les freins à l'adoption des plateformes d'achat public ? (en 2005-06)</li> <li>▪ Quelles problématiques et perspectives d'évolution pour l'e-gouvernement en France? (en 2010)</li> </ul>	<ul style="list-style-type: none"> <li>▪ Exploration des pratiques d'usage des plateformes d'achat public</li> <li>▪ Enquête auprès d'experts sur les perspectives d'évolution pour l'e-gouvernement en France</li> <li>▪ Publication de trois ouvrages (co-édités)</li> </ul>
Co-encadrement de la thèse d'Ana-Lidya Franzoni (2005-2009)	<ul style="list-style-type: none"> <li>▪ Est-il pertinent de prendre en compte les styles d'apprentissage d'un apprenant dans un système d'enseignement en ligne ?</li> <li>▪ Comment tirer profit de l'évolution des TIC et développer une pédagogie adaptée et personnalisée qui exploite les nouveaux médias électroniques (e-médias) ?</li> </ul>	<ul style="list-style-type: none"> <li>▪ Taxonomie des e-médias</li> <li>▪ Grille de correspondance triptyque : style d'apprentissage – stratégie pédagogique – e-média</li> <li>▪ Système expérimental d'enseignement en ligne</li> <li>▪ Expérimentation avec plusieurs groupes d'étudiants en informatique</li> </ul>
Co-encadrement de la thèse de Yaneli Cruz (20011-2014)	<ul style="list-style-type: none"> <li>▪ Expliquer l'adoption de l'enseignement en ligne mobile ("Mobile Learning")</li> <li>▪ Est-ce que les styles d'apprentissage d'un apprenant impactent l'usage et l'adoption du <i>Mobile Learning</i> ?</li> </ul>	<ul style="list-style-type: none"> <li>▪ Modèle corrélatif (c.-à-d. sous formes de variables corrélées statistiquement) pour expliquer l'adoption du <i>Mobile Learning</i></li> <li>▪ Quasi-expériences contrôlées avec plusieurs groupes d'étudiants en comptabilité</li> </ul>

Tableau I-2. Résumé des principaux travaux de recherche relatifs aux questions d'usage des artefacts technologiques

C'est dans le cadre d'un projet mené en 2005-2006 et financé par la fondation Louis Leprince-Ringuet (actuellement *Fondation Télécom*) que j'ai été amené à m'intéresser aux problématiques d'e-administration. Le projet ProAdmin<sup>9</sup> s'intéresse à ce qu'on appelle « la dématérialisation des marchés publics » (ou DEMAP), et qui consiste à remplacer progressivement les supports physiques par des supports électroniques pour réaliser les transactions d'achat dans les organismes publics. Décrété officiellement au 1<sup>er</sup> janvier 2005, la DEMAP est un chantier gigantesque pour moderniser les systèmes d'information publics. Etant donnée l'envergure du projet (toutes les institutions publiques) et la disparité des moyens dont disposent les multiples acteurs concernés (mairies, ministères, hôpitaux, universités, etc.), il soulève des problématiques complexes et sur une large échelle.

Les travaux réalisés dans le projet ProAdmin ont consisté à analyser les procédures d'achat telles qu'elles sont définies dans les textes de loi [publi#24, #23, #20], et à évaluer – sommairement – les principales plateformes d'achat public proposées à l'époque [publi#22, #10]. J'ai continué par la suite à observer le domaine du e-gouvernement, et j'ai mené une étude sur les perspectives d'évolution du domaine [publi#36].

L'autre volet de mes travaux dans ce thème porte sur l'enseignement en ligne. Une théorie importante dans les sciences de l'éducation est qu'il y a des manières différentes d'apprendre et d'assimiler les connaissances. Ces « manières » sont désignées comme étant des « styles d'apprentissage », et leur intérêt réside dans le fait qu'il serait potentiellement plus efficace d'accommoder et d'adapter le contenu et la pédagogie d'enseignement au style d'apprentissage de l'apprenant [129], [137]. Une première difficulté réside dans la définition, la caractérisation et l'identification des styles d'apprentissage chez les apprenants ; plusieurs approches existent et sont sujettes à de multiples débats dans le domaine des sciences de l'éducation [38]. La seconde difficulté concerne la personnalisation des

<sup>9</sup> Pour une description plus détaillée du projet, voir : <http://www-public.tem-tsp.eu/~assar/proadmin/>

supports et des pédagogies d'enseignement de manière à s'adapter aux multiples styles d'apprentissage d'une population d'apprenants. Une troisième question concerne l'évaluation de la pertinence et l'efficacité potentielles de cette approche, et l'analyse des facteurs susceptibles de l'influencer (p.ex. matière enseignée, âge, etc.).

Sur ces problématiques d'e-learning et de styles d'apprentissage, j'ai co-encadré deux thèses avec deux doctorantes rattachées à une institution d'enseignement supérieur au Mexique (ITAM<sup>10</sup>). Dans la thèse d'A-L. Franzoni, le résultat principal est une taxonomie triptyque qui relie les styles d'apprentissages aux médias électroniques et aux pédagogies d'enseignement (cf. Tableau I-2), et dont la publication suscite un nombre élevé de citations [publi#55]. Dans la thèse de Yaneli Cruz, nous nous sommes penchés sur les nouvelles possibilités qu'offre l'informatique mobile et l'impact potentiel des styles d'apprentissage. Le principal résultat est un modèle théorique de l'adoption [publi#28].

## I.5 Organisation du mémoire

La suite de ce mémoire est organisée en cinq chapitres principaux. Ces chapitres ne couvrent pas l'intégralité de mes travaux de recherche telle que je viens d'en donner un aperçu dans la section précédente. J'ai choisi d'abord d'aller un peu plus loin dans l'analyse de la multidisciplinarité, et d'évoquer des travaux récents – non mentionnés dans la section précédente – et qui portent sur des questions d'ordre méthodologique, c'est l'objet du chapitre II.

Ensuite, j'ai choisi de me focaliser sur les travaux qui reflètent aux mieux la multidisciplinarité telle que je la développe dans ce mémoire, et qui montrent les contributions qui me semblent les plus significatives. Ainsi, les chapitres III et IV appartiennent au thème 1 (modèles et méthodes d'ingénierie des SI), l'un relève des sciences de conception et l'autre relève des sciences des faits. Le chapitre V appartient au thème 2 (usage et impact des artefacts technologiques) et relève aussi des sciences des faits.

### Chapitre 2 : A propos des méthodes en sciences et en ingénierie

Le thème central de ce chapitre est la notion de « méthode ». Ce chapitre poursuit la discussion initiée dans l'introduction sur l'analyse de la multidisciplinarité en approfondissant le point de vue méthodologique. Le premier objectif de ce chapitre est donc de présenter une brève synthèse des différentes significations qui sont attachées au terme « méthode » selon les trois facettes des sciences informatiques. Comme l'utilisation d'une méthode aboutit à un *résultat*, il s'agit aussi d'évoquer la nature de ce résultat selon le sens attaché au terme « méthode ». Le second objectif du chapitre est d'évoquer mes travaux les plus récents qui concernent, directement ou indirectement, certaines des questions méthodologiques traitées dans ce chapitre (p.ex. l'ingénierie des méthodes, les méthodes créatives pour l'ingénierie des besoins, les méthodes empirique et expérimentales en ingénierie, etc.).

### Chapitre 3 : L'exécution de modèles conceptuels

Le sujet central de ce chapitre est une problématique importante en ingénierie : comment faire pour qu'un modèle d'un système puisse s'exécuter dans un environnement informatique ? Ce thème de recherche est une extension de mon travail initial de thèse de doctorat réalisé dans le cadre du projet Rubis et du langage de spécification exécutable Proquel. J'ai élargi le champ de recherche en explorant la problématique de l'expression de la sémantique opérationnelle d'un langage de modélisation. En effet, pour qu'un modèle puisse être exécuté, il faut définir le plus précisément possible sa sémantique. Celle-ci est généralement difficile à capturer, et on se contente de l'exprimer en langage naturel ou pseudo-formel. Néanmoins, pour construire un outil d'exécution de modèles, et en particulier si l'on souhaite automatiser cette construction, il faut définir explicitement la sémantique du modèle. Ce chapitre est basé sur les travaux réalisés dans deux thèses co-encadrées au CRI (cf. Tableau I-1).

---

<sup>10</sup> ITAM : Instituto Tecnológico Autónomo de México

**Chapitre 4 : Analyse des rapports d'erreurs logicielles et prédiction du temps de résolution**

Ce travail est le résultat d'une collaboration avec des chercheurs de l'université de Lund, Suède. Il comporte deux volets : le premier est la réplication d'une expérience antérieure ([182], publiée en ligne en 2012), le second est la mise à l'épreuve d'une hypothèse formulée par l'auteur de l'expérience répliquée (cf. §1.4.2). Ce travail exploite les rapports textuels d'erreur, et fait appel aux techniques de fouille de texte et d'apprentissage machine non supervisé.

**Chapitre 5 : Enseignement assisté par les TIC et prise en compte des styles d'apprentissage**

Les thèmes centraux de ce chapitre sont l'enseignement en ligne et l'impact potentiel des styles d'apprentissage. Les travaux qui y sont présentés concernent deux thèses co-encadrées dans mon institution de rattachement, Télécom École de Management (ex. INT École de gestion). Ils font appel aux théories de l'apprentissage et de l'impact potentiel des différences individuelles, appelées ici « styles d'apprentissage » ; ils font aussi appel aux théories d'adoption des artefacts (px. TAM, UTAUT). Un modèle théorique de l'adoption de l'enseignement mobile y est proposé (cf. Tableau I-2).

**Le chapitre 6 : Conclusion générale**

Ce chapitre conclut le mémoire et présente une synthèse de mes principales contributions, notamment en ce qui concerne la multidisciplinarité, et évoque plusieurs pistes et chantiers pour des recherches futures.

## II. Chapitre II : A propos des méthodes en science et en ingénierie

### Sommaire

<b><u>II.1</u></b>	<b><u>Préambule</u></b>	<b>22</b>
<b><u>II.2</u></b>	<b><u>Méthode(s) dans les sciences du formel</u></b>	<b>22</b>
<u>II.2.1</u>	<u>Méthodes du formel et étude du logiciel et du SI</u>	23
<b><u>II.3</u></b>	<b><u>Méthode(s) dans les sciences des faits</u></b>	<b>23</b>
<u>II.3.1</u>	<u>L'expérimentation et la méthode hypothético-déductive</u>	24
<u>II.3.2</u>	<u>Qualitatif versus quantitatif</u>	24
<u>II.3.3</u>	<u>Inductif versus déductif</u>	25
<u>II.3.4</u>	<u>Récapitulatif : la dualité du « fait » empirique</u>	25
<b><u>II.4</u></b>	<b><u>Méthode(s) de conception</u></b>	<b>26</b>
<u>II.4.1</u>	<u>Processus de conception, modélisation et exploration</u>	26
<u>II.4.2</u>	<u>Le "design scientifique" et la "science du design"</u>	27
<b><u>II.5</u></b>	<b><u>Le "Design Science" dans le domaine des SI</u></b>	<b>28</b>
<b><u>II.6</u></b>	<b><u>Mes contributions aux questions méthodologiques en informatique et SI</u></b>	<b>30</b>
<u>II.6.1</u>	<u>Ingénierie des méthodes : synthèse sur l'adaptabilité et l'adaptation</u>	30
<u>II.6.2</u>	<u>IDM et ingénierie des besoins : synthèse des travaux d'un Workshop</u>	32
<u>II.6.3</u>	<u>Créativité et ingénierie des besoins : une revue de la littérature</u>	33
<u>II.6.4</u>	<u>Synthèse sur les méthodes empiriques en ingénierie des SI</u>	33
<u>II.6.5</u>	<u>Synthèse sur les méthodes de revue de la littérature</u>	35
<u>II.6.6</u>	<u>Réplication d'une étude en génie logiciel empirique</u>	35
<u>II.6.7</u>	<u>Théorisation dans la recherche empirique sur les ERP : études exploratoires</u>	36
<b><u>II.7</u></b>	<b><u>Conclusion du chapitre</u></b>	<b>38</b>

## II.1 Préambule

Le thème central de ce chapitre est la notion de « méthode ». Le terme « méthode » renvoie à un sens général qui désigne la manière de mener une activité, de faire quelque chose. L'étude des méthodes scientifiques relève de *l'épistémologie*, qu'on peut définir comme **l'analyse rigoureuse des discours scientifiques** [12]. C'est une branche de la philosophie qui examine les modes de raisonnements scientifiques et les structures formelles des théories mises en œuvre et développées. L'étude des méthodes scientifiques est la tentative d'identifier et définir les activités qui garantissent le succès de la recherche scientifique [6]. Certaines activités sont universellement considérées comme caractéristiques fondamentales de la méthode scientifique, telles que l'observation et l'expérimentation systématiques, les raisonnements inductifs et déductifs, la formation et le test d'hypothèses et de théories, etc. ; néanmoins, de grandes variabilités existent dans la manière de conduire ces activités [6]. Malgré les définitions multiples qu'on peut lui accorder, la méthode scientifique reste, selon le philosophe des sciences M. Bunge, le seul moyen de démarcation entre la science et la "pseudo-science" ([34], p.36).

Ce chapitre poursuit la discussion initiée dans l'introduction sur l'analyse de la multidisciplinarité en approfondissant le point de vue méthodologique. Le premier objectif de ce chapitre est donc de présenter **une brève synthèse des différentes significations qui sont attachées au terme « méthode »** selon les trois facettes des sciences informatiques. Comme l'utilisation (ou l'application) d'une méthode aboutit à un *résultat*, il s'agit aussi d'évoquer **la nature de ce résultat** selon le sens attaché au terme « méthode ». Ces deux aspects, c.-à-d. la méthode et la nature du résultat, mettent en évidence une facette essentielle de la multidisciplinarité en informatique et en SI. Ils contribuent fortement à clarifier les méthodes utilisées dans mes travaux exposés dans le reste de ce mémoire. L'étude des méthodes étant un sujet très vaste, je vais brièvement présenter quelques principaux éléments et brosser une image d'ensemble de la question des méthodes en science et ingénierie.

Le second objectif du chapitre est d'évoquer **mes travaux les plus récents** qui portent, directement ou indirectement, sur des questions liées à la notion de « méthode ». Les thèmes suivants y seront abordés :

- pour les méthodes de conception, ce sont l'ingénierie des méthodes, l'ingénierie dirigée par les modèles, la créativité dans les méthodes d'ingénierie des besoins ;
- pour les méthodes de recherche, ce sont les méthodes empiriques et expérimentales en ingénierie, les méthodes de revue de la littérature, la question de la réplication d'une expérience empirique, et la problématique de construction de théories dans les sciences de faits.

## II.2 Méthode(s) dans les sciences du formel

Les sciences du formel désignent les mathématiques et tout autre domaine de connaissance qui est directement basé sur celles-ci, et dont l'informatique théorique (ou informatique formelle) en fait partie. Les mathématiques appliquent des **raisonnements logiques** à divers objets abstraits (nombres, figures, transformations, etc.) pour construire par **déduction et démonstration** des connaissances abstraites. La validité de ces connaissances repose intégralement sur la validité logique des démonstrations, et dépend d'un certain nombre de postulats de départ et d'axiomes fondamentaux. Ainsi, les mathématiques existent et se développent quasi indépendamment de toute observation ou expérience dans le monde réel<sup>11</sup>.

Les « méthodes » mathématiques, basées sur **l'abstraction, le calcul et la démonstration logique**, sont considérées comme infaillibles (dans la limite de l'acceptation des postulats et axiomes initiaux). La **mathématisation** d'un problème est donc une approche universelle pour raisonner sur celui-ci, et espérer obtenir une réponse dont la validité découle directement de la validité de la méthode mathématique ([34], p.468). Les mathématiques jouent de ce fait un rôle crucial dans le développement des sciences ; néanmoins, exprimer formellement un problème et y appliquer des raisonnements mathématiques ne garantit pas la véracité des résultats obtenus (Ibid., p.477)<sup>12</sup>.

<sup>11</sup> Néanmoins, l'étude de certains problèmes du monde réel aboutit parfois à de nouveaux développements mathématiques.

<sup>12</sup> Illustration avec « la théorie axiomatique des fantômes » (Bunge, M., "Scientific Research I", Springer, 1967, p.477).

L'informatique théorique étudie les fondements logiques et mathématiques de l'informatique, ses objectifs ne sont pas toujours directement reliés à des enjeux technologiques. Elle est de ce point de vue assez similaire à la physique théorique, où les méthodes mathématiques sont exploitées en profondeur pour développer de nouvelles connaissances sur les sujets fondamentaux de la discipline. En informatique, les sujets fondamentaux sont l'algorithme et le programme, ainsi que le langage de programmation indispensable pour écrire le programme. D'après [56], les principaux sujets de l'informatique formelle sont la calculabilité, les grammaires et les automates, la complexité du calcul algorithmique, et la sémantique formelle des langages de programmation.

### II.2.1 Méthodes du formel et étude du logiciel et du SI

Quel champ d'application des méthodes et connaissances de l'informatique théorique pour l'étude du logiciel et l'artéfact sociotechnique (le SI) ? Le logiciel est un artéfact dont la complexité dépasse largement celle d'un programme informatique, et contrairement à celui-ci, il ne peut pas être réduit à une expression de type algorithmique, notamment lorsqu'on considère l'interface homme-machine. Il est ainsi très vite apparu qu'on ne dispose d'aucun outil formel pour exprimer à un haut niveau d'abstraction ce que fait un logiciel, outil qui serait aussi flexible que sont les langages algorithmiques ou de programmation pour spécifier un algorithme ou un programme ([240], p.325). A titre d'exemple, les trois volumes de D. Knuth sur la programmation<sup>13</sup> ont posé de solides fondements pour l'étude du programme informatique ; rien de semblable n'est disponible aujourd'hui lorsqu'il s'agit de construire et spécifier les logiciels complexes utilisés dans les SI des organisations. Ce constat pose la limite des méthodes formelles car la construction d'un logiciel, ou d'un SI en surcroît, relève d'une problématique **d'ingénierie**, irréductible à des raisonnements de type mathématique, et où la réponse se construit selon une démarche de **conception**.

Néanmoins, beaucoup d'approches méthodologiques en génie logiciel et ingénierie des SI explorent et exploitent les méthodes formelles, c'est le cas par exemple des diagrammes d'état pour une spécification visuelle [97], les réseaux de Petri pour spécifier un flux de travail (Workflow) [68], la théorie et la logique des ensembles pour la modélisation conceptuelle [234], la théorie des agents autonomes en tant que méthode de développement [28], etc. ; sans oublier les méthodes formelles de spécification telle que Z [181]. Enfin, le courant de l'ingénierie dirigée par les modèles (cf. §II.6.2) requiert un certain niveau de formalisation pour exprimer la transformation des modèles avec une précision et une rigueur similaires à celles qui prévalent dans la théorie des langages et les outils de compilation.

## II.3 Méthode(s) dans les sciences des faits

Selon le philosophe des sciences Mario Bunge, les principes de la méthode scientifique sont universels ([34], p.9); sa trame fondamentale est présentée dans la Fig. II-1. Cette méthode est associée aux sciences des faits, et en premier lieu avec les sciences de la nature (p.ex. la physique), où l'objectif est de **comprendre** et **d'expliquer** des phénomènes et de découvrir les lois qui les gouvernent. Elle sous-entend une vision rationaliste et positiviste de la démarche scientifique, où l'on postule que les liens de causalité existent indépendamment des acteurs humains, et que le chercheur peut les découvrir et les formuler dans des lois. Elle recourt à ce qu'on appelle communément la **méthode hypothéti-co-déductive** [6] : l'analyse d'un problème et la formulation d'une question de recherche amènent le chercheur à interroger le corpus des connaissances disponibles et à formuler des hypothèses en tant qu'explications possibles (et plausibles) aux questions posées ([34], cf. Fig. II-1).

<sup>13</sup> D. Knuth, *The art of computer programming*, Vol. I, II, III, 1968-1973.

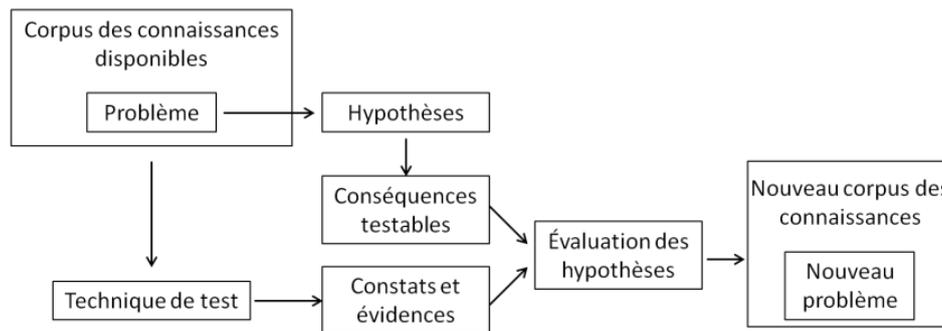


Fig. II-1. Le cycle de la recherche scientifique (d’après M. Bunge dans [34], p.9)

### II.3.1 L’expérimentation et la méthode hypothético-déductive

Pour évaluer les conséquences des hypothèses émises, on les soumet au contrôle de l’expérience. L’expérience scientifique est une mise à l’épreuve, rigoureuse et systématique, de quelque chose pour en vérifier les propriétés [139]. Elle requiert l’élaboration de techniques, instruments et protocoles pour tester les conséquences attendues ; elle nécessite la définition de mesures et la récolte de données, traces et observations dont l’analyse va **confirmer** ou **infirmer** la véracité des hypothèses émises. Et aboutir ainsi à l’enrichissement du corpus des connaissances par des réponses, partielles ou complètes, au problème posé.

La Fig. II-2 dresse le synopsis d’une expérimentation : le modèle d’investigation relie les variables et exprime les hypothèses du chercheur. Ces liens de dépendance, de corrélation, ou de cause à effet entre variables définissent ainsi un modèle d’investigation – théorique ou empirique – dans lequel les variables dépendantes correspondent à la finalité de l’étude. Comme le chercheur dispose d’un certain niveau de contrôle sur l’environnement de l’expérience (lieu, acteurs, contexte, mesures, etc.), et grâce au jeu des hypothèses et du modèle d’investigation, il espère comprendre le phénomène étudié et découvrir des relations entre les variables de son modèle, et ainsi, émettre ou valider une connaissance théorique nouvelle. Néanmoins, le chercheur n’ayant pas le contrôle sur l’intégralité de l’environnement, des **facteurs confondants** peuvent influencer sur les observations et remettre en cause la validité interne de l’expérience. L’identification de ces facteurs et l’analyse de leur effet potentiel est un élément clé de toute expérimentation.

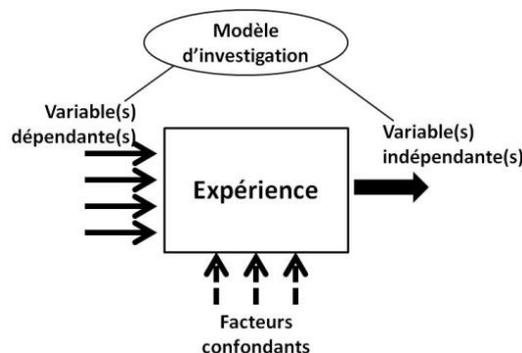


Fig. II-2. Synopsis d’une expérimentation

### II.3.2 Qualitatif versus quantitatif

La méthode hypothético-déductive, théorisée au 19<sup>e</sup> et début du 20<sup>e</sup> siècle, a eu – et continue d’avoir – un impact profond sur les sciences modernes, toutes disciplines confondues [12]. Elle suppose néanmoins l’existence d’un corpus préalable de connaissances pour fournir une explication logique susceptible d’être confrontée expérimentalement à la réalité empirique. Elle suppose aussi que le phénomène étudié peut être exprimé avec des mesures et des modèles quantitatifs, comme c’est le cas dans les sciences physiques. Dans le cas des sciences humaines, telles que la sociologie ou la psychologie, un corpus préalable de connaissances peut ne pas exister lorsque l’on étudie un phéno-

mène peu ou totalement méconnu (p.ex. le suicide chez l'adolescent ou les troubles du langage chez l'enfant). De plus, le phénomène étudié ne peut pas être réduit uniquement à des mesures et des modèles quantitatifs.

Le développement de ces sciences a ainsi donné lieu à d'autres visions des méthodes de recherche où l'observation minutieuse des faits empiriques et le recueil systématique de traces jouent un rôle prépondérant. Ces données, souvent subjectives, portent sur des aspects spécifiquement humains tels que la perception, la compréhension, l'attitude ou les convictions. Ces méthodes postulent que la connaissance est construite par l'acteur humain et qu'elle est sujette à **l'interprétation** autant par les acteurs du phénomène que par le chercheur lui-même. L'interprétation de ces données permet ainsi de construire des descriptions et des explications aux phénomènes étudiés. Ces méthodes, qualifiées généralement de « qualitatives » [64], permettent des descriptions riches et multiples de phénomènes complexes en traçant des événements et des situations uniques ou inattendues [220]. La connaissance obtenue est cependant descriptive et narrative ; l'explication du phénomène ne s'exprime pas toujours avec un lien clair de cause à effet, mais à l'aide d'un réseau complexe d'analyses et d'arguments conceptualisés. Pour baser l'interprétation sur des fondements rigoureux et systématiques, l'épistémologie des méthodes qualitatives fait appel – entre autres – à l'herméneutique (c.-à-d. l'analyse contextualisée des textes écrits) et à l'ethnographie (c.-à-d. l'enquête sociale sur le terrain et l'immersion dans le groupe étudié) [142].

### II.3.3 Inductif versus déductif

Les méthodes de recherche qualitatives sont-elles compatibles avec la méthode hypothético-déductive ? Quelle validité, quel niveau de **généralisabilité** peut-on accorder aux résultats des méthodes qualitatives ? Ces questions suscitent des débats qui dépassent largement le cadre de ce mémoire (p.ex. [64], [110]). Pour ma part, je retiens que même si le principe de l'induction ne permet pas de construire une connaissance scientifiquement valide (c.-à-d. on ne peut pas émettre un lien de corrélation directe ou une loi de cause à effet suite à une seule série d'observations concordantes), l'observation et l'analyse des données permettent d'abstraire et de conceptualiser les faits observés et de produire une connaissance à portée limitée (c.-à-d. très contextualisée). Cette connaissance, obtenue par un raisonnement inductif, apporte un éclairage précieux et indispensable pour **explorer** un phénomène nouveau ou peu connu. Elle peut alors prendre la forme d'un **modèle empirique** susceptible de devenir **une hypothèse évaluable expérimentalement**, dans une approche qui combine les méthodes qualitatives et quantitatives.

### II.3.4 Récapitulatif : la dualité du « fait » empirique

Après ce bref exposé, on peut dire que les méthodes de recherche dans les sciences des faits dépendent de la nature intrinsèque des faits analysés. Dans les sciences de la nature (la physique en premier lieu), les faits sont **objectivement mesurables**, et l'explication des phénomènes étudiés peut être capturée dans des modèles candidats à l'expérimentation. Dans les sciences humaines, une partie importante des faits étudiés est **subjective** avec un sens variable selon les acteurs humains et leurs contextes culturels, économiques, intellectuels, etc. Cette question de la **construction de connaissance en recourant à l'interprétation** reste sujette à débat en épistémologie, et fait qu'un courant important de recherche dans les sciences humaines fait le choix quasi-exclusif de l'analyse quantitative des données. C'est ce qu'on appelle communément le « behaviorisme » en psychologie et en sociologie ([141], p.41).

Pour ce qui est de l'informatique et des SI, les faits associés aux artefacts sont de nature autant **objective** (p.ex. nombre de lignes de code) que **subjective** (p.ex. facilité perçue d'apprentissage d'un langage). Toutes les méthodes de recherche issues des sciences de la nature et des sciences humaines peuvent alors potentiellement être mobilisées. Je reviendrai sur ce point dans la section §II.6.4 en évoquant la coordination d'un numéro spécial à ce propos.

## II.4 Méthode(s) de conception

D'après S. Vial, le concept de « design » est lié à celui de « projet », ([231], p.11). On attribue l'invention du concept à l'architecte Brunelleschi à la Renaissance italienne vers 1420 (Ibid., p.13). Le design est ainsi rattaché au projet architectural, où la notion de projet correspond à « une méthodologie de l'anticipation de l'œuvre à réaliser » (Ibid., p.15). Elle correspond aussi à l'introduction d'une forme nouvelle de division du travail : celle de la conception d'une part, et celle de la réalisation (ou production) d'autre part. En français, le « design » est recouvert avec les termes **dessein** (intention, but, visée) et **dessin** (image, figure, croquis). Ces deux sens voisins de *dessein intériorisé* et de *dessin extériorisé* se retrouvent dans l'italien « **disegno** » comme dans l'anglais « **design** » (Ibid., p.14). L'étymologie du terme « design » met ainsi en évidence deux propriétés intrinsèques de l'activité de conception :

- L'explicitation de **l'intention et des objectifs** de celui qui initie et souhaite la réalisation d'un projet est déterminante pour la réalisation de celui-ci ; néanmoins, l'intention peut être vague, partielle, incomplète, et l'activité de conception consiste aussi à clarifier cette intention et à **explorer** les champs du possible pour amener l'initiateur du projet à choisir et fixer sa finalité.

- Un projet de conception passe inévitablement par la création d'une (ou plusieurs) **représentation(s)** de l'artéfact. Cette **modélisation** est indispensable pour anticiper l'artéfact qu'on veut construire, elle est au centre de la division du travail entre conception et production. C'est aussi le support pour communiquer avec l'initiateur du projet et explorer les champs du possible.

### II.4.1 Processus de conception, modélisation et exploration

Comment se fait concrètement l'activité de conception ? Dans les multiples domaines où elle s'exerce, la conception a pendant longtemps été considérée comme un savoir qui ne peut se transmettre que par **la pratique et l'apprentissage** sur le terrain auprès de praticiens et experts de la conception. Parce que le résultat de la conception est visible et montrable (modèles, maquettes, prototypes ou produits finis), il est plus simple et plus commode d'évoquer ce résultat (ainsi que les différentes étapes successives par lesquelles il a transité) que de chercher à **explicitier l'activité de conception** elle-même. C'est ce qu'on qualifierait aujourd'hui de vision **orientée produit** du processus de conception (cf. §II.6.1).

Néanmoins, l'activité de conception est-elle assimilable à une démarche systématique, fondée sur l'objectivité et la rationalité, c.-à-d. une **méthode** qu'on pourrait exprimer avec précision et enseigner avec rigueur ? On considère que c'est en 1962 que l'activité de conception est devenue un sujet d'étude scientifique à part entière à travers ce qu'on a appelé le mouvement des « Design methods » formé par des praticiens et chercheurs en architecture et design industriel [231], [49]. L'œuvre de H. Simon sur « les sciences de l'artificiel » a fortement contribué à susciter l'intérêt pour ce courant de réflexion et de recherche [216].

Il y a de ce fait deux manières d'aborder la question de la méthode en design ([231], p.84). La première consiste à penser le design comme **une activité de nature quasi-artistique**, proche de l'artisanat en tant que pratique, certes descriptible et répétable, mais qui tend à valoriser l'intuition et l'inspiration créatrice du concepteur. La seconde manière consiste à penser le design comme **un acte de projet** fondé sur des approches rationnelles et pragmatiques. L'activité de conception est ainsi vue comme **une démarche méthodique de résolution de problèmes** inspirée des préceptes de la démarche scientifique. Il faut néanmoins signaler qu'une subtile divergence s'est créée entre les tenants d'une vision **machiniste**, inspirée par la cybernétique et l'intelligence artificielle, portée par H. Simon en particulier ; et les tenants d'une vision **phénoménologique**, plus soucieux de la complexité de la dimension humaine dans l'acte de conception, qui doutent de l'intérêt et la faisabilité de fixer cet acte dans un cadre rigide et formel (Ibid, p.86), et qui restent donc attachés aux visions artisanales, créatives et intuitives de l'acte de conception.

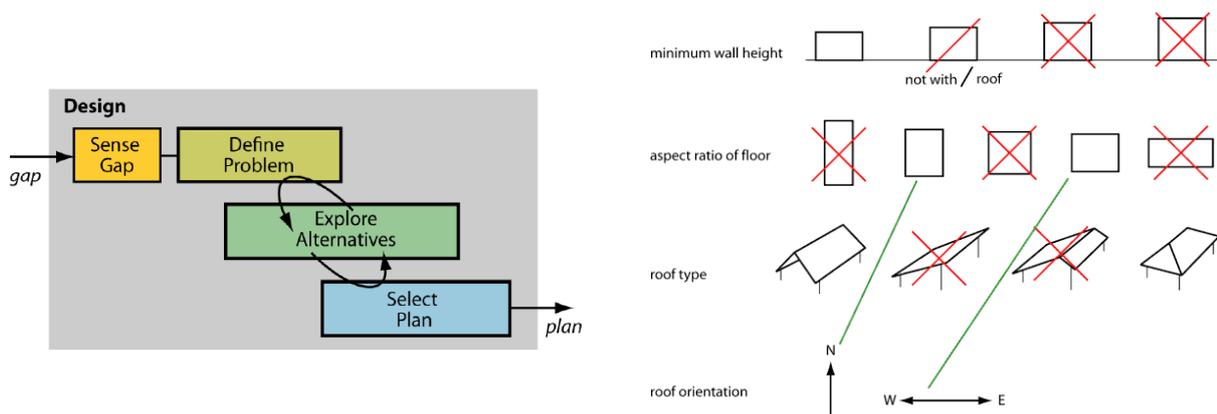


Fig. II-3. Synopsis d'un processus générique de conception (gauche), et droite, un exemple de l'exploration de l'espace des solutions pour la conception d'un abri de jardin facilement transportable (Source : [226], p.15 et p.46)

La Fig. II-3g présente le synopsis d'un processus générique de conception en tant que démarche structurée et rationnelle de résolution de problème [226]. L'exploration de l'espace des solutions est au centre de ce processus. La modélisation y joue un rôle essentiel : elle permet au concepteur de décrire des fragments de solution et les alternatives qu'elles autorisent (cf. Fig. II-3d). L'évaluation des alternatives et leurs confrontations avec l'énoncé du problème et la liste des besoins aboutissent à un cycle vertueux d'**analyse** et de **synthèse** qui va converger vers une (ou plusieurs) solution(s) adéquate(s). Le résultat final est un **plan d'action** sous forme de modèles, maquettes, prototypes, critères de choix, planification de projet, etc.

#### II.4.2 Le "design scientifique" et la "science du design"

Quels liens y'a-t-il – ou pourrait-il y avoir – entre les méthodes de conception et la science, c.-à-d. les connaissances scientifiquement obtenues, et les méthodes scientifiques de recherche ? C'est une question complexe, je vais mentionner trois remarques importantes et synthétiser mon point de vue :

- (i) Alors que la méthode scientifique produit une connaissance sur un phénomène dans le monde réel, la méthode de conception produit un plan d'action pour un artéfact. La valeur de la connaissance scientifique se mesure à l'aune de son **intérêt épistémique**, c.-à-d. sa contribution à comprendre et expliquer le monde ; la valeur d'un artéfact se mesure par son **intérêt pratique**, c.-à-d. sa contribution à satisfaire un besoin et résoudre un problème de manière efficace ([164], p.3-4). De prime abord, ces deux résultats sont radicalement différents.
- (ii) Les phases de génération et d'exploration de solutions exploitent les connaissances scientifiques pour imaginer, évaluer, combiner des artéfacts qui obéissent aux règles que la science a produites. C'est ce qu'on peut appeler la **« scientification » du design** ou le « design scientifique », et qui coïncide avec le *développement technologique* ([164], p.8). L'application de la connaissance scientifique est alors une **démarche innovante de conception** visant à satisfaire des besoins existants avec des artéfacts basés sur de nouvelles connaissances scientifique (p.ex. le scanner en médecine), ou pour proposer des artéfacts pour des usages nouveaux (p.ex. Twitter).
- (iii) A priori, cette « application » de la connaissance scientifique ne produit pas de nouvelles connaissances ; néanmoins, l'artéfact en lui-même contient une **connaissance de type structural ou architectural** (c.-à-d. sa construction propre et les principes qui la gouvernent) ; sa démarche de conception produit aussi une **connaissance sur le processus** de conception en termes de choix réalisés, de critères et techniques d'optimisation, etc.

Il en ressort que la conception d'artéfacts produit des connaissances de natures multiples, et qu'il y a une manière spécifique d'acquérir de la connaissance « en concevant », une manière a priori similaire à un savoir-faire acquis par la pratique ; cette manière spécifique de **produire de la connaissance à travers l'activité de conception** serait donc, d'après l'expert en *Design Studies* Nigel Cross, distincte des approches scientifiques et académiques [47],[48]. Une expérience a montré que les architectes résolvent un problème en générant rapidement et en les explorant des solu-

tions sous formes de dessins et maquettes, alors que les scientifiques étudient le problème en profondeur en cherchant à le formaliser et à le résoudre de manière systématique et rationnelle ([47], p.225).

Pour que la connaissance produite par la conception d'un artéfact puisse avoir le label « scientifique », il faut que sa production puisse être confrontée aux normes des méthodes scientifiques (cf. §II.3). Dans cette « science de conception », qui étudie donc autant **le processus** que **le produit** de la conception, l'impératif de **véracité scientifique** des connaissances produites porte sur *l'utilité* de l'artéfact, son *adéquation* avec le contexte d'usage (humain, technique, organisationnel, etc.), la *résolution concrète* du problème, la *structure interne* de l'artéfact par rapport à ses *propriétés externes* et sa *finalité*, ainsi que *la faisabilité* effective de sa production selon les contraintes de ressources (coûts, délais). Dans le domaine des systèmes d'information, c'est ce qu'on a appelé depuis une décennie le courant du « Design Science », c'est l'objet de la prochaine section.

## II.5 Le "Design Science" dans le domaine des SI

Selon Hevner et al. dans [104], l'objectif du *Design Science* est de rapprocher et combiner les recherches comportementales de type « usage et impact du SI » (cf. chapitre I), généralement conduites avec les méthodes des sciences de faits, avec des recherches en ingénierie des SI qui reposent sur la conception d'un artéfact. Cette intégration des deux univers prend la forme d'un cadre d'analyse (« framework ») et emprunte des idées issues des « sciences du design », où la connaissance est créée « en concevant », et où l'artéfact lui-même et son processus de conception sont sources et support de production de connaissances (cf. II.4.2). La Fig. II-4 présente ce cadre de référence.

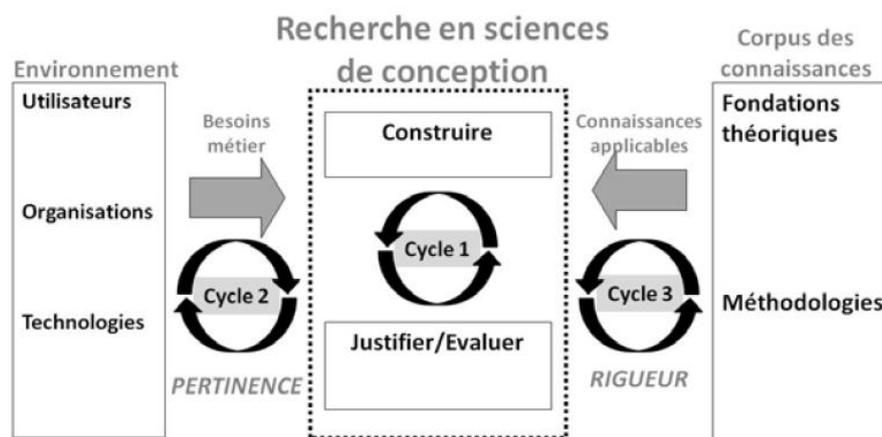


Fig. II-4. Cadre de référence de Hevner et al. ([103], p. 16)

Ce framework a eu un énorme succès dans la communauté SI, en particulier celle du Management des SI (MIS). Il a néanmoins été l'objet de quelques commentaires critiques (p.ex. [178], [247]). Ces critiques portent entre autres sur **la nature de la contribution** d'une recherche conduite selon ce framework : est-ce l'artéfact lui-même, est-ce la généralisabilité de cet artéfact (c.-à-d. sa capacité à répondre à une catégorie de problèmes et non pas à un problème en particulier), est-ce l'exploitation innovante de nouvelles connaissances scientifiques pour concevoir l'artéfact dans une logique de type « design scientifique », ou est-ce encore les résultats de l'évaluation empirique de l'artéfact ? Ou faut-il tout ça à la fois pour qu'il y ait contribution ? Il est vrai que l'article de Hevner *et al.* reste un peu vague sur l'articulation entre artéfact et connaissance.

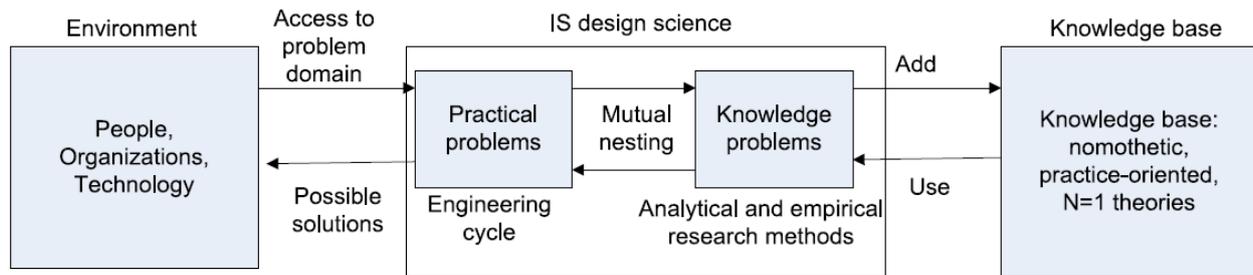


Fig. II-5. Cadre de référence de Wieringa et distinction entre **problème pratique** et **problème de connaissance** ([243])

Cette ambiguïté a été clarifiée par R. Wieringa dans une version alternative du cadre de Hevner *et al.* (cf. Fig. II-5). L'auteur fait une distinction salutaire entre le **problème pratique** dont la résolution nécessite la conception d'un artéfact à l'aide d'une méthode de conception, et le **problème** (ou question) **de connaissance** dont la résolution nécessite la production d'une connaissance nouvelle, rattachée certes à l'artéfact, mais produite selon les canons de la méthode scientifique (cf. §II.3). Le « Design Science » devient ainsi un processus récursif d'exploration d'un entrelacement de questions pratiques et de questions de connaissances [243]. La démarche de résolution alterne entre la conception d'un artéfact répondant aux exigences, et l'investigation empirique pour démontrer les propriétés de cet artéfact telles que faisabilité sous contraintes de ressources, qualité intrinsèque et qualité perçue, performance, etc.

## II.6 Mes contributions aux questions méthodologiques en informatique et SI

Dans cette section, je vais présenter brièvement une série de travaux qui concernent des questions d'ordre méthodologique. L'ensemble de ces travaux est récent et a été réalisé sur la période 2012-2016 ; à chacun correspond au minimum une publication. Ces publications illustrent mon intérêt croissant pour les questions d'ordre méthodologique et épistémologiques. Ils concernent des problématiques relatives :

- d'une part, aux **méthodes d'ingénierie** (l'ingénierie des méthodes, l'ingénierie dirigée par les modèles, la créativité dans l'ingénierie des besoins),
- et d'autre part, aux **méthodes de recherche** (les méthodes empiriques en ingénierie des SI, méthodes de revue de la littérature, la réplication en génie logiciel, théorisation et construction de théorie).

Il faut noter que la section II.6.6 (à propos de la réplication en génie logiciel) fait partie d'un travail de plus grande envergure auquel est consacré un chapitre entier de ce mémoire (cf. chapitre IV).

### II.6.1 Ingénierie des méthodes : synthèse sur l'adaptabilité et l'adaptation

Dès la fin des années 80, plusieurs investigations empiriques (p.ex. [204], [77], [79], [113]) ont montré que les méthodes d'ingénierie des SI sont parfois non utilisées, ou qu'elles le sont très rarement. Les méthodes peuvent se révéler inadaptées aux spécificités des organisations (p.ex. trop complexes pour être suivies), ou, à l'intérieur d'une organisation, inadaptées aux spécificités de ses différents projets [78]. La prise en compte du contexte spécifique de chaque projet et de chaque organisation, ainsi que les particularités de chaque SI en termes d'usage et d'architecture, se révèlent souvent très difficiles. La pratique réelle montre que les acteurs du projet construisent des méthodes "sur mesure" qui seront plus ou moins bien adaptées aux besoins et au contexte du projet en cours. Cette construction se fait dans la majorité des cas par l'adaptation d'une méthode existante, ou par capitalisation de pratiques et d'expériences méthodologiques antérieures pour construire une méthode complètement nouvelle.

Dans un ouvrage consacré à « L'adaptation dans tous ses états »<sup>14</sup>, j'ai participé à la rédaction de deux chapitres consacrés à cette dimension « Adaptation » des méthodes d'ingénierie [publi#17, #18]. La problématique de l'adaptabilité des méthodes relève de **l'ingénierie des méthodes**, un domaine de l'ingénierie des SI qui considère les méthodes elles-mêmes comme un sujet d'étude [192]. Une méthode est ainsi considérée comme un objet artificiel qui a un cycle de vie comportant des phases de définition, de description, d'adaptation et d'exploitation. Ce qui nécessite que **les méthodes soient représentées et modélisées**, et qu'on puisse *raisonner sur ces modèles* pour explorer l'espace du problème (c.-à-d. la situation et le contexte du projet) ainsi que l'espace des solutions (c.-à-d. les méthodes qui correspondent le mieux aux besoins de la situation). La modélisation des méthodes repose sur la distinction entre le **produit** et le **processus** : le modèle de produit prescrit ce que sont les caractéristiques attendues des produits fabriqués, il est exprimé dans les termes d'un méta-modèle de produit dont l'instanciation donne un modèle pour le projet en cours ; le processus est la route à suivre pour atteindre la cible que constituent les produits [168], il est exprimé dans les termes d'un modèle dont l'instanciation donne le chemin suivi dans le projet en cours.

L'adaptation de méthodes soulève des problématiques nouvelles relatives à l'expression de la variabilité dans la description de méthodes (la nature des livrables, les responsabilités des rôles, les étapes à suivre, etc.), aux critères de l'adaptation (profil des acteurs qui utiliseront la méthode, les contraintes de l'environnement, etc.) et au processus d'adaptation (suivi et guidage de l'adaptation). Les deux chapitres de l'ouvrage abordent l'adaptabilité des méthodes d'ingénierie selon deux approches : l'approche des **méthodes adaptables**, et celle des **méthodes adaptatives** (cf. Fig. II-6). Pour évaluer les approches proposées dans la littérature concernant ces deux perspectives, cinq critères ont été définis : (i) comment la variabilité est exprimée, (ii) quels sont les critères de l'adaptation, (iii) quelle est la portée de l'adaptation en termes de constituants de la méthode (les produits, les processus, les recommandations, etc.), (iv) quelles techniques sont proposées pour mettre en œuvre l'adaptation, et (v) la forme et le niveau de détail, de formalisation et de guidage du processus d'adaptation.

<sup>14</sup> P. Lopisteguy, D. Rieu, P. Roose (éditeurs), *L'adaptation dans tous ses états*, Cepaduès-éditions, France, 2012.

### L'approche des méthodes adaptables

Ces méthodes proposent des mécanismes pour « adapter » une méthode existante avant sa mise en application (cf. Fig. II-6g). Cette forme d'adaptation est très répandue dans la pratique, ses mécanismes ne sont pas toujours explicités et/ou formalisés. L'adaptation prend alors la forme générale d'une construction *ad-hoc* basée sur l'expérience et une prise en compte intuitive du contexte (p. ex. [80],[232],[13]). Plusieurs techniques existent, de la plus simple où l'on sélectionne des éléments dans ceux que propose une méthode telle qu'UML en se restreignant à ceux utiles au contexte, et qu'on peut aussi spécialiser et rendre plus spécifique au contexte ; jusqu'à des approches plus sophistiquées avec des motifs génériques (des « patterns méthodologiques ») qu'on peut imiter, ou encore la configuration d'une méthode « universelle » selon un ensemble de paramètres ; cette configuration peut être très limitée, p.ex. en indiquant si un élément de la méthode (un rôle, un livrable, une étape, etc.) est obligatoire ou pas.

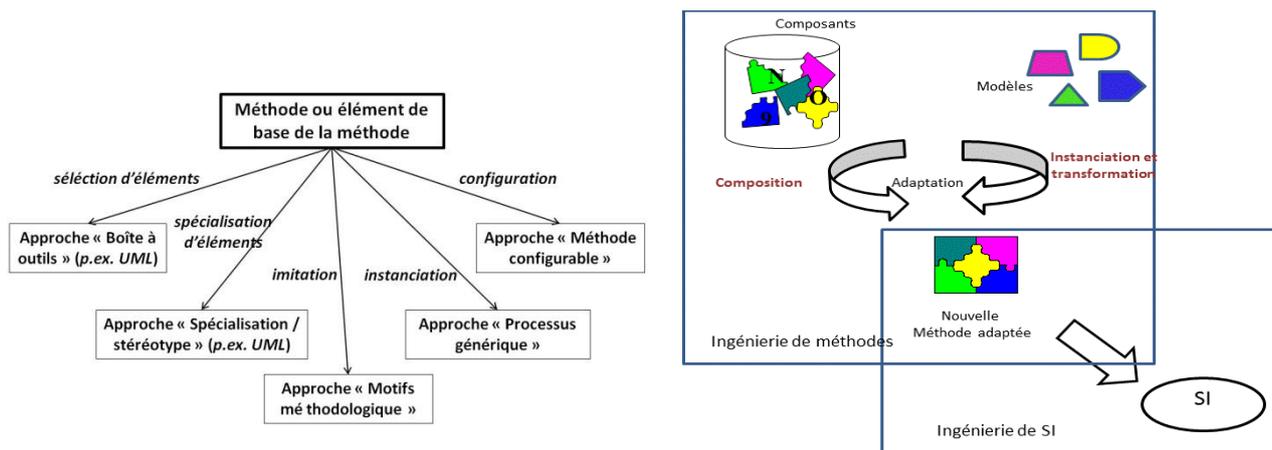


Fig. II-6. Principe des approches adaptables (g), et principes des méthodes adaptatives (d)

### L'approche des « méthodes adaptatives »

Ces méthodes proposent de construire « de toutes pièces » une méthode adaptée à une situation donnée (cf. Fig. II-6d). En général, cette approche introduit un niveau de méta-modélisation à partir duquel on construit par instanciation une nouvelle méthode. Ce niveau générique prend la forme de méta-modèles de produits, de méta-modèles de processus, de composants de méthodes, etc. La construction de ce niveau générique se fait à l'aide de langages de méta-modélisation (p.ex. GOPRR [124], MOF [169], SPEM [171], etc.), par composition de fragments méthodologiques (p.ex. [30], [183]), ou par orchestration de services web méthodologiques (p.ex. [193]).

### Commentaire à propos de l'ingénierie des méthodes

Les experts dans les sciences de conception reconnaissent clairement que les méthodes de conception dépendent fortement du domaine dans lequel s'exerce la conception, tels que l'architecture, le design industriel, ou en l'occurrence le génie logiciel et l'ingénierie des SI. Cette dépendance est directement liée au niveau de structuration et de formalisation des artefacts manipulés. Dans le cas du design industriel par exemple, la Conception Assistée par Ordinateur (CAO) a fait significativement progresser l'activité de conception. Ce progrès repose en grande partie sur la possibilité de visualiser et de manipuler des représentations mathématiques et informatiques des objets et composants du design industriel (p.ex. une pièce mécanique dans un avion). De même, les règles d'assemblage de ces composants sont formalisables, et il devient ainsi possible de construire des outils logiciels sophistiqués pour assister le concepteur dans le processus conception et dans l'exploration de l'espace des solutions possibles.

Dans le cas des méthodes d'ingénierie des SI, la principale difficulté réside dans la définition et formalisation des composants d'un artefact logiciel (vision produit) et des règles d'assemblage (vision processus). La définition exacte du niveau de granularité, de la forme et contenu d'un composant méthodologique ont fortement contraint le développement de l'ingénierie des méthodes [102]. Une autre difficulté est celle du langage ou formalisme à utiliser pour définir ce composant méthodologique, elle influe directement sur les mécanismes d'assemblage, et aussi sur l'interopérabilité des différentes approches. Les propositions actuelles, telles que le méta-modèle MOF pour les pro-

duits, le méta-modèle SPEM pour les processus [171], ainsi que la proposition de la norme ISO/IEC 24744 [101], constituent des avancées significatives. Elles ne permettent cependant pas la réalisation complète de l'approche par assemblage de composants méthodologiques, et peu de cas concrets d'application de cette approche existent. Cette question fait partie des principales critiques formulées récemment à l'encontre de l'ingénierie des méthodes [134].

## II.6.2 IDM et ingénierie des besoins : synthèse des travaux d'un Workshop

L'ingénierie Dirigée par les Modèles (IDM) est une vision globale de l'ingénierie logicielle, elle prône l'usage extensif de modèles tout le long du cycle de vie [17]. Dans cette vision, les modèles conceptuels sont transformés selon des règles précises en modèles de plus bas niveau d'abstraction – de préférence à l'aide d'outils appropriés ; l'obtention du code final de l'application est totalement automatisée à l'aide de générateurs de code. Une autre variante de l'approche est que les modèles conceptuels (ou une version de plus bas niveau obtenu après transformation) soit directement exécutable. De ce point de vue, le projet Rubis et les travaux que j'y ai menés dans ma thèse de doctorat autour du langage exécutable Proquel (cf. Chap.III) sont tout à fait précurseurs par rapport à cette vision de l'ingénierie logicielle.

L'IDM a connu beaucoup de développement avec notamment le langage Atlas pour l'expression précise de la transformation de modèles [119], ainsi que des applications dans de multiples domaines de l'ingénierie des SI (p.ex. [39],[115]). Néanmoins, une question importante se pose : à partir de quel niveau d'abstraction est-il possible d'appliquer des transformations automatiques, et plus précisément, est-il possible d'appliquer la vision de l'IDM aux modèles utilisés dans l'ingénierie des besoins ? C'est le thème principal de l'atelier de recherche MoDRE<sup>15</sup>, organisé annuellement depuis 2011 dans le cadre de la conférence internationale Requirement Engineering.

Dans un article de recherche publié dans ce même atelier en 2014 [publi#26], j'ai analysé l'ensemble des articles publiés dans les trois sessions précédentes cet atelier, soit un total de 29 articles, et réalisé une présentation synthétique. L'objectif est d'identifier les tendances en termes de problématiques, de solutions, et d'approches méthodologiques. Il en ressort quelques résultats intéressants :

- ❖ C'est surtout la modélisation des besoins fonctionnels (72% des cas) qui est traitée au dépend de la modélisation des besoins non-fonctionnels.
- ❖ De même, la problématique la plus traitée est celle de la modélisation, et plusieurs nouveaux langages de modélisation des besoins sont proposés combinée à des outils spécifiques de transformation ; quelques travaux considèrent des langages de modélisation existants tels qu'UML ou BPMN.
- ❖ Aucune approche parmi celles analysées ne concernait la modélisation à base de but, alors que celle-ci est particulièrement représentative des techniques de modélisation des besoins.
- ❖ En termes d'évaluation, beaucoup d'illustrations et d'études de cas, peu d'expérimentations contrôlées ; ceci est toutefois conforme avec la nature exploratoire des publications dans les ateliers de recherche.

Pour conclure, je dirais que le thème de l'atelier MoDRE est conforme à l'esprit des méthodes en génie logiciel et en SI qui ambitionnent l'automatisation de la construction de systèmes à partir de spécifications conceptuelles. Il y a néanmoins des ambiguïtés quand au positionnement optimal de l'IDM, est-ce au niveau de l'espace des problèmes (comme l'est la modélisation des buts) ou est-ce au niveau de l'espace des solutions ? Cette question détermine la portée (le « scope ») de l'IDM dans l'ingénierie des besoins et son potentiel global d'innovation en termes de méthode d'ingénierie.

<sup>15</sup> MoDRE : Model Driven Requirement Engineering ([www.modre2016.ece.mcgill.ca/](http://www.modre2016.ece.mcgill.ca/))

### II.6.3 Créativité et ingénierie des besoins : une revue de la littérature

Par rapport à l'étude précédente, celle présentée ici aborde une facette complètement différente de la problématique de l'ingénierie des besoins. Il s'agit d'explorer le potentiel des techniques de créativité pour améliorer et enrichir les approches de recueil des besoins. Dans toute activité de conception, le besoin initial, le problème constaté dans l'expérience de l'utilisateur (cf. §II.4.1) n'est pas toujours clair et bien défini, et le processus de conception consiste aussi à le clarifier et l'explorer. Dans le cas de l'ingénierie informatique, la difficulté à modéliser l'artéfact informatique et sociotechnique et à visualiser son impact sur le contexte d'usage complique encore un peu la phase de recueil des besoins et de définition du problème. En réalité, cette phase ne consiste pas seulement à « recueillir les besoins » mais à les explorer, et peut être même à les *inventer*, car « on ne connaît pas ce qu'on veut jusqu'à ce qu'on sache qu'il existe » ([189], p.48). Ce point de vue amène à considérer l'ingénierie des besoins comme un *processus d'innovation*, où la créativité joue un rôle important pour amener l'utilisateur et le concepteur à imaginer des solutions nouvelles qui répondent non seulement aux besoins initiaux, mais aussi à des besoins méconnus ou nouveaux qui émergent au fur et à mesure que le problème et sa solution sont explorés.

L'objectif de cette étude est de faire une synthèse des différentes techniques de créativité et quelles utilisations sont possibles. Cette étude est une revue de la littérature qui a été publiée dans une revue en ligne destinée aux praticiens et aux chercheurs [publi#01]. L'apport des techniques de créativité à l'ingénierie des besoins a été relativement peu explorée dans la recherche, et une revue similaire de la littérature en 2012 a identifié quatre groupes principaux qui travaillent sur le sujet, et a conclu qu'il y a peu de constats empiriques à ce propos [143]. En ciblant une population de praticiens, l'intérêt de cette étude menée conjointement avec un doctorant, est de contribuer à mieux faire connaître ces méthodes et leur potentiel apport au processus de conception en SI.

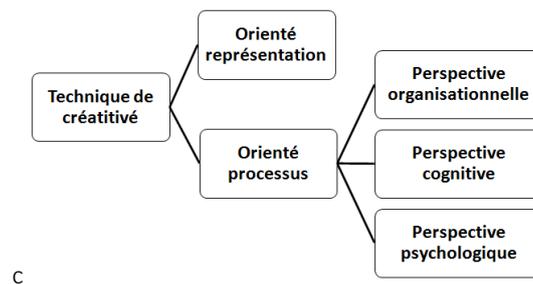


Fig. II-7. Typologie des techniques de créativité analysées dans [publi#01]

Les résultats de notre étude ont permis d'identifier dans la littérature **30 techniques qualifiées de créatives** par leurs auteurs. Certaines de ces techniques sont assez similaires, et analysées plus en profondeur, elles pourraient être réduites à une seule technique avec plusieurs variantes. D'autres sont peut être assez similaires pour les considérer comme étant la même, mais référencée sous un nom différent. Nous n'avons pas fait cette analyse approfondie, on s'est contenté de classer ces techniques selon une typologie hiérarchique (cf. Fig. II-7). Dans cette typologie, on distingue d'abord les techniques orientées « représentation » et qui agissent au niveau de la représentation des besoins pour stimuler la créativité ; les techniques connues de modélisation des buts, de rédaction de scénarios ou de « Topic maps » en font partie. Dans les techniques orientées « processus », on distingue celles qui portent sur l'organisation de l'étape de recueil des besoins (p.ex. « Brainstorming » ou « Roles playing »), celles qui ciblent l'activité cognitive des participants (p.ex. « Analogies », « Constraint removal », ou « Why why why »), et enfin celles qui visent la psychologie des participants (p.ex. « Influence positive emotions »).

### II.6.4 Synthèse sur les méthodes empiriques en ingénierie des SI

J'ai coordonné en 2015 un numéro spécial<sup>16</sup> de la revue « Ingénierie des SI » consacré aux méthodes empiriques et expérimentales. L'objectif de ce numéro spécial est de faire connaître ces méthodes et leur rôle dans la recherche en

<sup>16</sup> Revue *Ingénierie des SI*, Vol.20, n°6, décembre 2015, <http://isi.revuesonline.com/resnum.jsp?editionId=3045>

ingénierie, et de clarifier la vision émergente de « Design science » en tant que combinaison et alternance entre activité de conception d'artéfact et activité d'évaluation expérimentale ou empirique (cf. §II.5). Ce numéro spécial a rassemblé cinq articles de recherche, auxquels s'ajoute un article introductif de 22 pages que j'ai rédigé sur la problématique méthodologique dans l'ingénierie. Dans cet article [publi#03], j'explique d'abord les méthodes empiriques et leur rôle important dans l'ingénierie des SI, et je présente succinctement les principales méthodes telles qu'elles sont mentionnées dans la littérature. Ensuite, j'illustre la démarche empirique d'investigation à travers neuf exemples d'études empiriques en informatique et ingénierie des SI, en prenant soin de mettre en évidence l'importance de deux perspectives : **explorer** un phénomène ou **l'expliquer**. Si l'explication aboutit à une connaissance conceptuelle ou théorique nouvelle, l'observation permet néanmoins de suggérer des pistes pour de nouvelles recherches et pour découvrir de nouveaux artéfacts grâce notamment à l'observation inductive (cf. §II.3.3).

Le Tableau II-1 présente les principales méthodes empiriques. Ce tableau a été obtenu en confrontant la liste des 12 méthodes indiquées par Zelkowitz et Wallace en 1998 [254], avec celles présentées dans quatre sources plus récentes : Wohlin et al. en 2003 [250], Easterbrook et al. en 2008 [65], Shull et al. en 2008 [214], et Wohlin et al. en 2012 [251]. Les méthodes sont regroupées en trois catégories [254] : observationnelle, historique et contrôlée. Chaque méthode est caractérisée avec trois propriétés : la portée (exploratoire ou explicative), le contexte d'application (*in-vivo* si la méthode a vocation à être utilisée dans un contexte réel, ou *in-vitro* si c'est dans un environnement expérimental), et enfin si la méthode est quantitative ou qualitative (cf. II.3.2).

Catégorie	Méthode	Portée		Contexte		Approche	
		Exploratoire	Explicative	In vivo	In vitro	Quantitative	Qualitative
Observationnelle	1. Assertion	X			X	X	X
	2. Observation de projets	X		X		X	X
	3. Ethnographie et entretiens semi-structurés	X	X	X	X		X
	4. Focus group	X	X	X	X		X
	5. Enquête	X	X	X	X	X	
	6. Étude de cas	X	X	X		X	X
	7. Étude longitudinale de terrain	X	X	X		X	X
	8. Recherche action	X	X	X		X	X
Historique	9. Revue de littérature	X	X	NA		X	X
	10. Leçons apprises	X		X	X	X	X
	11. Analyse Post-Mortem	X		X	X	X	X
	12. Fouille de données	X	X	X	X	X	
Contrôlée	13. Expérience(s) contrôlée(s)	X	X	X	X	X	X
	14. Analyse comparative	X	X		X	X	X
	15. Simulation	X	X		X	X	X

Tableau II-1. Principales méthodes empiriques en ingénierie des SI ([publi#03], p.21)

Cette étude synthétique sur les méthodes empiriques met en évidence qu'en ingénierie, par rapport aux sciences de la nature et aux sciences humaines, la logique des découvertes est différente. C'est la maîtrise des lois qui gouvernent le fonctionnement interne des artéfacts qui permet, par application innovante de ces lois, de créer une multitude d'artéfacts. Les seules limites sont l'imagination créatrice de l'Homme et l'applicabilité de ces lois. Cependant, quelle utilité, quelle applicabilité, bref, quel avenir pour ces artéfacts ? C'est là où l'empirisme et l'expérimentation trouvent leur place dans les sciences de l'ingénieur. Ce mémoire d'HDR, et ce chapitre en particulier, développent ce constat et en font un leitmotiv qui est la base de ma vision multidisciplinaire de l'informatique et des SI.

### II.6.5 Synthèse sur les méthodes de revue de la littérature

Dans le cycle de la méthode universelle de recherche (cf. Fig. II-1), le chercheur doit interroger la base des connaissances existantes lorsqu'il mène sa recherche. Cette étape de revue de la littérature est incontournable dans tout travail de recherche, et c'est généralement la porte d'entrée du doctorant dans le monde de la recherche. Avec l'évolution des technologies (Internet, librairies digitales, etc.), la revue de littérature est devenue une tâche plus complexe qui exige rigueur dans le choix et la sélection des sources, et objectivité dans la formulation d'une synthèse concluante sur un domaine ou une question particulières. Les méthodes de revue de littérature sont ainsi devenues un sujet d'étude, d'abord dans les sciences des faits (p.ex. en psychologie [15]) et ensuite dans le génie logiciel avec ce qu'on appelle « la revue systématique de la littérature » [127].

Dans un article publié dans un séminaire de recherche dédié aux méthodes de recherche en SI, j'ai présenté une étude sur les méthodes de revue de littérature [publi#35]. J'ai analysé **16 publications** relatives aux méthodes de conduite de la revue de la littérature. Pour chacune, j'ai identifié le domaine dans lequel elle a été publiée (informatique, génie logiciel empirique, management, ou SI), sa perspective (qualitative, narrative, quantitative, systématique, intégrative, etc.), et j'ai présenté très brièvement la démarche qu'elle préconise et les techniques et recommandations pour l'appliquer.

Il ressort de cette étude quelques points intéressants d'un point de vue méthodologique :

- Au-delà de la rigueur nécessaire dans la sélection des sources, la problématique clé reste celle de la **synthèse**. Quel résultat concret peut-on obtenir lorsque la littérature contient plusieurs dizaines de publications sur un sujet donné ? La tendance est forte à simplement comptabiliser les publications selon des catégories, ou à énumérer les résultats et conclusions de chaque publication, et rester ainsi dans un mode *narratif* de la synthèse. Le mode narratif (appelé aussi « qualitatif ») peut être une approche intéressante pour analyser en profondeur un ensemble de publications. Elle est cependant difficilement applicable lorsque le nombre de publication est élevé, ou que les résultats des recherches étudiées sont de nature quantitative.

- La « méta-analyse » est une méthode basée sur des **techniques statistiques pour synthétiser de résultats** de nature quantitative ; elle est a été initialement développée dans les disciplines de médecine [67]. Malgré le fait que sa mise en œuvre soit délicate et un peu difficile, son usage se confirme dans d'autres disciplines (p.ex. en sciences de l'éducation [224] ou en génie logiciel [95]). En effet, cette méthode répond au problème évoqué de comment synthétiser de manière méthodique et rigoureuse les résultats trouvés dans la littérature.

La question des méthodes de revue de la littérature reste un domaine actif de recherche, et des points de vue nouveaux émergent (p.ex. sur l'applicabilité de la méthode systématique [23], ou sur sa répliquabilité [249]). Ce travail a attiré mon attention sur l'importance de la revue de la littérature, et m'a permis de mieux maîtriser les méthodes à appliquer. Plusieurs publications récentes font suite à ce travail [publi#32, #06, #05, #01].

### II.6.6 Réplication d'une étude en génie logiciel empirique

Considéré de manière isolée, le résultat d'une expérimentation ou d'une série d'observations empiriques peut difficilement prétendre, à lui seul, être une description ou explication complète de la réalité sous-jacente. La réplication est la répétition d'une recherche empirique ou expérimentale pour révérifier ses résultats et augmenter ainsi la confiance qu'on y accorde. Elle implique la **reproduction de l'étude** dans des conditions identiques ou similaires pour contrôler que ce qui a été constaté se reproduit de manière régulière et stable, qu'il n'est pas dû au hasard ou à une conjonction de facteurs autres que ceux identifiés [200]. C'est une question importante dans la science des faits, en tant que complément indispensable à la méthode hypothético-déductive pour renforcer la validité (ou l'invalidité) des hypothèses. Le besoin d'augmenter le nombre et la qualité des réplifications est évoqué dans plusieurs publications récentes (p.ex. [213] en génie logiciel, [208] en psychologie, et [116] dans la revue *Science*).

J'ai été amené à réfléchir sur la question de la réplication dans le cadre d'un travail de recherche réalisé durant un séjour sabbatique en 2012 à l'université de Lund, Suède. Au sein de l'équipe SERG, spécialisée dans la recherche empirique en génie logiciel, j'ai mené une étude empirique qui analyse les rapports textuels d'erreurs logicielles (les

« bugs reports ») pour prédire le temps de résolution. Ce travail repose sur une étude publiée en 2013 dans la revue *Empirical Software Engineering* [182], et qui stipule que la catégorisation automatique des rapports textuels des erreurs selon une technique d'apprentissage automatique non supervisé, pourrait être exploitée pour prédire le temps de résolution d'une erreur lorsque celle-ci est déclarée (c.-à-d. décrite dans un rapport textuel).

Dans le travail que nous avons réalisé [publi#02], la première étape consistait à répliquer l'étude initiale pour vérifier la validité de ses résultats. La réplication soulève plusieurs questions importantes relatives au niveau de similarité nécessaire – ou souhaitable – entre la réplication et l'étude initiale qui est répliquée. Peut-elle être réalisée par les mêmes chercheurs ou doit-elle être réalisée par d'autres ? Doit-on utiliser les mêmes données, le même procédé opératoire, les mêmes outils d'analyse, etc. ? Au-delà du constat qu'une réplication exacte d'une étude empirique – une répétition à l'identique – est pratiquement impossible ([31], p. 370), une telle réplication n'aurait en réalité qu'un intérêt limité. Certes, son succès confirmerait les conclusions et la validité interne de l'étude initiale, mais elle n'apporterait aucun élément nouveau concernant la généralisation de ses conclusions [126]. En réalité, ce sont **les différences entre l'étude initiale et la réplication** qui ont le potentiel de générer de nouvelles connaissances [122]. En cas de succès, une réplication différenciée confirmerait nettement les conclusions de l'étude initiale et étendrait le périmètre de validité de ses résultats. En cas d'échec, cela ne remettrait pas entièrement en cause l'étude initiale, mais constituerait un sérieux indicateur de l'existence de facteurs inconnus jusqu'alors, qui ont influencé le résultat de l'étude et qui contribuent donc à en limiter le périmètre de validité ([200], p. 6).

Dans la réplication que nous avons menée, plusieurs facteurs ont été modifiés, tels que le jeu de données, l'outil et l'algorithme d'apprentissage automatique, etc. Nous avons réalisé appliqué les mêmes tests statistiques que ceux de l'étude initiale (quoique avec des outils logiciels différents). Les résultats obtenus ont globalement confirmé ceux de l'étude initiale. Forte de ce résultat, notre étude s'est penchée sur l'évaluation de l'hypothèse émise par l'auteur de l'étude initiale à l'aide d'un protocole expérimental basé sur la simulation et la variation des paramètres. Cet aspect de l'étude fait l'objet d'un chapitre ultérieur de ce mémoire (chapitre IV).

### II.6.7 Théorisation dans la recherche empirique sur les ERP : études exploratoires

La construction de théories est au cœur de la démarche scientifique. Sans rentrer dans les détails, une théorie est un ensemble d'hypothèses et de présomptions qui réfèrent à un contexte bien identifié dans le monde réel (p.ex. les utilisateurs d'un ERP dans une PME). Cet ensemble d'hypothèses et de présomptions doit former un système hypothético-déductif testable ; l'objectif de la théorie ainsi formée est de fournir des descriptions, des explications et des prédictions sur l'univers factuel concerné ([34], p. 381-383). Une théorie s'attache donc à exprimer, de manière synthétique, une certaine connaissance générique conforme au cycle général de recherche scientifique (cf. Fig. II-1). Dans les sciences des faits, les théories sont des éléments fondamentaux que le chercheur peut mobiliser pour définir les hypothèses de travail et élaborer un moyen de les tester et construire des schémas descriptifs et des explicatifs.

Dans les domaines du génie logiciel et des SI, l'importance des théories et de la démarche de théorisation a été soulevée par plusieurs chercheurs. Dans son article séminal en 2006 [90], Shirley Gregor pose la problématique de la théorisation dans la recherche en SI, et définit **cinq grandes catégories de théories** : Analytique (A) – Explicative (E) – Prédicative (P) – Explicative et prédictive (EP) – Conception et action. C'est sur la catégorie EP qu'elle insiste : des théories capables d'explicitier **un lien de cause à effet** ou de **corrélation**, et susceptibles d'être utilisées pour prédire l'occurrence des phénomènes étudiés<sup>17</sup>. A la même période, un article dans la revue IEEE TSE traitait de la question des théories dans le domaine du génie logiciel [96]. Les auteurs analysent les théories mobilisées dans un panel de 103 articles de recherche; il en ressort que seulement 24 articles font appel à des théories, avec une tendance vers l'interdisciplinarité (c.-à-d. font appel à des théories d'autres disciplines). Les auteurs appellent ainsi à accroître la théorisation des recherches en génie logiciel ; les travaux de R. Wieringa et son équipe vont dans ce sens [245], [244].

<sup>17</sup> En prenant l'exemple des ERP dans les PME, une **hypothétique théorie explicative et prédictive** serait du genre « *le succès de l'implémentation d'un ERP dans une PME, mesuré par la progression de son chiffre d'affaire, dépend du niveau de qualification de ses employés (mesuré par le ratio de personnel diplômé) et de sa maturité organisationnelle (mesurée selon l'échelle CMMI)* ».

Dans le travail réalisé avec R. El Amrani [publi#29, #25], on s'est penché sur une problématique très étudiée, celle des usages et des impacts des ERP. C'est une question qui suscite beaucoup d'intérêts à cause de l'importance des ERP dans l'informatique des entreprises, et un grand nombre de publications sont disponibles sur le sujet (p.ex. plus de 800 articles analysés dans [206]). Malgré ce grand nombre d'article, aucune théorie générale n'a vu le jour, et l'objectif de notre étude est de chercher à expliquer cette absence de théorie et comment y remédier.

Notre étude est pour le moment exploratoire et porte sur un petit échantillon d'articles publiés dans deux revues (*European Journal of IS*, *Journal of IS*). L'objectif est, dans un premier temps, de constater comment les résultats des études empiriques sur les ERP sont décrits, s'il y a une théorisation des résultats ou pas. On cherche aussi à identifier les théories – existantes – utilisées pour décrire et expliquer les phénomènes liés à l'usage des ERP. Dans un second temps, notre objectif est de comprendre comment les contributions théoriques dans ce domaine sont construites par les auteurs, et d'envisager la possibilité de construire une théorie, générale ou de moyenne envergure, qui explique le succès de l'usage et l'adoption des ERP.

## II.7 Conclusion du chapitre

Ce chapitre aborde un aspect fondamental de mon analyse de la multidisciplinarité en informatique et en SI, ainsi que les contributions que j'ai récemment réalisées à cet égard. Ces contributions m'ont donné l'opportunité de me pencher sur des questions qui relèvent parfois de la philosophie des sciences et de l'épistémologie. Grâce à ces travaux, j'ai pu développer une compréhension holistique de la notion de « méthode », telle qu'elle est définie et pratiquée dans les sciences en général, et en informatique et en SI en particulier. C'est cette compréhension des aspects méthodologiques, dans les sciences des faits et d'ingénierie, qui constitue un leitmotiv de ce mémoire d'HDR.

Il en ressort clairement qu'il y a **une certaine ambiguïté** sur le sens exact du terme « méthode » dans les différentes disciplines. La méthode de conception, en tant que savoir-faire pratique pour concevoir des objets et des projets, et qui est au cœur de l'activité d'ingénierie, est difficile à formaliser et à capturer. La **créativité, l'intuition et l'expertise** du concepteur jouent un rôle important dans la conception d'artéfacts ingénieux, utiles et innovants. Néanmoins, un travail significatif a été réalisé dans les sciences de conception pour guider, structurer et supporter le processus de conception. Dans le domaine de l'informatique et des SI, et malgré le fondement formel de l'informatique, il est plus difficile de **formaliser le processus de conception** sous-jacent à l'ingénierie informatique. L'ingénierie des méthodes, la vision IDM, l'apport des techniques de créativité sont des pistes de recherche que j'ai explorées ou pratiquées dans ma recherche.

Cette ambiguïté du terme « méthode » en ingénierie se manifeste au regard de la notion de méthode telle qu'elle est définie dans les sciences des faits. Il y a parfois une confusion entre les deux, où la méthode d'ingénierie est considérée comme une méthode de recherche. Par exemple, dans une thèse de doctorat à l'université de York (R. Ramsin, p.35, [185]), une méthode d'ingénierie est explicitement présentée en tant que méthode de recherche. De même, dans le *framework* initial du Design Science de Hevner et al. [104], la production de connaissance n'est pas clairement explicitée. Bref, il y a une certaine tendance à confondre **la production de l'artéfact** avec la **production de la connaissance**.

Il est cependant tout à fait envisageable de clarifier cette différence entre production d'artéfact et production de connaissance. Le principe de « la découverte par la conception<sup>18</sup> » pose effectivement les bases d'une science de conception où **la création des connaissances va de pair avec la conception de l'artéfact** ; elle nécessite cependant une distinction claire entre les deux. Le développement de théories dans le domaine de la conception est une piste à explorer pour articuler ces deux facettes de l'activité d'ingénierie. C'est actuellement un sujet de débat dans la communauté de recherche en SI [20].

---

<sup>18</sup> "Designerly ways of knowing", Nigel Cross, 1982

# III. Chapitre 3 : L'exécution de modèles conceptuels

## Sommaire

<a href="#"><u>III.1</u></a>	<a href="#"><u>Préambule</u></a>	40
<a href="#"><u>III.2</u></a>	<a href="#"><u>Problématique</u></a>	41
<a href="#"><u>III.2.1</u></a>	<a href="#"><u>Définir un langage</u></a>	41
<a href="#"><u>III.2.2</u></a>	<a href="#"><u>Définir sa sémantique</u></a>	42
<a href="#"><u>III.2.3</u></a>	<a href="#"><u>Exécution d'un langage vs. "process enactment"</u></a>	43
<a href="#"><u>III.2.4</u></a>	<a href="#"><u>Conclusion et positionnement</u></a>	43
<a href="#"><u>III.3</u></a>	<a href="#"><u>Bref retour sur l'exécutabilité du modèle REMORA</u></a>	44
<a href="#"><u>III.4</u></a>	<a href="#"><u>Une approche ad-hoc pour interpréter les cartes MAP</u></a>	45
<a href="#"><u>III.4.1</u></a>	<a href="#"><u>Contexte : modélisation intentionnelle et carte MAP</u></a>	45
<a href="#"><u>III.4.2</u></a>	<a href="#"><u>La sémantique d'une carte Map</u></a>	46
<a href="#"><u>III.4.3</u></a>	<a href="#"><u>Tentative de formalisation de la sémantique d'un MAP</u></a>	47
<a href="#"><u>III.4.4</u></a>	<a href="#"><u>Une architecture de base de données pour exécuter les cartes</u></a>	48
<a href="#"><u>III.4.5</u></a>	<a href="#"><u>Conclusion sur l'approche ad-hoc</u></a>	49
<a href="#"><u>III.5</u></a>	<a href="#"><u>Vers une approche générique pour l'expression de l'exécutabilité</u></a>	49
<a href="#"><u>III.5.1</u></a>	<a href="#"><u>Une approche de type IDM pour définir l'architecture d'exécution</u></a>	49
<a href="#"><u>III.5.2</u></a>	<a href="#"><u>Modélisation événementielle de la sémantique</u></a>	50
<a href="#"><u>III.5.3</u></a>	<a href="#"><u>Exploitation de la sémantique opérationnelle</u></a>	51
<a href="#"><u>III.5.4</u></a>	<a href="#"><u>Application sur le modèle de carte MAP</u></a>	53
<a href="#"><u>III.6</u></a>	<a href="#"><u>Conclusion et perspectives de recherche</u></a>	54
<a href="#"><u>III.7</u></a>	<a href="#"><u>Annexe 1 : Diagramme de classe du moteur d'exécution d'un modèle de carte (niveau instances)</u></a>	56

### III.1 Préambule

Les travaux présentés dans ce chapitre relèvent du thème 1 (cf. chapitre I), c.-à-d. des modèles et méthodes pour l'ingénierie des SI ; la perspective d'étude est celle des sciences de conception. L'objet d'étude dans ces travaux est le **langage de modélisation** et ses propriétés. Le langage de modélisation (appelé aussi « grammaire de modélisation » [236]) permet de construire des représentations qu'on appelle « modèle », « schéma », ou encore « script » [236]. Ces représentations sont un élément clé dans la réduction de la complexité en ingénierie des SI [144].

Dans le cas de la modélisation dite « conceptuelle », ces modèles sont des **représentations abstraites** de l'artéfact technique et/ou sociotechnique et sont censées être indépendantes de toute contrainte liée à l'infrastructure technologique. Ces modèles contiennent des connaissances sur l'univers organisationnel et humain auquel est rattaché le SI, ainsi que des connaissances sur le fonctionnement et l'architecture de l'artéfact technique (cf. Fig. III-1). Par ailleurs, un courant important de recherche milite pour que la modélisation conceptuelle soit plus centrée sur les besoins des utilisateurs finaux et enracinée dans les objectifs et finalités de l'organisation utilisatrice (p. ex. [196]).

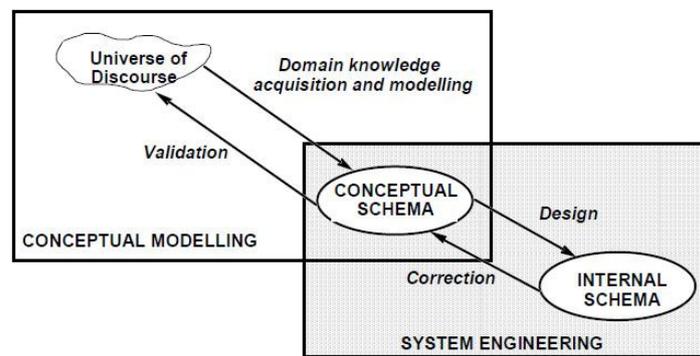


Fig. III-1. La modélisation conceptuelle dans l'ingénierie des SI (d'après Rolland et Prakash (2000), [196])

Un modèle conceptuel n'a pas, à priori, l'objectif d'être exécuté par une machine. Étant une représentation de haut niveau, un peu comme le plan de l'architecte, un modèle n'a pas vocation à **remplacer l'artéfact modélisé**. Néanmoins, dans le domaine de l'informatique et des SI, il est envisageable qu'un modèle puisse s'exécuter. Cette perspective s'appuie sur deux idées directrices :

- La première découle du principe des langages de programmation ; ceux-ci offrent des concepts de plus haut niveau d'abstraction que ceux de l'ordinateur, ils sont néanmoins exécutables grâce aux techniques de compilation.
- La seconde s'appuie sur le fait qu'un modèle conceptuel englobe une abstraction d'un artéfact lui-même abstrait (le logiciel), et qu'il est donc envisageable de **transformer** ces représentations d'un niveau d'abstraction vers un autre.

Ces transformations, et les questions qu'elles soulèvent en termes d'expression et d'implémentation, sont au centre des travaux présentés ici. Elles s'appuient sur l'étude des techniques de représentation d'un langage de modélisation (tel que la méta-modélisation), sur l'étude de l'expression implicite ou explicite de sa sémantique, et sur l'apport éventuel des techniques logicielles issues du monde des compilateurs pour les langages de programmation.

Ces travaux sont une continuité de mon travail initial en thèse autour de la méthode Remora et de l'environnement expérimental Rubis pour le prototypage d'applications [7]. Ils ont été menés en collaboration avec le Centre de Recherche en Informatique (CRI) dans le cadre du co-encadrement de deux thèses de doctorat (50% l'une et 75% l'autre). Ces travaux ont donné lieu à la publication d'un article dans une revue nationale, de 5 articles dans conférences et ateliers de recherche, et d'un poster dans une conférence internationale ; ainsi qu'à l'encadrement de 4 mémoires de Master recherche.

## III.2 Problématique

Le processus de construction d'un SI peut être vu (cf. Fig. III-2) comme une mise en correspondance (« mapping ») ou transformation entre un phénomène du monde réel et une série de scripts (modèles, programmes) interprétables par une machine ([235], p.220). Cette vision induit une **tension forte et paradoxale** entre la capacité des modèles à capturer la complexité du monde réel et celle d'être proche de l'univers de la machine. Les deux facettes de cette tension ont donné lieu à de larges corpus de travaux pour, d'une part, évaluer l'aptitude des modèles à décrire le monde réel, ainsi que proposer de nouvelles approches, techniques et langages de modélisation ; et d'autre part, pour transformer et interpréter ces modèles dans différents niveaux d'abstraction.

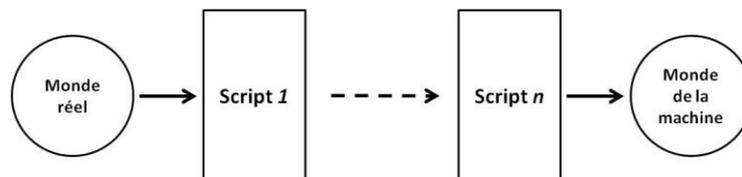


Fig. III-2. Modèle transformationnel de l'analyse, la conception et l'implémentation d'un SI ([237], p.63) ; ou encore, le *mapping* à l'aide de scripts du monde réel vers le monde de la machine ([235], p.220)

Pour ce qui est de l'expressivité des langages de modélisation, les travaux de Wand et Weber (p.ex. [234], [235]) ont eu un large écho car ils font appel à une ontologie rigoureuse et reconnue du monde réel, celle du philosophe Mario Bunge [35], pour confronter les concepts d'un langage de modélisation avec ceux de l'ontologie. Pour ce qui est de la transformation et l'interprétation des modèles, l'essor de l'ingénierie dirigé par les modèles a largement contribué à populariser la question de rendre un modèle exécutable en le transformant (p.ex. [52]), ainsi que celle de relever le niveau d'abstraction des langages de programmation [81].

### III.2.1 Définir un langage

La conception, l'implémentation et l'évaluation des langages sont des problématiques initialement liées à la programmation, le génie logiciel et la compilation [128]. Néanmoins, ces questions ont émergé dans le champ d'investigation de l'ingénierie des SI avec le développement de l'ingénierie dirigée par les modèles (IDM) et des modèles spécifiques au domaine (DSM) [72]. Un DSM est un langage de modélisation qui propose des concepts proches d'un domaine spécifique d'application, tel que le concept de « moyen de paiement » dans le domaine des SI bancaires. L'apport d'un DSM est d'encapsuler dans les concepts spécifiques qu'il propose une connaissance générique sur le domaine, et que le concepteur d'une part, n'a plus besoin de capturer dans ses modèles, et d'autre part, qu'il peut directement utiliser pour construire le modèle pour son projet en cours [156]. Un DSM est généralement défini et implémenté au sein d'un portefeuille de projets dans un domaine spécifique, et de ce fait, suit un cycle de vie qui lui est propre [131].

La définition d'un langage comporte trois volets : la syntaxe abstraite (structure), la syntaxe concrète (forme) et la sémantique (signification). Pour un langage de modélisation, la syntaxe abstraite est généralement décrite à l'aide d'un méta-modèle [128]. Un méta-modèle est une représentation, souvent graphique, des concepts sous-jacents au langage, des liens entre ces concepts ainsi que d'éventuelles contraintes que doivent satisfaire les instances de ce langage. La **sémantique d'un langage** indique le sens que peuvent prendre les constructions de celui-ci lorsqu'elles sont instanciées dans un modèle. C'est une notion difficile à appréhender et qui est fréquemment source d'imprécision et de confusion autant pour le concepteur que pour l'utilisateur du langage [98], [41]. De plus, une étude empirique sur le développement d'un DSM a montré que quelque soit l'outil de développement utilisé, l'implémentation de la sémantique du langage occupe une partie majeure du code ([132], p.396).

### III.2.2 Définir sa sémantique

Définir la sémantique d'un langage est une tâche complexe qui nécessite **l'interprétation** des constructions du langage dans un champ sémantique cible [21]. L'exprimer de manière rigoureuse, lisible et utilisable est un défi majeur pour le concepteur du langage. Ainsi, lorsqu'un nouveau langage est défini, la sémantique est rarement explicitée ou formalisée. Dans les manuels de référence de l'OMG par exemple, la sémantique des diagrammes d'UML a initialement été définie informellement en langage naturel. Et il en est de même pour la sémantique du méta-modèle SPEM qui est aussi décrite en langage naturel [171]. Ce n'est que très récemment que la sémantique d'un sous-ensemble d'UML (fUML) a été rigoureusement spécifiée dans un document de 400 pages [170].

Dans le domaine de l'IDM, la question d'exécuter un modèle a rapidement été liée à celle de la définition explicite de sa sémantique. Breton et Bézivin (2001) ont été parmi les premiers à s'interroger sur comment combiner méta-modélisation statique (structures) et dynamique (sémantique) ; ils avaient alors proposé une expression *structurelle* de la sémantique sous forme d'un ensemble d'états et de liens entre ces états pour exprimer la sémantique en tant que passage d'un état vers un autre [29]. Par la suite, Muller et al. (2005) introduisent l'idée que les langages de méta-modélisation doivent contenir des constructions spécifiquement définies à cet effet, et permettre ainsi l'expression explicite de la sémantique d'un langage [162]. Ce besoin d'étendre les langages de méta-modélisation pour la prise en compte de la sémantique est ainsi intégré en tant que thématique fondamentale de recherche en méta-modélisation.

La Fig. III-3 illustre l'apport d'une **représentation explicite de la sémantique** d'un langage et son rôle dans la construction d'outils logiciels supports. Lorsque la sémantique n'est pas explicitement décrite avec le méta-modèle du langage (cf. Fig. III-3a), elle est implicitement codifiée de manière redondante dans chacun des outils logiciels supports tels que éditeur, générateur de code ou interpréteur. Par contre, une expression explicite de la sémantique réduit les risques d'incohérence et contribue à une spécification plus rigoureuse du langage car la spécification de la sémantique est associée au méta-modèle indépendamment des outils supports (cf. Fig. III-3b). Par ailleurs, cette spécification explicite de la sémantique peut être exploitée pour dériver les outils logiciels supports et automatiser, totalement ou partiellement, l'implémentation du langage [32], [70].

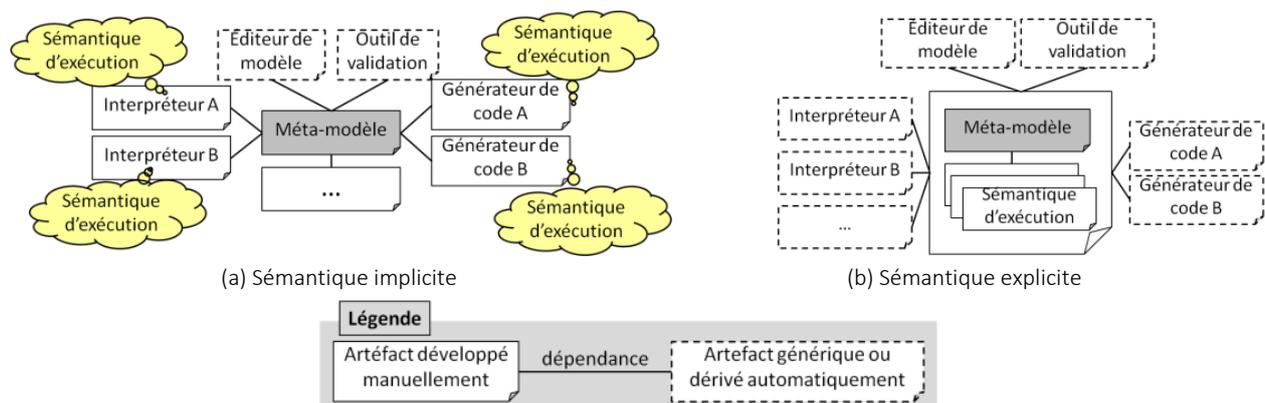


Fig. III-3. Représentation implicite et explicite de la sémantique d'un langage (d'après [153], p.57)

Dans l'univers des langages de programmation, les techniques développées pour exprimer la sémantique appartiennent à deux grands paradigmes d'interprétation : le **paradigme algébrique** pour l'approche axiomatique et l'approche *dénotationnelle*, et le **paradigme calculatoire** pour l'approche opérationnelle et l'approche *translationnelle* (par traduction) [246], [21], [128]. Dans l'univers de l'IDM et du DSM, de nombreux travaux de recherche ont exploré comment adapter ces techniques et les intégrer pour exprimer la sémantique dans les méta-modèles. On peut citer à titre d'exemples :

- l'utilisation des diagrammes UML dans [69],
- l'utilisation du langage de méta-modélisation KerMeta dans [46],
- l'utilisation du langage formel Maude dans [188],

- l'utilisation du langage des réseaux de Petri dans [45],
- ou encore l'utilisation du formalisme des machines à états abstraits dans [83].

Cependant, aucun consensus ne semble émerger, et intégrer l'expression de la sémantique d'un langage avec son méta-modèle reste un challenge [222], [32]. De plus, la plupart des travaux sont dans un état embryonnaire où les langages traités – en tant qu'exemple ou cas d'étude – sont généralement simples et les champs sémantiques cibles sont à un faible niveau d'abstraction :

- dans Combemale et al. (2009) par exemple [45], c'est la sémantique d'un sous-ensemble de SPEM qui est exprimée avec le formalisme du réseau de Petri ;
- dans Gargantini et al. (2009), c'est la sémantique d'un langage de machine à états finis (FSM) qui est exprimée dans le champ des machines à états abstraits (ASM) [83] ;
- enfin, dans l'approche xMOF de Mayerhofer et al. (2013), c'est le formalisme du réseau de Petri lui-même qui est traité dans une approche qui combine MOF et UML [153].

### III.2.3 Exécution d'un langage vs. *"process enactment"*

La question de l'exécution d'un modèle s'est aussi posée dans l'univers de la modélisation des processus d'ingénierie. Dès 1988, Osterweil utilise le terme « enactment » pour signifier « l'activation » ou « l'opérationnalisation » automatique d'un modèle de processus de développement logiciel [173]. Le sens de ce terme est subtilement différent de celui « d'exécution » car les modèles qui sont considérés décrivent des processus où plusieurs acteurs sont susceptibles d'intervenir, où la réalisation des tâches peut être réparties sur différents environnement techniques, et où la coordination et la synchronisation sont des enjeux de premier plan dans des processus longs et complexes [73], [62].

Ces modèles de processus d'ingénierie sont en fait analogues aux modèles des flux d'activité (« workflow ») et des processus métier, où la même problématique de « enactment » se pose [68]. Pour exprimer la sémantique des modèles de processus, le langage des réseaux de Petri est souvent choisi (p.ex. [59]), car il repose dispose d'un fondement formel. On retrouve ce même langage (réseaux de Petri) dans les travaux récents relatifs à l'expression de l'exécutabilité des modèles de processus d'ingénierie (p.ex. [16]).

### III.2.4 Conclusion et positionnement

Ce qui ressort de cette analyse de l'état de l'art est que l'expression et le traitement de l'exécutabilité d'un langage de modélisation sont des questions complexes qui relèvent en grande partie des sciences du formel (c.-à-d. de l'informatique théorique). Les solutions apportées sont, pour certaines, issues de l'univers des langages de programmation et de la compilation, alors que d'autres, font appel à des langages de modélisation des processus ayant eux même une sémantique formelle, tel que le langage des réseaux de Petri. Ainsi, la problématique de l'exécutabilité se rencontre dans le domaine de l'ingénierie des langages, dans le domaine de l'IDM, et dans celui de la modélisation des processus métier et des processus de développement logiciel.

Les travaux que j'ai menés concernant cette problématique de l'exécutabilité des modèles suivent trois directions complémentaires :

- La première (§III.3) a été abordée dans mon travail initial de thèse avec la construction d'un environnement logiciel complexe pour pouvoir « énoncer » le schéma conceptuel d'un SI.
- La seconde (§III.4) est une variante de la première, dans le sens où une architecture logicielle ad-hoc est définie pour exécuter des modèles conceptuels basés sur le concept d'intention. Néanmoins, la formalisation de la sémantique est explorée à travers l'utilisation – à petite échelle – du langage formel Z pour spécifier le fonctionnement de l'architecture.
- La troisième (§III.5) s'inspire des travaux dans le domaine de l'IDM pour définir la sémantique d'un langage à l'aide d'une notation événementielle, à partir de laquelle il devient possible de dériver l'architecture d'un outil d'exécution. Cette architecture repose sur l'utilisation de motifs génériques d'exécution distribuée du type « publier/souscrire ».

### III.3 Bref retour sur l'exécutabilité du modèle REMORA

Dans mon travail initial de thèse, j'avais déjà été confronté à une problématique similaire d'exécutabilité de modèle. Remora est un langage de modélisation conceptuel qui prône la représentation explicite des événements [195]. La notation graphique proposée permet de construire un schéma dynamique d'un SI. Rubis est un projet de type R&D mené sur la période 1984-1990 pour construire des outils logiciels autour de cette méthode [194]. C'est un Atelier de Génie Logiciel (AGL) avec un ensemble intégré d'outils pour modéliser, valider, exécuter et prototyper les spécifications conceptuelles d'une application construite avec la méthode Remora et le langage Proquel.

Un premier volet de mon travail au sein de l'équipe Rubis fut de participer pendant deux ans (1988-1990) à la conception du langage Proquel, et à la réalisation et l'implémentation du compilateur – interpréteur associé.

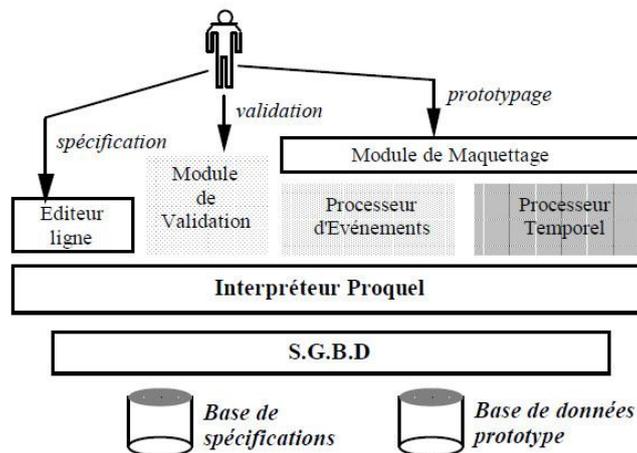


Fig. III-4. Architecture de l'atelier Rubis

Dans l'atelier Rubis, l'exécutabilité d'un modèle Remora est obtenue grâce à la **combinaison** de plusieurs composantes de l'atelier Rubis :

- ❖ Le modèle conceptuel de l'application est **directement spécifiable** dans le langage Proquel [publi#71, #64]; à chaque concept du modèle, c.-à-d. *c-objet*, *c-événement*, *c-opération*, *c-condition* et *c-facteur*, correspond des structures syntaxiques du langage pour spécifier le concept en détail. Grâce à cette fonctionnalité, le concepteur reste au niveau d'abstraction des concepts définis par le modèle.
- ❖ Le **langage Proquel** est directement exécutable grâce à un compilateur – interpréteur qui traduit les instructions du langage en un pseudo-code interprété par une machine virtuelle capable d'interagir avec une SGBD et de réaliser tous les calculs standards. Proquel est un langage de 4e génération (L4G) qui intègre les instructions du langage SQL avec celles d'un langage de programmation classique.
- ❖ Rubis inclut un outil pour la création automatique de **maquettes d'écrans** de saisie selon la structure des messages rattachés aux événements externes [147].
- ❖ Enfin, et au cœur de l'atelier, le **processeur d'événements** traite de manière dynamique l'arrivée de messages (événements externes) et l'occurrence d'événements internes [147]. Il fait systématiquement appel à l'interpréteur Proquel pour évaluer les prédicats des événements, les conditions et facteurs de déclenchement, et pour exécuter les opérations déclenchées.

Cette brève présentation met en évidence le besoin d'une architecture avec plusieurs composants logiciels pour arriver à opérationnaliser la sémantique exécutable d'un modèle conceptuel Remora. On constate aussi que la sémantique d'exécution du modèle n'est pas explicitement exprimée, et que l'approche dans sa globalité appartient donc à la catégorie où la sémantique est implicite (cf. Fig. III-3)

Dans le second volet de mon travail de thèse, et dans le but de transformer un schéma Remora en un ensemble sémantiquement équivalent de transactions sur un SGBD cible, j'ai indirectement abordé la question de l'expression explicite de la sémantique. J'ai en effet introduit un cadre général basé sur l'usage des techniques de méta-

modélisation pour formuler la transformation d'un modèle source vers un modèle cible (cf. Fig. III-5). Dans ce cadre, je préconise l'utilisation d'un langage de méta-modélisation pour décrire les produits (c.-à-d. la syntaxe abstraite des langages source et cible), et d'exprimer la transformation sous forme d'un modèle de processus. La sémantique du modèle est ainsi transcrite dans ce modèle du processus de transformation et de génération de code. La sémantique n'est toujours pas tout à fait explicite, elle est néanmoins intégrée dans la description d'un processus et n'est plus codifiée de manière ad-hoc dans des outils logiciels.

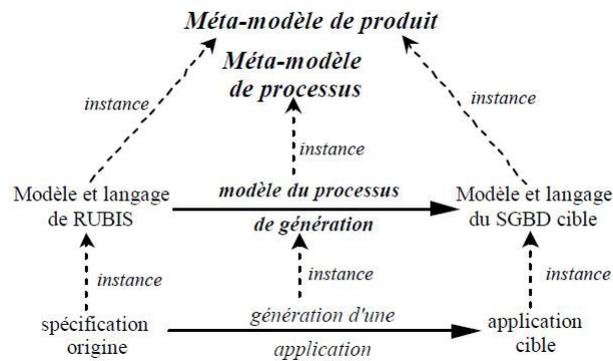


Fig. III-5. Cadre général proposé dans ma thèse de doctorat pour l'expression du processus de transformation de modèles ([7], p.9)

L'approche préconisée par cadre général est très similaire à celle introduite une décennie plus tard dans plusieurs travaux relatifs à l'ingénierie dirigée par les modèles (p.ex. [52] p.623, [119] p.129) ; la principale différence est que l'IDM considère la transformation comme un **algorithme**, défini certes de manière déclarative sous forme de règles, alors que je l'ai considéré comme étant **un processus** qui peut se réaliser de différentes manières en interaction avec un concepteur qui effectue des choix concernant l'implémentation de l'application cible.

### III.4 Une approche ad-hoc pour interpréter les cartes MAP

Ce travail a été réalisé dans le cadre de la thèse de Marc-Henri Edme en collaboration avec Prof. C. Rolland au CRI. La problématique de la thèse porte sur l'opérationnalisation d'un modèle intentionnel d'une carte MAP. La carte MAP a été introduite en tant que notation d'un haut niveau d'abstraction [197], elle vise à capturer les objectifs des acteurs et des organisations dans l'univers de discours [196].

La sémantique opérationnelle d'une carte MAP était faiblement définie ; cette absence de formalisation est un point fort de l'approche car elle autorise une grande flexibilité dans l'expression de processus complexe. Néanmoins, il était nécessaire de clarifier certains détails concernant la logique sous-jacente à une carte MAP, et c'est dans le cadre d'une première publication à ce propos que j'ai contribué à définir les premiers éléments concernant l'exécution d'une carte [publi#62] ([8]).

#### III.4.1 Contexte : modélisation intentionnelle et carte MAP

La carte Map est un formalisme qui repose sur le concept d'intention. Elle spécifie les modèles de processus sous forme de carte intentionnelle. Une carte est construite à l'aide de deux principaux concepts: l'Intention et la Stratégie. Une intention est un but à atteindre, ce but peut être atteint par plusieurs stratégies. Une stratégie est donc une manière concrète, opérationnelle, de réaliser une intention. Un autre concept dérivé de la carte est celui de la section ; une section de la carte est un triplet composé d'une intention source, d'une stratégie, et une intention cible (p. ex. <Spécifier une Classe, Manuellement, Spécifier une Association> dans la Fig. III-6g).

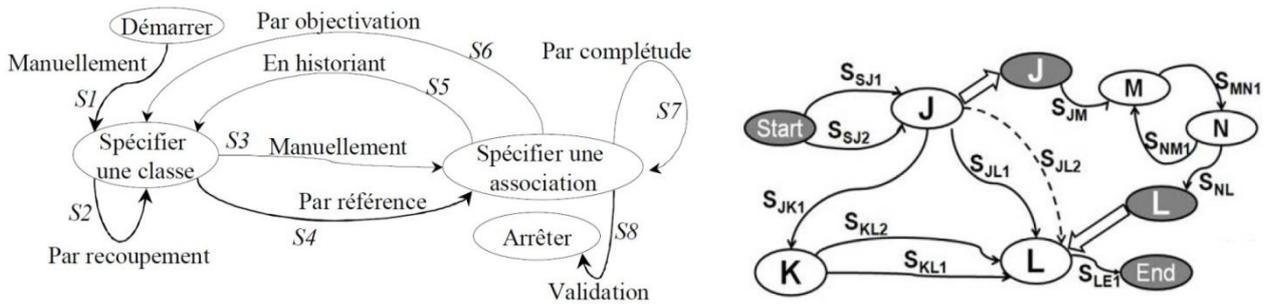


Fig. III-6. Deux exemples de carte MAP

Deux catégories de sections sont définies:

- une section atomique, dans laquelle la stratégie est opérationnelle car elle est associée à un code exécutable (une procédure, un composant, un service, etc.), p. ex. la section  $\langle J, S_{JK1}, K \rangle$  ;
- une section non-atomique correspond à une stratégie complexe de réalisation de l'intention nécessitant d'être affinée en une autre carte, p.ex. la section  $\langle J, S_{JL2}, L \rangle$  est affinée en une sous-carte comprenant les intentions M et N. Grâce à ce mécanisme d'affinement, le concepteur peut exprimer un processus à différents niveaux d'abstraction comme une hiérarchie de cartes.

Le modèle MAP a été présenté dans Rolland et al. (1999) comme étant un ordonnancement flexible d'intentions et de stratégies. Toute carte comporte en particulier une intention "Démarrer" et une intention "Arrêter", et représente un ensemble de chemins allant de "Démarrer" à "Arrêter". Chacun de ces chemins est à lui seul un modèle de processus. C'est pourquoi la carte est un multi-modèle. La représentation graphique d'une instance du modèle MAP est composée de nœuds (les intentions) et d'arcs orientés (les stratégies) reliant les nœuds. Chaque instance du modèle MAP est appelée une carte (ou un Map).

### III.4.2 La sémantique d'une carte Map

La sémantique opérationnelle du modèle intentionnel MAP ne correspond pas à un processus séquentiel dans lequel des tâches se succèdent (comme c'est le cas d'une machine à états finis) ; c'est plutôt une navigation dans un graphe suivant les intentions de l'utilisateur et selon le contexte situationnel par rapport à l'état du produit. Ceci autorise une grande variabilité dans l'ordonnancement de réalisation des intentions et dans le choix des stratégies à appliquer. Pour expliquer la sémantique opérationnelle d'une carte Map, je me base sur l'exemple dans Fig. III-6d :

- A partir de l'intention J (une fois que l'intention J a été réalisée), il y a deux chemins possibles pour progresser : atteindre l'intention K avec la stratégie  $S_{JK1}$ , ou bien réaliser l'intention L par la stratégie  $S_{JL1}$  ou la stratégie  $S_{JL2}$ .
- Dans certains cas, il est nécessaire de définir des contraintes de précédence et/ou d'exclusion entre les sections afin de restreindre l'ordre dans lequel les sections sont exécutées, ou pour exprimer des relations d'exclusivité entre les alternatives.

Pour l'exprimer plus précisément, la sémantique d'exécution d'une carte a nécessité des **références vers le produit**. En effet, la réalisation d'une intention correspond à l'**exécution d'une action** sur **une instance** d'un fragment d'un produit dans le cadre d'une stratégie spécifique [8]. En exécutant une section  $\langle J_i, S, L \rangle$ , nous entendons atteindre l'intention L, en utilisant la stratégie S, et à partir de  $J_i$  qui est une réalisation antérieure de l'intention J (identifiée à l'aide d'un index i). Cela signifie que la même section dans une carte peut être exécutée pour différentes instances d'un même fragment de produit. Si  $J_{i1}$ ,  $J_{i2}$  et  $J_{i3}$  sont les réalisations passées de l'intention J avec différentes instances d'un même fragment de produit (comme le montre la figure Fig. III-6d), alors les sections  $\langle J, S_{JK1}, K \rangle$ ,  $\langle J, S_{JL1}, L \rangle$  ou  $\langle J, S_{JL2}, L \rangle$  peuvent être exécutées pour chacune de ces réalisations passées, et le résultat sera les réalisations d'intention  $K_{i1}$ ,  $K_{i2}$  et  $K_{i3}$  pour l'intention K (ou  $L_{i1}$ ,  $L_{i2}$  et  $L_{i3}$  pour l'intention L).

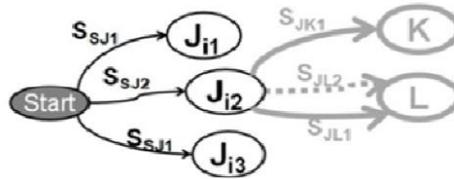


Fig. III-7. Illustration des sections candidates dans une carte Map

La combinaison d'une intention réalisée, d'une stratégie et d'une intention (p.ex. le triplet  $\langle J_{i1}, S, L \rangle$ ) est appelée une **section candidate** et c'est un concept essentiel pour exprimer la sémantique opérationnelle du MAP (Fig. III-7). A chaque exécution d'une section, une nouvelle collection de sections candidates (c.-à-d. des sections qui pourraient être exécutées dans l'étape suivante) doit être calculée.

Compte tenu d'un certain état du produit (c.-à-d. l'ensemble des instances du produit) et de l'historique des réalisations d'intentions, le calcul de l'ensemble des sections candidates se fait en calculant les sections qui correspondent aux chemins possibles et en vérifiant les contraintes. Comme cet ensemble se développe rapidement, différentes stratégies de progression peuvent être définies pour le réduire [165].

### III.4.3 Tentative de formalisation de la sémantique d'un MAP

Dans le travail de thèse de M.-H. Edme, nous avons cherché à définir précisément la sémantique du MAP à l'aide d'une notation semi-formelle. Cette spécification, de forme mathématique, est composée d'un ensemble de structures de données et d'opérations qui agissent sur ces structures. Ces spécifications définissent en fait l'architecture d'un outil d'exécution (cf. III.4.4), c'est une expression opérationnelle de la sémantique d'exécution du modèle MAP qui est ainsi exprimée. Cette spécification est composée d'un ensemble de 18 opérations, la figure Fig. III-8 présente un extrait de cette spécification du moteur d'exécution.

**Opération MAJ\_ReaIntInitiale\_Carte**

*Eléments requis*

- ExeCteDispo : une référence d'exécution de carte qui n'appartient pas encore à l'ensemble EXECUTION\_CARTE
- CodeCarteAExecuter : la référence de la carte qui doit être exécutée
- IdReaIntInitiale : la référence d'une réalisation d'intention qui initialise la nouvelle exécution de carte.

*Objectif*

Créer les premières candidates de la carte à exécuter.

*Prédicat*

$$\begin{aligned}
 \text{imIdSec} = & \{x \in \text{SECTIONS} / \\
 & \text{APPARTENIR}(x) = \text{CodeCarteAExecuter} \\
 & \wedge \text{ETRE\_SOURCE}(x) \in \text{POSSEDER}^+(\text{Demarrer}) \\
 & \wedge \neg (\exists y \in \text{CONTRAINTE} / \\
 & \quad \text{ATTEINDRE}(y) = x \\
 & \quad \wedge \text{CARACTERISER}(y) = \text{'Precedence'}) \\
 & \left. \right\} \\
 & \wedge \forall \text{IdSec} \in \text{imIdSec}, \text{Opération}(\text{Couleur}; \text{IdReaIntInitiale}, \text{IdSec}, \text{ExeCteDispo})
 \end{aligned}$$

*Fin prédicat*

*Interprétation textuelle du prédicat*

- imIdSec : ensemble des sections de la carte de référence CodeCarteAExecuter :
  - o ayant « démarrer » comme intention source,
  - o n'étant pas la cible d'une contrainte de précédence.
- Pour chacun des éléments IdSec de imIdSec, faire
  - o Utiliser l'opération Couleur avec les arguments (IdReaIntInitiale, IdSec, ExeCteDispo)

*Fin interprétation textuelle du prédicat*

*Fin opération*

Fig. III-8. Exemple de spécification d'opérations ([66], p.211)

### III.4.4 Une architecture de base de données pour exécuter les cartes

Le second volet du travail de thèse de M.-H. Edme a été d'utiliser la spécification formelle pour dériver l'architecture logicielle d'un outil d'exécution. La dérivation est basée sur un ensemble de règles élaborées en même temps que les spécifications elles-mêmes. Ces règles de dérivations permettent d'obtenir des structures algorithmiques à partir des spécifications déclaratives semi-formelles (cf. [66], p. 238).

L'application des règles de transformation aboutit à une architecture logicielle qui s'appuie sur un référentiel implémentée par une base de données relationnelle. La figure Fig. III-9 présente le schéma relationnelle de cette base de données. Le référentiel est l'organe logiciel qui va gérer le modèle d'une carte MAP, et qui renseigne le moteur d'exécution sur les intentions et stratégies qui apparaissent dans la carte en cours d'exécution.

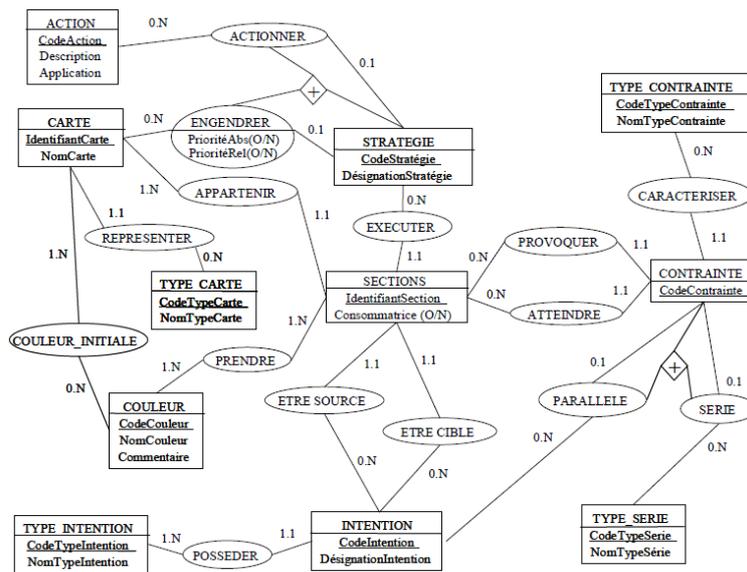


Fig. III-9. Modèle conceptuel des données pour le référentiel du moteur d'exécution des modèles de carte MAP ([66], p.130)

Comme l'exécutabilité de la carte repose sur le calcul des sections candidates (cf. §III.4.1), il est nécessaire de connaître les intentions qui ont été réalisées et les stratégies qui ont été exécutées, ainsi que les instances du modèle de produit qui sont associées. Ceci implique une **extension du référentiel** pour gérer les instances de la carte et garder **traces** de l'exécution ; la figure Fig. III-10 présente le schéma d'une telle extension.

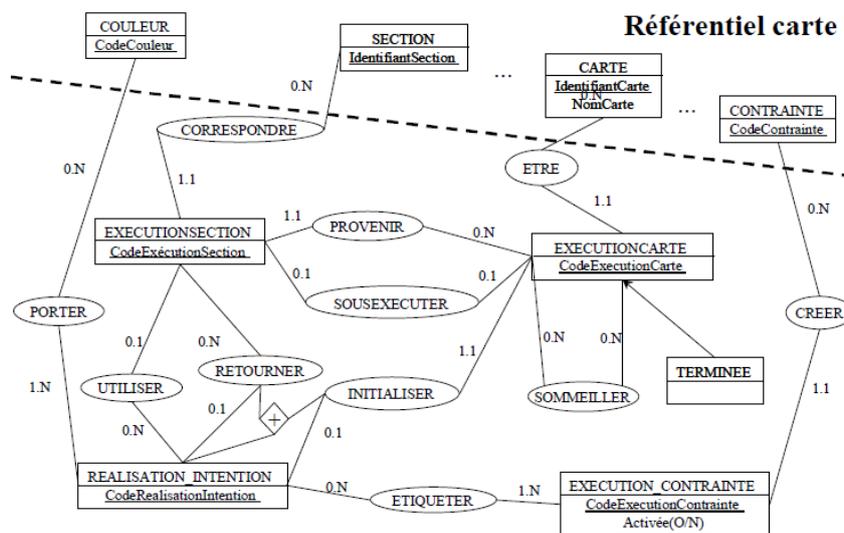


Fig. III-10. Modèle conceptuel des données de la trace d'exécution des cartes Map ([66], p.131)

Cette architecture a été implémentée dans un outil logiciel expérimental basé sur le SGBD Access et le langage de programmation VBA. Elle a été testée et montrée à l'aide d'un exemple de carte MAP qui décrit un processus de gestion des commandes de produits orthopédiques ([66], p.157).

### III.4.5 Conclusion sur l'approche ad-hoc

L'expression de la sémantique dans l'approche développée dans le travail de thèse de M-H. Edme repose sur la spécification d'une architecture logicielle. Par rapport à un travail antérieur ayant un objectif similaire (cf. [227]), l'originalité et la nouveauté de ce travail résident dans la spécification semi-formelle de cette architecture. Cette spécification fait appel à un langage de nature mathématique, ayant certes lui-même une sémantique claire, sans toutefois disposer d'une représentation graphique. De plus, l'architecture logicielle qui en est dérivée est assez rudimentaire, elle est construite à l'aide d'une base de données relationnelle et des formulaires d'interaction.

A partir de ce travail, ma démarche de recherche a été d'explorer des pistes plus sophistiquées pour exprimer la sémantique des cartes Map et pour dériver l'architecture logicielle d'un moteur d'exécution. C'est l'objet de la prochaine section.

## III.5 Vers une approche générique pour l'expression de l'exécutabilité

Le travail présenté dans cette section a été réalisé dans le cadre de la thèse de Sana Mallouli en collaboration avec la Prof. C. Souveyet au CRI. L'objectif de cette thèse est d'aller plus loin dans la maîtrise du processus d'expression de la sémantique d'un modèle et de la dérivation d'un outil logiciel d'exécution.

Une première idée directrice a été de chercher à exploiter les outils de type méta-CASE, capables de prendre en compte la spécification d'un modèle sous forme de méta-modèle. Notre choix s'est porté sur l'outil Meta-Edit commercialisé par l'entreprise MetaCase<sup>19</sup> et son langage de méta-modélisation GOPRR<sup>20</sup> [123]. Nous avons ainsi mené un projet exploratoire dans lequel nous avons utilisé Meta-Edit pour tenter de construire un outil d'exécution pour les cartes Map (cf. [100], [publi#34]).

Nous avons ainsi pu constater quelques limites telles que le faible pouvoir d'expression du langage de méta-modélisation ; ce qui fait que la spécification faite avec Meta-Edit est insuffisante pour générer automatiquement tout le code de l'architecture logicielle du moteur d'exécution. A partir de ce constat, nous avons défini une liste de cinq exigences que les environnements de type metaCASE devraient satisfaire. Toutes ces exigences portent sur le langage et l'environnement de méta-modélisation, car les possibilités d'automatisation de la construction de l'outil dépendent directement de la richesse d'expression du méta-modèle et de son exploitabilité.

C'est ainsi que le travail de thèse de S. Mallouli a pris la direction de chercher une nouvelle approche pour l'expression de la sémantique d'un langage et pour la dérivation d'une architecture logicielle.

### III.5.1 Une approche de type IDM pour définir l'architecture d'exécution

La démarche proposée dans ce travail se démarque de l'état de l'art sur plusieurs points :

- Elle adopte certes le paradigme calculatoire pour interpréter le méta-modèle, cependant, la technique proposée cible le champ sémantique des systèmes interactifs et introduit une spécification événementielle de la sémantique d'exécution d'un langage. En effet, nous nous intéressons en particulier aux modèles de processus métier ou d'ingénierie qui décrivent des activités humaines assistées par l'informatique et qui manipulent divers artefacts ;

- C'est une démarche d'ingénierie dirigée par les modèles qui comporte une étape de **transformation** de la spécification événementielle de la sémantique vers une architecture logicielle d'un outil d'exécution de modèles. La transformation cible des motifs génériques d'exécution (des « patterns ») de type publier/souscrire [71]. Ces motifs architecturaux sont adaptés à la programmation de systèmes asynchrones faiblement couplés [105]. Utilisés ici pour transcrire

<sup>19</sup> <http://www.metacase.com/>

<sup>20</sup> GOPRR : Graph, Object, Property, Relationship, Role

la sémantique d'exécution d'un modèle de processus, ils permettent une spécification logicielle qui respecte la nature événementielle de celle-ci.

– Enfin, cette approche est appliquée au langage de modélisation MAP pour spécifier une sémantique non séquentielle qui repose sur l'interaction avec l'utilisateur et avec l'environnement du système. Ce choix est conforme à des recommandations de concepteur d'outils et d'environnement IDM qui préconisent de travailler à partir d'un scénario riche et complexe pour stimuler l'exploration de l'espace du problème et celui des solutions ([174], p.534).

### III.5.2 Modélisation événementielle de la sémantique

Pour élaborer notre approche, nous avons cherché une spécification de la sémantique selon plusieurs contraintes :

- Elle doit être de nature déclarative pour faciliter son élaboration.
- Elle doit avoir une représentation graphique qui facilite sa lecture et sa compréhension.
- Elle doit posséder une sémantique claire.
- Cette spécification doit être suffisamment riche et précise pour qu'il soit possible d'en déduire, par génération de code, un outil d'exécution de modèles.

D'autre part, et de par sa nature interactive, la sémantique d'exécution d'un modèle de processus souvent implique la description de réactions suite à des échanges de messages avec un environnement et à des changements d'états des objets manipulés par le processus. Nous considérons donc que **l'outil d'exécution est par essence un système interactif** qui réagit à des occurrences d'événements. Pour capturer cette sémantique, il nous semble donc plus naturel et pertinent de recourir au concept d'événement et à la modélisation événementielle. Ceci nous a amené à choisir le formalisme dynamique de la méthode Remora (Rolland et al., 1988) pour sa concision (peu de concepts) et son expressivité pour la vue systémique (événement et échange de messages).

La figure Fig. III-11 décrit notre approche pour la spécification d'un langage de modélisation et la construction d'un outil d'exécution. Notre approche s'élabore selon une logique d'IDM et plus précisément selon l'architecture MDA de l'IDM. En effet, notre objectif étant de construire des outils logiciels (pour exécuter des modèles de processus), il nous paraît naturel de favoriser l'usage de modèles et d'exploiter ainsi la démarche IDM.

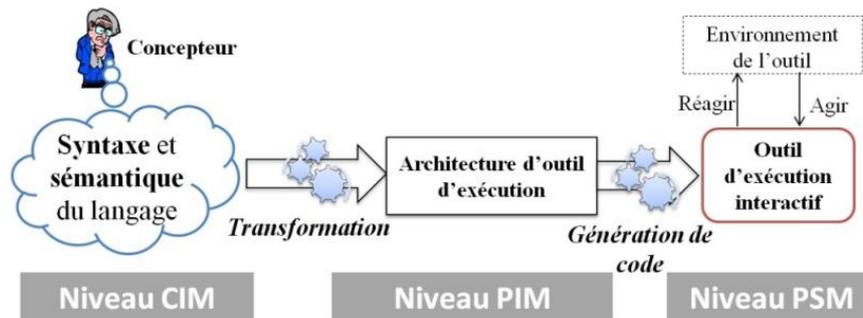


Fig. III-11. Synopsis de l'approche

La suite de cette section présente les trois étapes de la démarche, le niveau CIM et introduit la spécification événementielle de la sémantique d'exécution du méta-modèle. Le passage au niveau PIM et la transformation de cette spécification sont présentés à la section §III.5.3.

#### Étape 1 : élaborer le méta-modèle statique

Dans la première étape, le concepteur définit le méta-modèle structurel, i.e. la syntaxe abstraite du langage, sous forme d'un diagramme de classes UML. Cette étape permet de décrire l'ensemble des concepts du méta-modèle, leurs interconnexions et les méthodes nécessaires à leur usage.

#### Étape 2 : introduire le schéma à deux niveaux et ajout d'attributs

Dans notre approche, la sémantique d'exécution d'un modèle va être interprétée selon une logique opérationnelle de fonctionnement d'un outil d'exécution. L'expression de cette logique requière des références aux instances des

concepts définis dans le méta-modèle statique. De ce fait, on commence par rajouter aux méta-classes du diagramme UML un ensemble de classes pour **représenter les instances** des concepts du modèle. Ainsi, à chaque méta-classe est définie par extension une classe d'instances. Cet ensemble de classes et de méta-classes, organisé en deux niveaux d'abstraction, contient les objets sur lesquels va porter le schéma événementiel Remora.

**Étape 3 : modélisation événementielle du comportement des concepts**

A l'étape suivante, le concepteur exprime la sémantique d'exécution du langage à l'aide d'un schéma événementiel Remora. Dans ce schéma, l'opérationnalisation de la sémantique d'exécution repose sur les changements d'état constatés sur les classes d'instances (événements internes) et sur les interactions avec l'environnement d'exécution (événements externes). Ces événements déclenchent des opérations qui mettent à jour des objets du niveau instance, assurent l'envoi et la réception de messages, et la demande d'exécution de composants logiciels externes.

**III.5.3 Exploitation de la sémantique opérationnelle**

Cette section a pour objet de montrer la démarche à déployer pour extraire une architecture exécutable à partir de la spécification conceptuelle (c.-à-d. le méta-modèle et le schéma événementiel) d'un langage de modélisation de processus. Avant d'expliquer comment transcrire au niveau technique le concept d'événement, il faut introduire la structure générale du moteur d'exécution que l'on peut construire à partir de cette spécification.

Au niveau de la transcription du concept d'événement, nous allons utiliser le motif générique « publier/souscrire » issu de l'univers de la programmation distribuée, et considéré comme le paradigme adéquat pour le développement d'applications asynchrones et réactives [105]. La raison principale derrière ce choix est que ces motifs génériques permettent d'implémenter une exécution simple sous forme de réactions à des occurrences d'événements sans tenir compte de l'ordre dans lequel ils surviennent.

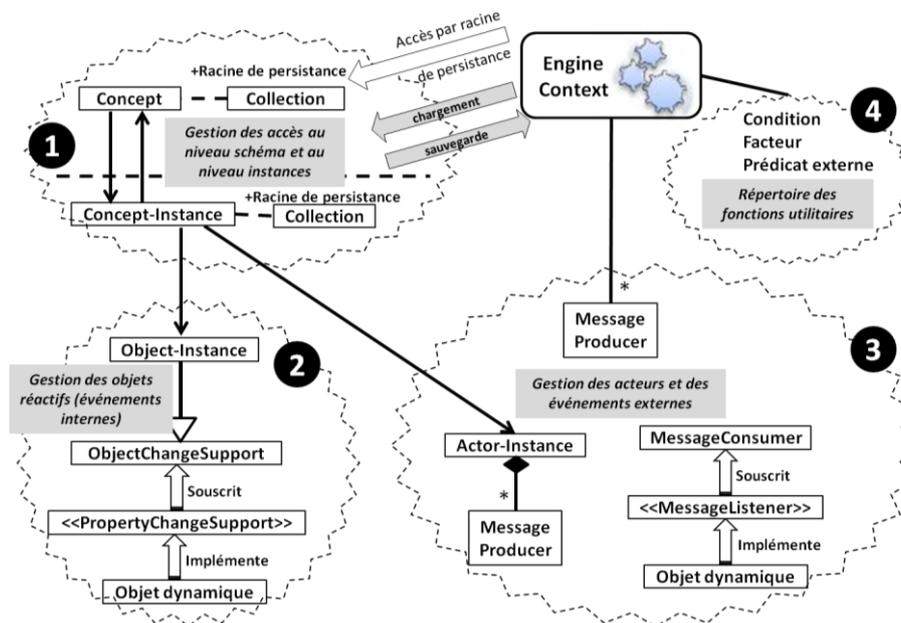


Fig. III-12. Structure générale du moteur d'exécution

Le principe fondamental sous-jacent à « publier/souscrire » est le suivant : un objet (appelé « objet dynamique ») qui doit réagir à l'occurrence d'un événement constaté sur un objet « observé » va souscrire à cet événement. A chaque occurrence de celui-ci, l'objet observé publie l'occurrence d'événement aux objets abonnés. Le traitement spécifique inclus dans chaque objet abonné est exécuté à chaque notification d'occurrence de l'événement.

La structure générale de l'architecture du moteur d'exécution est présentée dans la Fig. III-12. Cette structure a été conçue après analyse et exploration de l'espace des solutions logicielles possibles pour traduire la sémantique d'exécution d'un schéma événementiel Remora. L'architecture générale du moteur d'exécution comporte 4 parties principales (cf. Fig. III-12) :

- la partie 1 correspond à la gestion des données du niveau schéma et du niveau instance en deux niveaux d'abstraction inter-reliés,
- la partie 2 prend en charge les événements internes associés aux objets de niveau instance,
- la partie 3 permet de gérer les acteurs et les événements externes de la spécification,
- la partie 4 rassemble l'implémentation des conditions, des facteurs et des prédicats des événements externes dans un répertoire de fonctions utilitaires.

La dérivation de cette architecture se fait appel à l'aide de trois règles principales de transformation R1, R2 et R3 détaillant respectivement la traduction d'un événement interne, d'un événement externe et une opération externe. L'application successive de ces trois règles aboutit à la structure présentée dans la Fig. III-12. Ces règles ne sont intégralement pas décrites ici, uniquement la règle 1 sera présentée brièvement dans la prochaine sous-section.

**Exemple d'une règle de transformation**

Le concept d'événement interne exprime les changements d'états d'un objet qui nécessitent le déclenchement de traitement (voir partie (b) de la Fig. III-13). Ce traitement est décrit dans les opérations déclenchées par l'événement, alors que le changement d'état est décrit dans le prédicat de l'événement. Le prédicat d'un événement se définit avec l'état *avant* et l'état *après* d'une seule propriété d'objet.

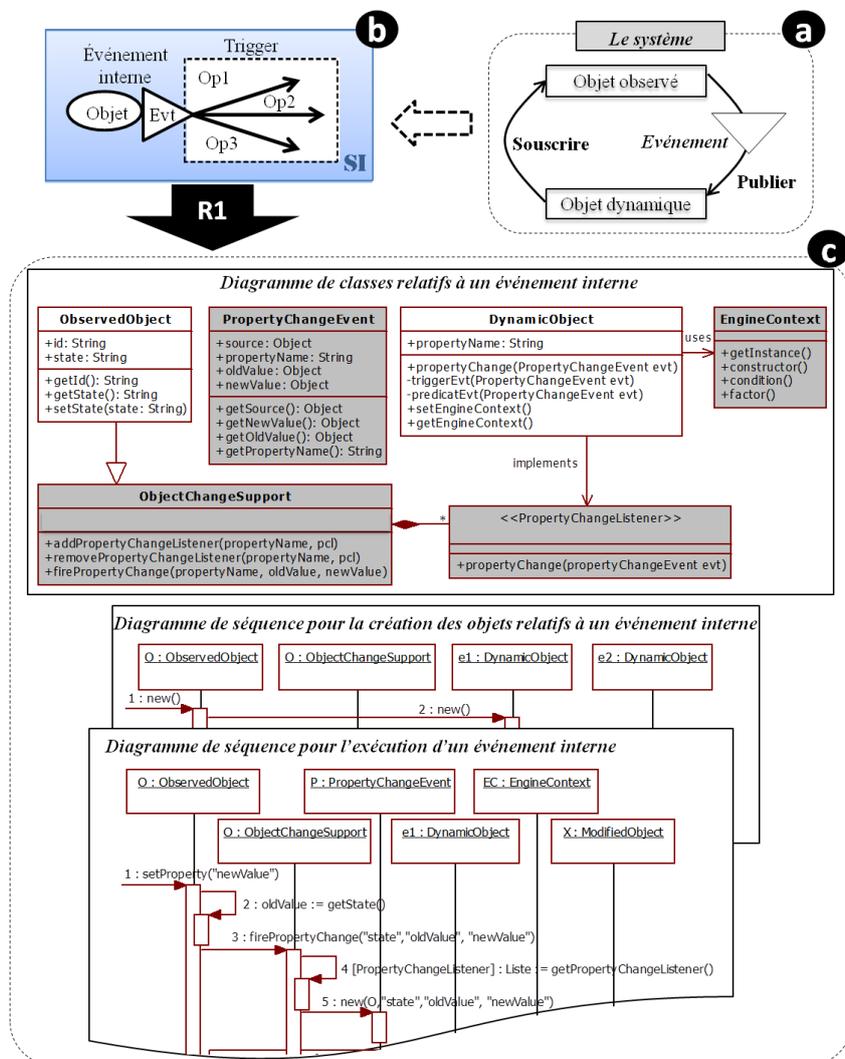


Fig. III-13. Transcription de l'événement interne en une spécification orientée objet

L'implémentation des événements internes Remora que nous proposons, s'inspire de la technologie des JavaBeans qui permet à des objets de type JavaBeans de publier des événements en fonction des changements d'états d'une de leurs propriétés. Des objets de type *PropertyChangeListener* peuvent s'abonner à ces événements afin de déclencher les traitements appropriés.

La partie (c) de la Fig. III-13 explique la spécification conceptuelle UML (extrait de schéma de classe et diagrammes de séquence associés) obtenue à l'issue de la dérivation R1 exploitant la spécification Remora d'un événement interne (partie (b) de la Fig. III-13 ). Les autres règles de dérivation sont présentées en détail dans [publi#04].

### III.5.4 Application sur le modèle de carte MAP

La solution développée dans cette section a été appliquée sur le modèle de carte MAP. La Fig. III-14 présente le résultat de l'application des deux premières étapes (cf. Fig. III-12). Le méta-modèle statique s'organise autour du concept de **section** (cf. Fig. III-6). Une section est associée au concept abstrait *ImplemExeSection* qui renseigne sur la manière d'exécuter la stratégie associée à une section (service web, ensemble de directive ou application externe). Une section est aussi reliée à un produit initial source sur lequel porte l'exécution de la section et à un produit final cible qui résulte de cette exécution. Pour opérationnaliser cette structure statique, on commence par ajouter des structures d'instances sous forme d'un ensemble de classes-instances.

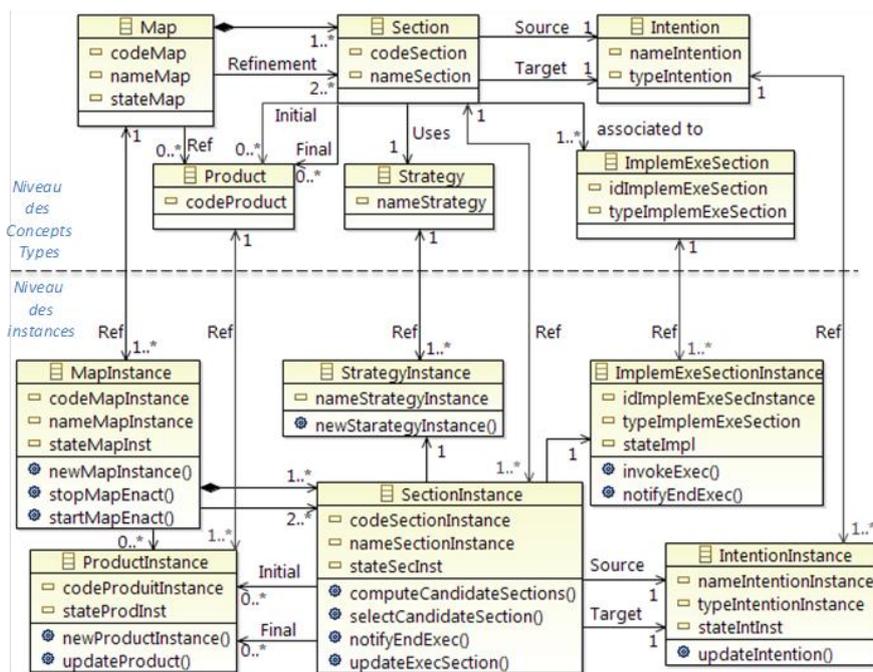


Fig. III-14. Modèle structurel à deux niveaux (concepts-types et instances) du Map

Ensuite, en se basant sur la sémantique d'exécution du modèle MAP, on complète par des méthodes et des attributs d'états nécessaires pour l'exécution du moteur. Certains de ces ajouts se font lors de l'étape suivante lors de la modélisation événementielle du comportement des concepts. La Fig. III-15 présente le résultat de cette étape. Ce schéma événementiel Remora exprime la sémantique d'exécution du MAP. Dans ce schéma, on considère chaque classe du modèle structurel comme un objet susceptible d'avoir des changements d'états et sur lequel peuvent être constatés des événements. Nous définissons par ailleurs trois acteurs externes :

- *MapActor* pour représenter l'utilisateur final qui exécute une carte ;
- *StrategyApplicationExecutor* pour un agent externe qui exécute les actions atomiques associées aux stratégies ;
- *ProductManager* pour un agent qui gère le contenu du produit et la trace d'exécution.

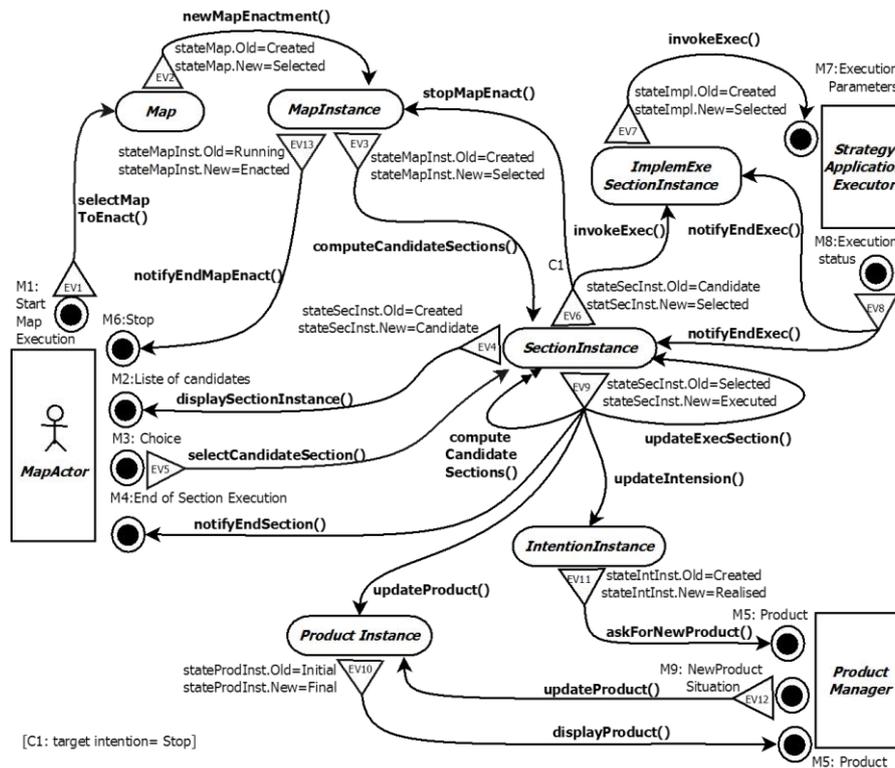


Fig. III-15. Spécification événementielle de la sémantique d'exécution du Map

Après l'application de l'ensemble des règles de dérivation, nous obtenons le schéma de classe présenté à l'annexe 1. Ce schéma de classe est complété par un ensemble de diagrammes de séquences (non présentés) donnant une spécification exécutable complète du moteur d'exécution.

### III.6 Conclusion et perspectives de recherche

Le développement et l'implémentation d'un nouveau langage de programmation ou de modélisation se heurtent à la question ardue de l'expression de sa sémantique, en particulier lorsque cette sémantique est exécutable comme c'est le cas avec les langages de modélisation de processus. Par rapport aux approches existantes, j'ai dans ces travaux exploré deux techniques de résolution. Ces techniques reposent sur l'usage de la méta-modélisation et de la transformation de modèles. Mon travail initial en thèse et celui co-encadré dans la thèse de M-H. Edme considèrent une approche ad-hoc où l'exécutabilité est obtenue en construisant une architecture logicielle spécifique capable d'interpréter les modèles. Dans le cas de Rubis, l'interpréteur du langage Proquel et le processeur d'événement collaborent pour exécuter un schéma Remora ; dans le cas du prototype de M-H. Edme, le « moteur de processus intentionnels » fait appel à un « éditeur de produit » pour réaliser le « enactment » d'une carte Map.

Ces deux approches restent spécifiques et sans spécification explicite de la sémantique des langages sous-jacents. J'ai ainsi cherché à surmonter cette limitation, et la thèse de S. Mallouli a exploré, d'une part, la possibilité d'utiliser les outils méta-CASE existants, et d'autre part, la possibilité de définir un langage de méta-modélisation avec lequel il est possible de décrire explicitement la sémantique et d'avoir une représentation graphique de cette spécification. C'est le langage Remora lui-même que nous avons proposé d'utiliser. L'expression de la sémantique sous forme d'un schéma événementiel est la principale originalité de ce travail. Comme tout travail d'ingénierie qui explore de nouvelles approches pour résoudre un problème, la **faisabilité technique** et **l'utilité concrète** restent à démontrer.

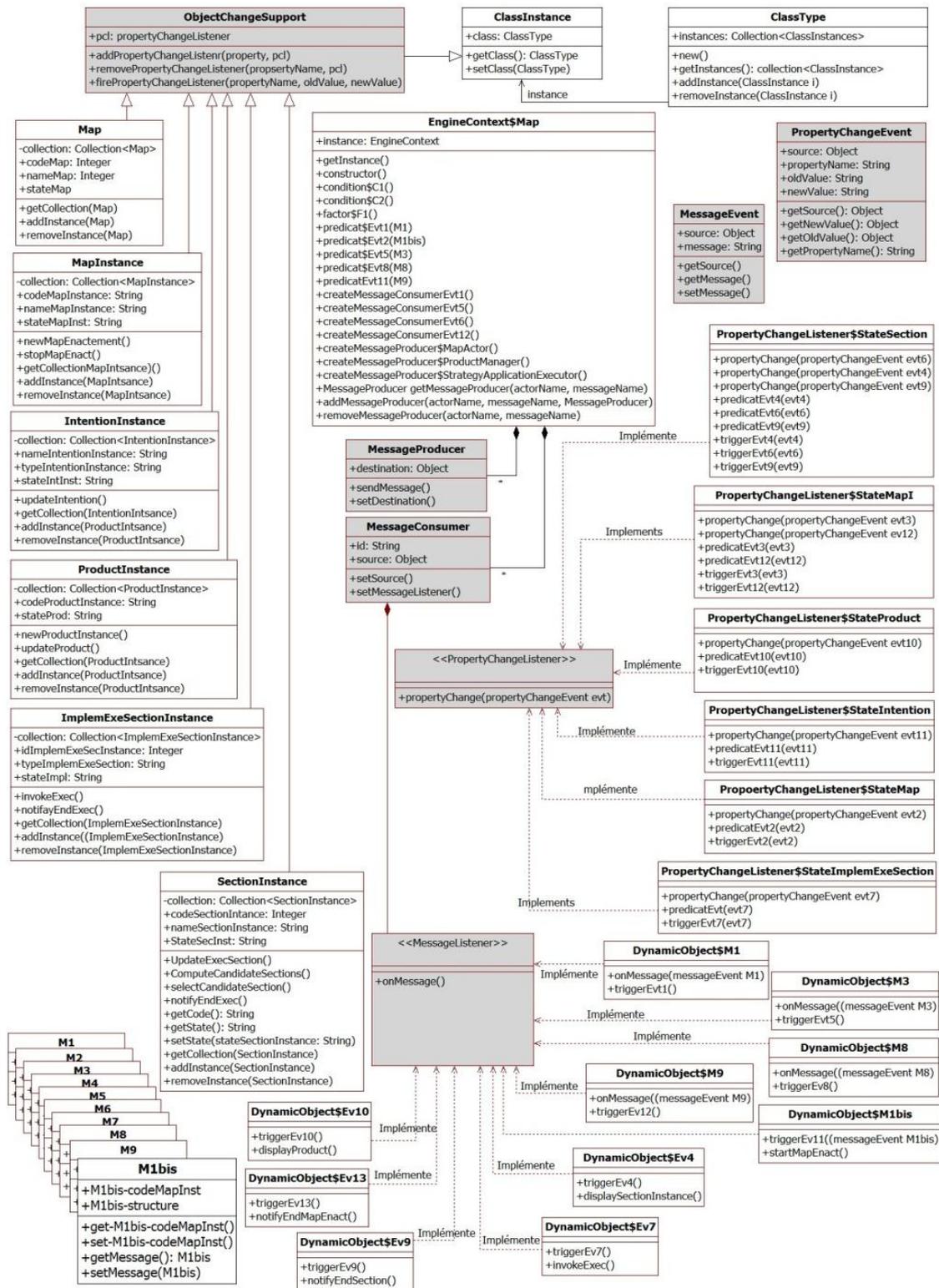
Néanmoins, et en comparaison avec l'état de l'art dans le domaine de l'exécution de modèles conceptuels, on constate que l'exécutabilité des modèles orienté but n'est pas envisagée. Dans l'analyse des travaux de l'atelier ModRE<sup>21</sup> que j'ai réalisé en 2014 (publi #26), j'ai pu constater que même si le thème général de l'atelier est

<sup>21</sup> ModRE'14 : Model Driven Requirement Engineering (<http://www.modre2014.ece.mcgill.ca/>)

d'appliquer l'approche IDM dans le contexte de l'ingénierie des besoins, les langages de modélisation orienté but ne sont pas traités ; constat confirmé dans des publications très récentes (p. ex. [84]). A ce jour, les travaux les plus avancés pour construire des systèmes, à partir de spécifications conceptuelles et des besoins de l'utilisateur, s'appuient sur des langages de modélisation conventionnels sans prise en compte explicite du concept de but (p.ex. [179]).

Plusieurs travaux futurs sont envisagés pour ce thème de recherche. Le premier concerne d'une part, la formalisation des règles de transformation utilisé dans la thèse de S. Mallouli avec un formalisme tel que le langage ATL [118], et d'autre part, le développement d'un prototype avec des outils logiciels adéquats. Le second concerne la généralisation de la démarche présentée ici vers d'autres langages de modélisation. Je pense en premier au méta-modèle SPEM où la vision systémique et événementielle peut contribuer à une meilleure prise en compte de l'exécution distribuée d'un processus de développement. Je pense ensuite à d'autres langages de modélisation orientés but, tel que le langage KAOS [136]. Enfin, la complexité de la démarche nécessite une réflexion sur comment guider et aider l'ingénieur dans l'élaboration de la spécification de la sémantique.

### III.7 Annexe 1 : Diagramme de classe du moteur d'exécution d'un modèle de carte (niveau instances)



# IV. Chapitre IV : Exploitation des rapports d'erreurs logicielles pour la prédiction du temps de résolution

## Sommaire

<a href="#"><u>IV.1</u></a>	<a href="#"><u>Préambule</u></a>	58
<a href="#"><u>IV.2</u></a>	<a href="#"><u>Gestion et traitement des erreurs logicielles</u></a>	59
<a href="#"><u>IV.3</u></a>	<a href="#"><u>Bref état de l'art sur la prédiction du temps de résolution des erreurs logicielles</u></a>	60
<a href="#"><u>IV.3.1</u></a>	<a href="#"><u>Apprentissage supervisé avec la technique <math>\alpha</math>-kNN</u></a>	60
<a href="#"><u>IV.3.2</u></a>	<a href="#"><u>Apprentissage non supervisé avec la technique du partitionnement</u></a>	61
<a href="#"><u>IV.4</u></a>	<a href="#"><u>Problématique</u></a>	62
<a href="#"><u>IV.4.1</u></a>	<a href="#"><u>Brève présentation de l'étude initiale</u></a>	62
<a href="#"><u>IV.4.2</u></a>	<a href="#"><u>Estimation de la répliquabilité de l'étude initiale</u></a>	63
<a href="#"><u>IV.4.3</u></a>	<a href="#"><u>Techniques de simulation pour le test d'hypothèse</u></a>	64
<a href="#"><u>IV.5</u></a>	<a href="#"><u>Partie 1 : répliquabilité de l'étude Raja (2013)</u></a>	64
<a href="#"><u>IV.5.1</u></a>	<a href="#"><u>Description des jeux de données</u></a>	65
<a href="#"><u>IV.5.2</u></a>	<a href="#"><u>Le protocole expérimental de la répliquabilité</u></a>	66
<a href="#"><u>IV.5.3</u></a>	<a href="#"><u>Les résultats de la répliquabilité</u></a>	66
<a href="#"><u>IV.5.4</u></a>	<a href="#"><u>Conclusion sur la partie 1</u></a>	68
<a href="#"><u>IV.6</u></a>	<a href="#"><u>Partie 2 : test opérationnel de l'hypothèse émise par Raja (2013)</u></a>	69
<a href="#"><u>IV.6.1</u></a>	<a href="#"><u>Le protocole expérimental du scénario de simulation</u></a>	69
<a href="#"><u>IV.6.2</u></a>	<a href="#"><u>Premiers résultats</u></a>	72
<a href="#"><u>IV.6.3</u></a>	<a href="#"><u>Modification de la taille de l'ensemble de test</u></a>	74
<a href="#"><u>IV.6.4</u></a>	<a href="#"><u>Augmentation du nombre de partitions (<math>K = 6, 8</math> et <math>10</math>)</u></a>	75
<a href="#"><u>IV.6.5</u></a>	<a href="#"><u>Conclusion sur la partie 2</u></a>	76
<a href="#"><u>IV.7</u></a>	<a href="#"><u>Conclusion et perspectives de recherche</u></a>	78

## IV.1 Préambule

Le travail présenté dans ce chapitre relève des sciences des faits, et concerne une problématique liée au développement de systèmes logiciels. Il s'appuie sur une collaboration initiée lors d'un séjour sabbatique à l'université de Lund en Suède au printemps 2012. L'équipe d'accueil (SERG<sup>22</sup>) est reconnue internationalement pour son expertise dans les études empiriques en génie logiciel, et l'équipe a publié deux ouvrages méthodologiques de référence sur le sujet<sup>23, 24</sup>. C'est naturellement donc que la collaboration s'est orientée vers une problématique de génie logiciel susceptible d'être traitée par une approche méthodologique de type empirique. La collaboration s'est focalisée sur le traitement des **rapports textuels d'erreurs logicielles** (« *textual bug report* »). L'équipe SERG avait en effet une bonne maîtrise des outils techniques nécessaires à ce genre de traitement, et avait réalisé des travaux significatifs à ce sujet (p.ex. [203], [24]). Par ailleurs, un des co-auteurs de ce travail disposait de plusieurs jeux de données issus de projets logiciels libres (Android, Eclipse) et d'un projet logiciel propriétaire réalisé dans une compagnie industrielle en Suède (Company A).

Nous nous sommes rapidement focalisés sur la question de la prédiction du temps de résolution d'une erreur logicielle nouvellement arrivée, de par l'importance de cette question pour la maintenance des logiciels. Une revue rapide de la littérature nous a orientés vers les techniques de fouille textuelle (« *text mining* ») et l'exploitation des textes des rapports pour prédire le temps de résolution. L'idée fondamentale sous-jacente est que si deux rapports d'erreurs sont **textuellement similaires**, alors ils décrivent des **erreurs analogues** et le temps nécessaires pour les résoudre sera vraisemblablement proche. Cette hypothèse forte avait déjà été formulée pour l'estimation des charges dans le domaine de gestion de projet [211], et elle est au fondement du raisonnement basé sur les cas (« *Case-Based Reasoning* ») [51]. Elle suppose néanmoins la définition d'une mesure précise de la similarité et d'une approche méthodologique pour la mise en œuvre opérationnelle de cette hypothèse.

Une autre direction de recherche a orienté le travail que nous avons réalisé, il s'agit de la **réplication d'études empiriques**. J'ai déjà évoqué cette question dans le chapitre II (cf. §II.6.6) : la réplication est la répétition d'une recherche empirique ou expérimentale pour vérifier ses résultats et augmenter ainsi la confiance qu'on y accorde. Une étude récente venait de paraître dans la revue *Empirical Software Engineering* ; cette étude (qu'on appellera Raja (2013)) utilise les techniques de fouille textuelle et d'apprentissage non supervisé pour analyser les rapports d'erreurs logicielles [182]. Les résultats de cette étude ont attiré notre attention, et en particulier, la proposition faite par l'auteur : la catégorisation automatique des rapports textuels selon une technique d'apprentissage automatique non supervisé, pourrait être exploitée pour **prédire le temps de résolution** d'une erreur lorsque celle-ci est déclarée.

Nous nous sommes donc penchés sur l'étude de Raja (2013) et avons entrepris un travail de recherche autour de deux questions :

- RQ1: Les résultats de l'expérience rapportée dans Raja (2013) sont-ils répliquables en utilisant une technique de partitionnement entièrement automatisée ?
- RQ2: Quelle serait la précision d'une telle technique (c.-à-d. basée sur le partitionnement automatique des rapports d'erreurs) pour la prédiction de temps de résolution d'une erreur ?

Ce chapitre présente le travail réalisé pour répondre à ces deux questions. Il présente un bref état de l'art, les deux protocoles expérimentaux et les résultats obtenus, avec un focus sur la seconde question de recherche (RQ2). Il est basé sur l'article paru en ligne en 2015 (et publié en 2016) dans la revue *Empirical Software Engineering* (publi#02).

<sup>22</sup> SERG : Software Engineering Research Group, <http://serg.cs.lth.se/>

<sup>23</sup> Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., & Wesslén, A. (2000). *Experimentation in software engineering: an introduction*. Norwell, MA, USA: Kluwer Academic Publishers. [réédité en 2012]

<sup>24</sup> Runeson, P., Rainer, A., Höst, M., & Regnell, B. (2012). *Case Study Research in Software Engineering: Guidelines and Examples*. John Wiley & Sons.

## IV.2 Gestion et traitement des erreurs logicielles

La gestion des erreurs logicielles est un aspect essentiel dans la phase de maintenance du cycle de vie des applications informatiques. La détection et l'élimination de ces erreurs nécessitent des efforts importants [22]; le coût des erreurs logicielles à l'économie américaine était estimé à 59,5 milliards \$US par année [225]. Lorsqu'une erreur est détectée, des ressources appropriées doivent être allouées pour l'identifier, la localiser et la corriger. L'allocation des ressources est basée sur des estimations de la gravité et la criticité de l'erreur et de son impact potentiel, ainsi que sur la disponibilité des ressources pour la traiter. Un aspect important à considérer dans l'allocation des ressources est le **Temps de Résolution d'une Erreur (TRE)**, c.-à-d. le temps estimé nécessaire pour corriger l'erreur et résoudre le problème. Le TRE reflète également le temps que les utilisateurs finaux devront attendre avant de recevoir le logiciel corrigé, un paramètre particulièrement important dans le cas de logiciels développés en déploiement continu. La prédiction du temps de résolution pour des erreurs nouvellement soumises est essentielle pour les gestionnaires de projets, car elle détermine l'allocation des ressources et la planification de la distribution du logiciel.

La Fig. IV-1 décrit le cycle de vie général d'un rapport d'erreur logicielle. Les systèmes de gestion des erreurs permettent aux utilisateurs de signaler, suivre, décrire, commenter et classer les rapports d'erreurs. Un rapport d'erreur est livré avec un certain nombre de champs prédéfinis tels que le produit logiciel concerné, la version du système d'exploitation et la gravité auto-déclarée de l'incident, ainsi que des champs en texte libre tels que le titre de l'erreur et sa description. En outre, les utilisateurs et les développeurs peuvent laisser des commentaires et inclure des pièces jointes, qui prennent souvent la forme de modification de code (« patches ») ou des captures d'écran. Lorsqu'une erreur est initialement déclarée, elle est en état d'attente non confirmée jusqu'à ce qu'une première évaluation soit faite pour déterminer si le rapport est valide et que l'erreur n'est pas déjà connue, c.-à-d. le rapport d'erreur ne soit pas une duplication d'un rapport d'erreur déjà traité. Les rapports d'erreurs peuvent passer à travers plusieurs étapes avant d'être finalement résolus. Les rapports d'erreurs qui sont résolus reçoivent une des désignations suivantes: *duplication, invalide, résolu, insoluble* ou « *marche pour moi* ». Ceux-ci indiquent la raison pour laquelle le rapport a été terminé; par exemple, « marche pour moi » et invalide indiquent qu'on a été incapable de reproduire le problème décrit dans le rapport. Parfois, un rapport d'erreur doit être ré-ouvert (cf. arcs gris dans Fig. IV-1). Dans ce cas, le cycle de vie de l'erreur (cf. arcs noirs) commence avec le statut «réouvert».

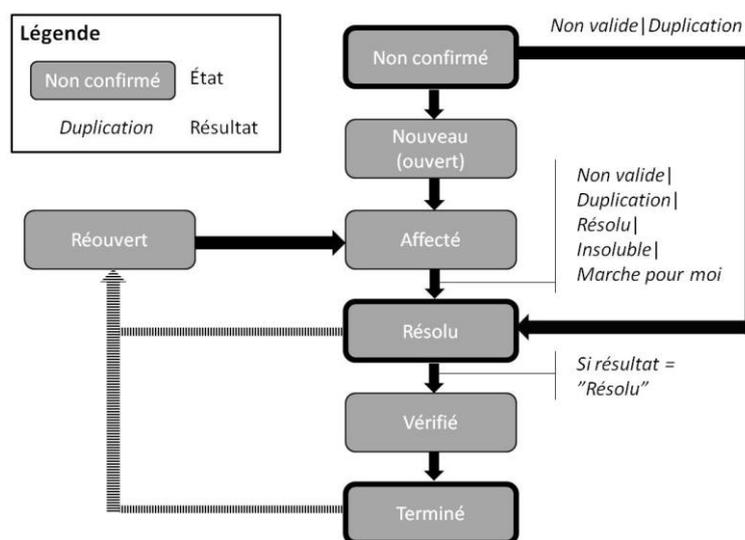


Fig. IV-1. Cycle de vie d'un rapport d'erreur (adapté de (Zeller 2009) [255])

Les erreurs sont inhérentes au développement de logiciels; la production de code sans défaut reste un défi fondamental en génie logiciel [22]. La prédiction et la détection automatique des erreurs dans le code est une question importante qui a reçu beaucoup d'attention (p.ex. [146], [53]) ; cette question ne fait pas partie de notre champ de recherche. Dans le travail présenté dans ce chapitre, le focus est sur la résolution des erreurs une fois qu'elles ont été rapportées et sur le problème de la prédiction de leur temps de résolution.

### IV.3 Bref état de l'art sur la prédiction du temps de résolution des erreurs logicielles

La prédiction du temps de résolution des erreurs logicielles (TRE) a été étudiée de manière intensive dans la dernière décennie [223]. La principale idée sous-jacente est d'exploiter les données historiques afin d'en déduire des modèles prédictifs. Ces approches utilisent plusieurs attributs d'un rapport d'erreur tels que sa gravité et le nombre associé de commentaires utilisateurs [175], ou le nombre de personnes concernées par le traitement de l'erreur [5]. Plusieurs techniques de fouille de données ont été expérimentées, tels que la régression logistique [175], les arbres de décision [85], l'analyse de régression univariée et multivariée [18], ou encore le modèle de Markov [256]. Néanmoins, aucune étude n'obtient de résultats définitifs, les auteurs concluent généralement qu'il y a une multitude de facteurs susceptibles d'influer sur la validité et la précision de l'approche de prédiction, et que la même approche appliquée sur des jeux de données différents peut donner des résultats contradictoires [26].

Lors de mon séjours sabbatique à l'université de Lund et la collaboration avec l'équipe de recherche SERG, nous sommes particulièrement intéressés à l'approche de prédiction du TRE qui exploite **la description textuelle** du rapport d'erreur logicielle. Nous avons alors analysé deux études :

- la première étude, datant de 2007, et qui met en œuvre une technique d'apprentissage supervisé [242];
- l'autre l'étude, celle de Raja (2013) qui venait d'être publiée en ligne, et qui met en œuvre une technique d'apprentissage non supervisé [182].

#### IV.3.1 Apprentissage supervisé avec la technique $\alpha$ -kNN

L'apprentissage supervisé cherche à inférer de manière empirique des règles à partir d'un ensemble de données d'apprentissage (ou d'entraînement<sup>25</sup>), constitué en général des cas déjà traités et validés. L'apprentissage est considéré comme « supervisé » car le contenu et la taille de l'ensemble d'apprentissage sont des paramètres définis par le chercheur, qu'il peut contrôler et qui impactent le résultat final (c.-à-d. la validité et la précision de la règle inférée).

L'étude de Weiss et al. (2007) utilise 567 rapports textuels de réclamation extraits des données du projet libre JBoss [242]. Ces rapports comprennent les catégories « erreur logicielle », « demande de nouvelle fonctionnalité », « tâche » et « sous-tâche ». Les auteurs appliquent leur approche de prédiction pour chaque nouveau rapport entrant. Pour le N<sup>ième</sup> rapport entrant, la prédiction exploite l'ensemble composé des N-1 rapports précédemment arrivés (cf. Fig. IV-2). L'approche du plus proche k-voisin (kNN) est utilisée pour trouver un ensemble d'apprentissage de k rapports d'erreurs les plus similaires, k variant de 1 à 3, 5, 7, 9 et  $\infty$  (la totalité des rapports précédents). La similarité textuelle entre les rapports est mesurée à l'aide du moteur de recherche libre Apache Lucene [154].

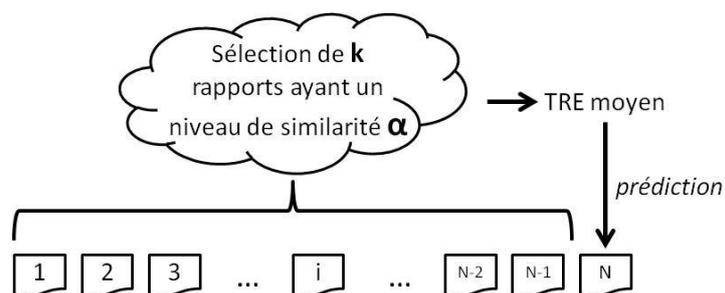


Fig. IV-2. Synopsis de la technique de prédiction par apprentissage supervisé développé dans Weiss et al. 2007

Le paramètre  $\alpha$  est utilisé pour définir un seuil pour le degré de similarité, allant de 0 (aucune similarité entre le texte du rapport et ceux de l'ensemble d'apprentissage) à 1 (les deux textes sont identiques). Pour des valeurs élevées de  $\alpha$ , il arrive souvent qu'aucun rapport similaire ne puisse être trouvé et l'approche renvoie "Inconnu". **L'applicabilité** est mesurée comme le pourcentage de réponses différentes de "Inconnu". La **précision** est calculée comme le pourcentage de prédictions du temps de résolution présentant une erreur relative inférieure ou égale à 25% et 50%, respectivement. Pour kNN (sans seuil), la précision est médiocre car aucune similarité n'est imposée sur les rapports

<sup>25</sup> En anglais "training set"

formant l'ensemble d'apprentissage. Seulement 30% des prévisions se situent alors dans  $\pm 50\%$  de l'intervalle du temps de résolution réelle, et ce quelque soit le choix de  $k$ .

Le réglage du seuil  $\alpha$  introduit en fait un **compromis** entre l'**applicabilité** et la **précision**. Pour  $k = \infty$  (recherche dans tous les défauts précédents) et  $\alpha = 0,9$  (limité à des rapports d'erreurs très similaires), la précision est excellente : la prédiction moyenne est décalée de seulement 7 heures par rapport au temps réel de résolution, et presque une prédiction sur deux se trouve à  $\pm 50\%$  du temps réel de résolution. Toutefois, dans cette configuration particulière, l'applicabilité est très limitée car les prévisions sont faites pour seulement 13% de l'ensemble de données : pour 87% des données, l'approche ne donne aucun résultat, car il n'y a pas de rapports d'erreurs avec 0,9 degré de similarité à comparer avec et à utiliser pour obtenir les estimations (c.-à-d. l'ensemble d'apprentissage est vide). Pour des valeurs inférieures de  $\alpha$  et pour  $k = \infty$ , l'applicabilité augmente (car l'ensemble d'apprentissage n'est pas vide) mais la précision diminue. Ainsi, pour  $\alpha = 0,5$ , seulement 30% des valeurs prédites sont dans la marge d'erreur de 25%, et pour  $\alpha = 0$  (c.-à-d. aucune similarité n'est exigée pour constituer l'ensemble d'apprentissage), il tombe à 20%.

En fait, avec  $k = \infty$  et  $\alpha = 0$ , l'approche correspond à une **approche d'estimation naïve**, à savoir, le temps moyen de résolution de tous les rapports précédents est pris comme fonction de prédiction. Enfin, les résultats varient beaucoup selon le sous-projet JBoss analysé, ainsi que la catégorie du rapport. Les auteurs expliquent leurs résultats par la diversité des descriptions textuelles dans les rapports et suggèrent que des champs supplémentaires, tels que l'information de version ou les pièces jointes pourraient être exploitées pour améliorer la précision de la prédiction.

### IV.3.2 Apprentissage non supervisé avec la technique du partitionnement

L'apprentissage non supervisé consiste à inférer des règles à partir d'un jeu de données sans qu'il y ait un ensemble d'apprentissage pour construire la règle. La principale technique d'apprentissage non supervisé est celle du partitionnement (appelé aussi *regroupement*), qui consiste à identifier des groupes homogènes (ou des classes, des catégories, des motifs) dans un ensemble de données [10]. La recherche de groupes homogènes nécessite la définition d'une **mesure de la distance** entre les points de données. L'algorithme de partitionnement va chercher ensuite à construire des partitions où la distance, entre les points et un certain centre, est homogène. Plusieurs algorithmes de partitionnement existent, et le résultat n'est pas systématiquement pertinent ou correcte. En effet, il est généralement possible de partitionner un ensemble de données de plusieurs façons ([114], p.658).

L'étude de Raja (2013) utilise le partitionnement textuel<sup>26</sup> des rapports d'erreurs pour analyser le temps de résolution des erreurs et proposer cette technique pour la prédiction du TRE [182]. Elle a montré que les différents groupes de rapports d'erreurs – obtenus par partitionnement automatique – ont des temps de résolutions moyen **significativement différents**. Sur la base de cette observation, Raja affirme que « *la classification basée sur le texte des rapports d'erreurs peut être un indicateur précoce et utile du temps de résolution de l'erreur* » (p.135), ce qui implique que son approche pourrait être appliquée pour prédire la TRE des erreurs entrantes. Par exemple, le TRE moyen de la partition à laquelle appartient un rapport d'erreur nouvellement déclaré pourrait être utilisé pour prédire le temps nécessaire pour résoudre cette erreur. Il convient de noter que Raja ne fournit pas d'instructions sur la façon de concevoir dans la pratique une telle méthode de prédiction. Elle n'a pas non plus vérifié l'applicabilité et la qualité d'une telle méthode de prédiction.

---

<sup>26</sup> En anglais "textual clustering"

## IV.4 Problématique

Comme indiqué au début de ce chapitre, ce travail de recherche s'est construit autour de deux questions (cf. IV.1) : (i) vérifier les résultats l'étude initiale de Raja (2013) en réalisant une réplcation, et (ii) conduire une expérimentation empirique pour tester la validité de la proposition émise par l'auteur concernant l'usage de la technique de partitionnement textuel pour la prédiction du TRE. Cette section présente l'étude initiale de Raja (2013) et les interrogations que soulèvent les deux questions de recherche.

### IV.4.1 Brève présentation de l'étude initiale

Raja (2013) étudie cinq jeux de données contenant des rapports d'erreurs issues de logiciels libres, avec un total d'environ 14 000 rapports d'erreurs à l'état « terminé ». Les données sont d'abord prétraités et préparées pour la phase de partitionnement : élimination des mots d'arrêts, recherche des radicaux, pondération et indexation sémantique latente (LSI). Elle utilise ensuite l'outil **SAS Text Miner** pour partitionner les rapports d'erreurs. Elle applique la classification automatique basée sur la minimisation de l'entropie avec intervention humaine (grâce à une fonction interactive de SAS Text Miner) pour réduire et optimiser les partitions obtenues. Ces interventions manuelles ne sont pas explicitement détaillées. Le résultat final est trois à cinq partitions pour chaque jeu de données, avec pour chaque partition, une extraction des termes les plus descriptifs (c.-à-d. les plus fréquents dans la partition).

Test statistique	Objectif du test	Résultats
Test de <i>Kolmogorov-Smirnov</i>	La normalité de la distribution du TRE dans les partitions	Partiellement positif – pour les données de chaque projet, " <i>il y avait une légère violation de l'hypothèse de normalité</i> " pour certaines partitions (p.129)
Test de <i>Levene</i>	Vérifier l'homogénéité de la variance du TRE entre les partitions	Négatif – variance inégale pour les données de toutes les partitions (p.129)
Test ANOVA	Tester la variance des moyennes du TRE entre les partitions	Positif – pour chaque projet, la moyenne du TRE d'au moins une partition est différente de toutes les autres (p.130)
Test de <i>Brown-Forsythe</i>	Similaire à ANOVA en cas d'hétérogénéité de la variance	Positif – pour tous les projets, les résultats étaient consistants avec ceux du test ANOVA (p.130)
Test post-hoc de <i>Games-Howell</i>	Identifier les différences de moyennes du TRE entre les partitions	Partiellement positif – pour certaines partitions, les différences entre les moyennes du TRE ne sont pas significatives (p.130)

Tableau IV-1. Les analyses statistiques appliquées dans l'étude initiale Raja (2013)

Raja effectue ensuite un traitement statistique en trois étapes (cf. Tableau IV-1) :

– La première étape est un ensemble de statistiques descriptives sur les partitions obtenues ; elle remarque que "*le temps moyen pris pour résoudre une erreur varie considérablement entre les projets et entre les partitions d'un même projet*" ([182], p 127 ).

– Dans la deuxième étape, elle applique le test ANOVA pour une analyse unidirectionnelle de la variance et vérifier la signifiante de la variation des moyennes du TRE entre les partitions. Au-delà de l'indépendance des observations dans l'échantillon, ANOVA repose sur des hypothèses de normalité de l'échantillon et de l'homogénéité de la variance. Raja effectue donc au préalable le test statistique de *Kolmogorov-Smirnov* de normalité et celui de *Levene* d'homogénéité. Elle interprète les résultats comme étant globalement positifs, à savoir, hypothèse de normalité est globalement établie même si elle est violée dans certaines partitions. Bien que le test de *Levene* révèle une variance inégale, l'auteur fait valoir que le test ANOVA est applicable, parce qu'il peut être considéré comme robuste en cas de variances inégales. Néanmoins, elle applique le test de *Brown-Forsythe* qui est une variante du test standard ANOVA et qui est plus robuste vis-à-vis de les contraintes de normalité et d'homogénéité.

– Enfin, la dernière étape est le test post-hoc pour l'analyse détaillée de la variance inter-partitions, en utilisant le test de *Games-Howell*. Les résultats montrent que seules certaines combinaisons de pôles présentent une variation statistiquement significative dans les moyens de DRT.

Comme les tests post-hoc entre les partitions ont donné des résultats contradictoires, l'auteur cherche une explication en analysant deux facteurs possibles. D'abord, elle analyse les termes clés figurant dans chaque partition et les relations potentielles avec la nature du composant logiciel sur lequel l'erreur a été constatée. Par exemple, si les termes clés d'une partition sont liés à l'interface de développement, tandis que les termes clés d'une autre (dans le même projet) sont liés au code du serveur, les différences de temps de résolution peuvent s'expliquer par une différence de complexité entre les codes sources. Un autre facteur explicatif pourrait être le nombre de téléchargements de logiciels, et donc, la popularité du logiciel et de son impact potentiel sur l'équipe de maintenance en termes de priorité donnée à la résolution de l'erreur.

#### IV.4.2 Estimation de la répliquabilité de l'étude initiale

Dans le chapitre II concernant les méthodes, j'ai évoqué brièvement la question de la répliquabilité d'une étude empirique et son importance dans la démarche d'investigation scientifique (cf. §II.66). La répliquabilité d'études empiriques aurait le potentiel de répondre au problème d'inconstance des résultats obtenus : l'effet X constaté dans une certaine situation ne l'est pas dans d'autres situations, et on peut y avoir autant de preuves ou de facteurs explicatifs pour l'effet X comme contre cet effet [155]. Il y a en effet un grand nombre de points de variation à considérer lors de la répliquabilité d'une étude [213], et plusieurs stratégies de répliquabilité sont possibles [31]. Une **répliquabilité interne** est assurée par la même équipe de l'étude initiale et qui répète sa propre expérience, alors qu'une **répliquabilité externe** est menée par des chercheurs indépendants. En outre, une **répliquabilité exacte** (ou dépendante) doit suivre d'aussi près que possible les procédures et procédés de l'expérience initiale, alors qu'une **répliquabilité conceptuelle** (ou indépendante) évalue la même question de recherche en utilisant des procédures et procédés expérimentaux différents et en altérant l'ensemble des variables indépendantes [213], [121].

Pour aider à la répliquabilité de recherches empiriques en génie logiciel, González-Barahona et Robles (2012) présentent une grille d'analyse pour évaluer la reproductibilité d'une étude [89]. Cette grille comporte huit caractéristiques à extraire de l'étude empirique et à analyser. Nous avons évalué la reproductibilité de l'étude originale de Raja (2013) selon cette grille. Cette évaluation est basée sur des informations extraites directement de l'article original, ainsi que sur des échanges de courriels avec l'auteur.

	Identifi- cation	Descrip- tion	Disponi- bilité	Persis- tance	Flexi- bilité	Evaluation	Commentaires
Source de données	Oui	Détaillée	Publique	Probable- ment	–	Utilisable	La source des données est le référentiel public <i>Sourceforge</i> ; néanmoins, les rapports d'erreurs ne sont pas directement téléchargeables
Méthode d'interrogation	Partielle	Partielle	Non	Non	–	Difficile	Les paramètres de la requête ne sont pas fournis
Jeu de donnée brut	Oui	Détaillée	Non	Non	–	Difficile	La collection des rapports d'erreurs n'est pas mise à disposition par l'auteur
Méthode d'extraction	Oui	Partielle	Non	Non	–	Difficile	Des étapes complexes de prétraitement textuel sont décrites brièvement
Paramètres	Partielle	Non	Non	Non	–	Difficile	Les paramètres de prétraitement ne sont pas décrits
Jeu de données traité	Partielle	Partielle	Non	Non	–	Difficile	Les rapports d'erreurs prétraités ne sont pas disponibles
Méthode d'analyse	Oui	Partielle	Non	Non	–	Utilisable	Utilise un logiciel propriétaire avec un paramétrage spécifique
Jeu de données résultat	Oui	Détaillée	Oui	Oui	Non	Utilisable	Les résultats sont présentés en détail dans l'article

Tableau IV-2. Evaluation de la répliquabilité de l'étude Raja (2013)

Comme on peut le voir (cf. Tableau IV-2), l'étude Raja (2013) est **difficile à reproduire**. Bien que la source des données soit facilement identifiable (le référentiel de logiciels libres *sourceforge.com*), l'ensemble des données brutes, c.-à-d. l'ensemble des rapports d'erreurs traités, ne sont pas disponibles. En outre, nous avons contacté l'auteure et demandé les jeux de données brutes, elle a cependant été incapable de fournir un ensemble identique à ce qui a été utilisé dans son expérience initiale. Enfin, l'ensemble des données traitées (c.-à-d. l'ensemble des rapports d'erreurs après le nettoyage manuel) et les méthodes de récupération et d'extraction correspondantes sont difficiles à reproduire, car elles sont mal documentées dans l'étude originale et n'ont pas été mises à la disposition du public. En ce qui concerne la méthode d'analyse des données, elle est également difficile à reproduire car elle repose sur un outil logiciel statistique propriétaire (SAS Text Miner) et utilise des paramètres spécifiques non documentés (p.ex., partitionnement interactif).

#### IV.4.3 Techniques de simulation pour le test d'hypothèse

Comment mettre en œuvre de manière opérationnelle la proposition faite dans Raja (2013), à savoir, que les variations du TRE entre les partitions des rapports d'erreurs peuvent être utilisées comme un indicateur précoce du temps de résolution pour une erreur nouvellement rapportée ? Pour définir un contexte opérationnel, nous adoptons une approche de simulation [161]. Une simulation est une imitation du comportement de certains processus [233], ce qui dans notre cas, correspond à **l'afflux de rapports d'erreurs**. Le défi réside dans la conception d'un scénario opérationnel *plausible* et *réalisable* qui offre un terrain pour la collecte de données utiles.

Une approche possible, par exemple, serait de réaliser un partitionnement automatique des rapports d'erreurs chaque fois qu'un nouveau rapport arrive. Les données partitionnées contiendraient alors les rapports d'erreurs déjà terminés et le rapport ouvert. Une prédiction possible du TRE serait alors la moyenne des TRE des rapports d'erreurs terminés dans la partition à laquelle le rapport d'erreur entrant appartient. Néanmoins, un tel scénario événementiel est difficilement réalisable car le partitionnement nécessite des calculs intensifs et ne peut guère être refait à chaque fois qu'un rapport d'erreur nouveau arrive. Ce qui requiert donc une approche différente d'opérationnaliser le partitionnement et de simuler l'afflux de rapports d'erreurs. La solution choisie sera présentée dans la section 6.

### IV.5 Partie 1 : réplique de l'étude Raja (2013)

À la lumière de l'évaluation faite dans la section précédente (cf. §IV.4.2), une réplique exacte de l'étude initiale (Raja 2013) exigerait l'utilisation des mêmes outils d'extraction de données et l'application de la même technique de partitionnement, ainsi qu'une étroite collaboration avec l'auteure afin d'utiliser de façon identique les divers paramètres de traitement des données. Cependant, nous ne souhaitons pas réaliser une réplique exacte de l'étude initiale. Une réplique exacte réussie confirme les résultats expérimentaux et théoriques de l'expérience initiale : une réplique exacte consolide **la validité interne** et la conclusion de l'expérience initiale ([201], p.5). Par contre, une réplique indépendante, non-exacte, c.-à-d. avec la variation d'un ou plusieurs aspects majeurs de l'expérience, est beaucoup plus utile pour démontrer qu'un effet ou une technique est robuste face aux changements de sujets, de paramètres ou d'outils [126], [121]. En cas de succès, une réplique conceptuelle non-exacte étend la validité potentielle des connaissances théoriques acquises à partir de l'expérience initiale et contribue à sa validité externe [201], [88].

Nous avons donc choisi de mener une réplique **conceptuelle, indépendante et non-exacte**, de l'étude initiale de Raja (2013) avec **trois points** principaux de **variation** (cf. Tableau IV-3). Tout d'abord, alors que dans l'expérience originale cinq jeux de données issus de logiciels libres ont été utilisés, nous utilisons deux jeux de données provenant de logiciels libres et une série de données issue d'un projet mené dans une entreprise privée. Deuxièmement, nous avons utilisé un outil logiciel différent pour le partitionnement textuel, à savoir, RapidMiner au lieu de SAS Text Miner, et une technique de partitionnement différente (k-moyenne au lieu de *entropie minimisation*). Troisièmement, nous appliquons le partitionnement automatisé. Ceci est différent de l'expérience initiale qui nécessitait l'intervention humaine et des connaissances experts pour optimiser les partitions obtenues.

	L'expérimentation initiale	La réplication conceptuelle réalisée
Projets étudiées (#erreurs)	<u>Logiciels libres</u> : FileZilla (956), jEdit (1682), phpMyAdmin (1521), Pidgin (2689), Slash (3580)	<u>Logiciels libres</u> : Android (4684), Eclipse(4158) <u>Logiciel propriétaire</u> : Company A (6790)
Outil utilisé	SAS Text Miner	RapidMiner
Prétraitement manuel	Suppression des rapports en double, examen manuel des rapports résolus en moins d'une heure	Suppression des rapports en double
Prétraitement automatique	Mots d'arrêt, recherche des radicaux, pondération (poids entropique), Indexation sémantique latente (LSI <sup>27</sup> )	Mots d'arrêt, recherché des radicaux, pondération (TF-IDF <sup>28</sup> )
Algorithme de partitionnement	Entropy minimization	Partitionnement en K-moyenne
Optimisation manuelle des partitions	Plusieurs étapes de traitement intensif	Aucune
# de partitions	3-5	4

Tableau IV-3. Comparaison entre l'expérience initiale dans Raja (2013) et la réplication que nous avons réalisée

#### IV.5.1 Description des jeux de données

Le tableau IV-4 présente les données utilisées dans notre étude. Les rapports d'erreurs issus du projet Android ont été recueillis en 2011 et mis à disposition par Shihab et al. sous forme d'un fichier au format XML contenant 20,176 rapports. Ce fichier a été distribué dans le cadre du défi de fouille de données réalisé à la conférence sur la fouille de référentiels logiciels [212]. Le projet Eclipse se compose de plusieurs sous-projets avec leurs propres rapports d'erreurs respectifs. Les rapports d'erreurs du projet Eclipse, gérés dans des bases de données *Bugzilla*, ont été utilisés pour la prédiction du TRE dans plusieurs études antérieures (p.ex. [175], [85], [135]), ce qui a motivé leurs inclusions dans cette étude. Nous avons extrait 4.158 rapports d'erreurs avec le statut « terminé » du sous-projet "Plate-forme Eclipse" en Mars 2012. Les données brutes de ces deux projets sont disponibles sur un site Web accessible au public<sup>29</sup>.

	Android	Eclipse	Company A
Le domaine du produit	Télécom, système d'exploitation	Développent d'applications	Automatisation de processus, système de contrôle industriel
Caractéristiques du projet	Logiciel libre, géré par Google	Logiciel libre, géré par des organisations à but non lucratif	Proprietary, iterative stage-gate process, safety-critical
Distribution initiale	2008	2004	2003
#Utilisateurs / #Téléchargement	~500 million d'appareils activés	~5 million de téléchargement	Base limitée d'utilisateurs
# de rapports d'erreurs	20,176	9,0691	7,697
# rapports d'erreurs nettoyés	4,684	4,158	6,790
# moyen de caractères / rapport d'erreur	821	3,699	1,208
Principaux auteurs des rapports d'erreurs	Communauté logiciel libre	Communauté logiciel libre	Développeurs internes / testeurs

Tableau IV-4. Aperçu des jeux de données utilisées dans cette étude

<sup>27</sup> LSI : *Latent Semantic Indexing*

<sup>28</sup> TF-IDF (*Term Frequency-Inverse Document Frequency*) est une technique standard de pondération des termes pour la recherche de données textuelle. Elle est basée sur la longueur du document et la fréquence des termes, dans un document ainsi que dans l'ensemble des documents de la collection dans laquelle s'effectue la recherche [219].

<sup>29</sup> <http://serg.cs.lth.se/research/experiment-packages/clustering-defect-reports/>

Le troisième jeu de données est issu d'un projet logiciel réalisé dans entreprise industrielle en Suède (dénommé Company A pour des raisons de confidentialité). Company A est une multinationale active dans le secteur de l'énergie et de l'automatisation. Le contexte de développement est celui des logiciels embarqués et des systèmes de contrôle industriel avec des contraintes fortes de sécurité. Nous avons extrait 7,697 rapports d'erreur correspondant à un projet de développement unique qui, à son tour, est divisé en plusieurs sous-projets de développement.

#### IV.5.2 Le protocole expérimental de la réplication

La Fig. IV-3 résume les principales étapes de notre étude. La partie consacrée à la simulation est présentée dans la section 6 de ce chapitre. Pour chaque rapport d'erreur, nous avons extrait et fusionné le titre (un résumé d'une ligne du rapport d'erreur) avec sa description. Comme dans l'expérience initiale Raja (2013), nous avons nettoyé manuellement les rapports d'erreur avant d'effectuer le partitionnement automatique. Nous avons éliminé les rapports dupliqués, et nous avons exclu tous les rapports d'erreur qui étaient encore ouverts au moment de la collecte des données.

Alors que dans l'expérience initiale tous les rapports d'erreur fermés avec des TRE de moins d'une heure ont été analysés manuellement pour vérifier s'ils sont valides, nous n'avons pas effectué une telle analyse. Le nettoyage des données que nous avons effectué se limite à ce qui est exécutable automatiquement (par exemple, la suppression des doublons), et sinon, les données non filtrées passent directement à l'étape de partitionnement automatique.

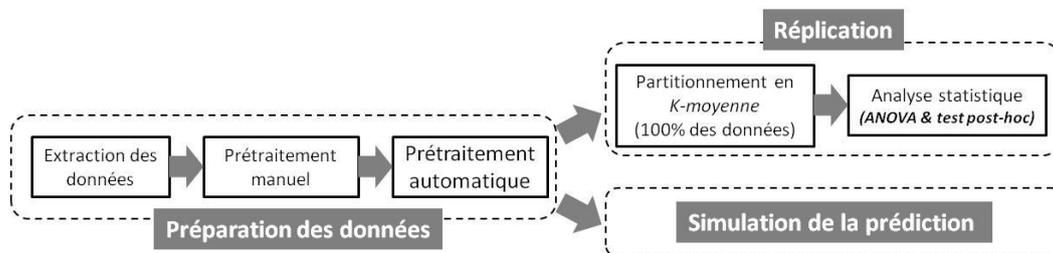


Fig. IV-3. Synopsis de l'étude expérimentale.

Pour partitionner les rapports d'erreur en fonction de leur contenu textuel, nous avons utilisé *RapidMiner*, un outil logiciel libre d'exploration de données [108]. Un paramètre important est le nombre de partitions. Ce nombre peut être fixé par l'utilisateur ou défini automatiquement par l'algorithme. La détermination automatique de ce **nombre K** est un problème fondamental pour la validité du partitionnement, et il est reconnu comme difficile [63].

Dans l'expérience initiale Raja (2013), l'auteur a déterminé manuellement ce nombre K par partitionnement itératif à l'aide de la technique de *minimisation de l'entropie*, un processus qu'elle décrit comme étant intensif et consommateur de ressources. Elle indique que le nombre de partitions a été itérativement optimisé (c.-à-d. jusqu'à ce qu'un critère d'arrêt ait été atteint) en utilisant une fonction interactive du logiciel *SAS Text Miner*. Elle a également procédé à un filtrage manuel des termes, pour assurer que seuls les termes pertinents soient inclus dans les partitions finales, en supprimant par exemple des mots et des termes mal orthographiés ou liés au code source. Ce processus a abouti à la réduction du nombre de partitions de 40, le nombre maximal de partitions dans *SAS Text Miner*, à entre trois à cinq partitions pour les cinq jeux de données.

En ligne avec notre objectif d'automatiser totalement l'étape de partitionnement, nous ne voulions pas accomplir un réglage manuel des partitions produites par l'outil *RapidMiner*. Au lieu de cela, nous avons choisi de **fixer ce paramètre à 4**. Ce choix correspond à la moyenne des valeurs rapportées comme optimales dans l'étude initiale ; il y a aussi une justification théorique pour ce choix que je ne détaillerais pas ici (cf. publi#02, p.1452).

#### IV.5.3 Les résultats de la réplication

Dans cette section, nous présentons un extrait des résultats de la réplication et les comparons avec ceux de l'expérience initiale. La Fig. IV-4 montre comment le temps de résolution est distribué pour l'ensemble des rapports d'erreur pour chaque jeu de données. Les données sont clairement très fortement aplaties à droite, et évidemment, elles ne sont pas normalement distribuées.

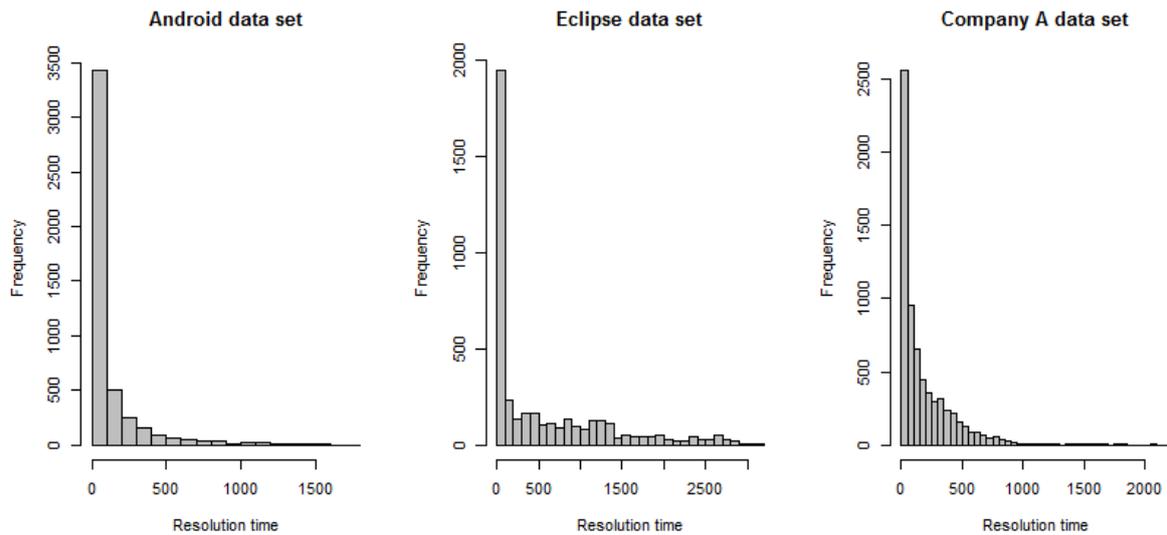


Fig. IV-4. Distribution des TRE pour les trois jeux de données (temps de résolution exprimé en jours)

La Fig. IV-5 présente graphiquement les propriétés des quatre partitions pour chaque jeu de données. La moyenne du TRE varie entre les partitions pour chaque jeu de données, avec de grands écarts.

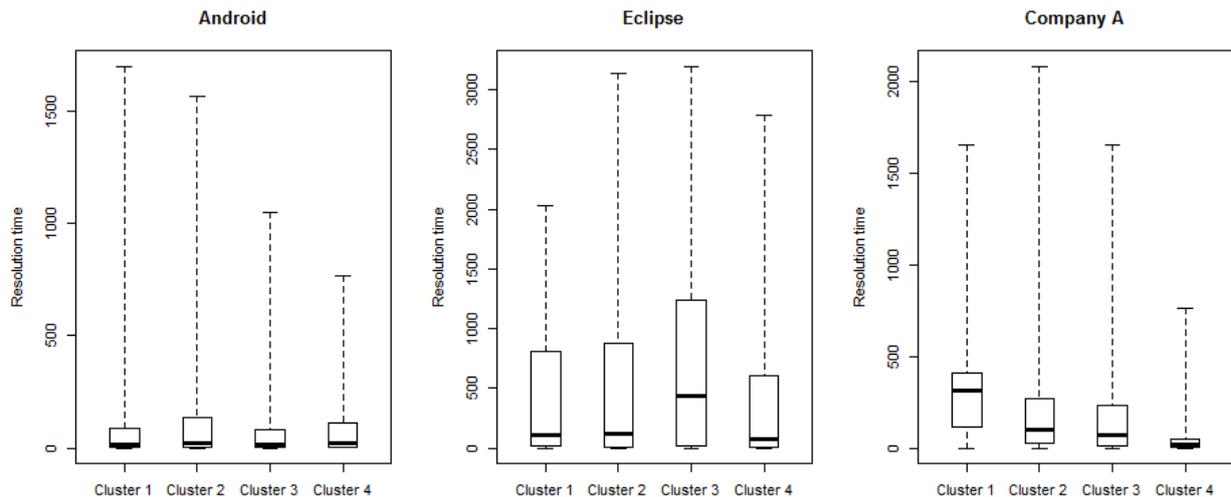


Fig. IV-5. Les moyennes des TRE dans les quatre partitions pour chaque jeu de données

L'étude initiale Raja (2013) a suggéré que les différences dans les moyennes du temps de résolution entre les partitions peuvent être utilisées pour la prédiction du temps de résolution (p. 127). Il faut au préalable établir que ces différences (dans les moyennes du temps de résolution entre les partitions) sont statistiquement significatives, c.-à-d. qu'elles ne sont pas dues au hasard. Le test ANOVA permet de le vérifier ; ce test requière trois conditions (cf. §IV.3.2), dont la condition de normalité. L'application du test de Kolmogorov-Smirnov (non rapporté ici) montre que cette condition n'est pas vérifiée pour les trois jeux de données. Néanmoins, c'était le cas aussi dans l'étude initiale où l'auteure indiquait que certaines partitions qu'elle a obtenu ne satisfont pas le test de Kolmogorov-Smirnov (p.129). Elle a estimé que la grande taille des ensembles de données dans les partitions permettait de **s'affranchir de cette contrainte**. Ceci est effectivement confirmé dans la littérature, le test ANOVA est considéré robuste par rapport à un décalage de la présomption de normalité ; même des données fortement aplaties ne poseraient pas problème, du moment que les ensembles comparées présentent tous un niveau similaire d'aplatissement [205].

Concernant la condition d'homogénéité de la variance, nous avons obtenu les mêmes résultats (non montrés ici) que l'étude initiale à l'aide du test de *Levene*. Ce test a confirmé que la variance du TRE n'est pas homogène. Ceci implique d'utiliser, en plus du test ANOVA dont les résultats peuvent être faussés par la violation de la troisième con-

trainte, le test de *Brown-Forsythe* qui est plus robuste vis-à-vis de cette condition. La Tableau IV-5 montre les résultats obtenus avec ces deux tests, et qui sont identiques à ceux de l'étude initiale.

	Test ANOVA unidirectionnel		Test de <i>Brown-Forsythe</i>	
Android	F=13.47	valeur-p= 0.0000 ***	F=14.03	valeur-p= 0.0000 ***
Eclipse	F=21.44	valeur-p= 0.0000 ***	F=18.78	valeur-p= 0.0000 ***
CompA	F=92.99	valeur-p= 0.0000 ***	F=48.5	valeur-p= 0.0000 ***

Tableau IV-5. Analyse de la variance pour les quatre partitions dans les trois jeux de données

Les tests ANOVA et *Brown-Forsythe* indiquent que la variation dans les moyennes des temps de résolution pour les quatre partitions obtenues est significative (valeur-p < 0.05). Néanmoins, ces test de la variance n'indiquent pas si ce résultat est valable pour *toutes* les paires de partitions, ou uniquement pour certaines d'entre elles. Le test *post-hoc* permet d'identifier les paires de partitions pour lesquelles cette variation est significatives et celles pour lesquelles elle ne l'est pas. Les résultats sont indiqués dans le Tableau IV-6.

	Partition 2	Partition 3	Partition 4
<b>Android</b>			
Partition 1	0.0004 ***	<b>0.0472 *</b>	<b>0.9127</b>
Partition 2		0.0000 ***	<b>0.8930</b>
Partition 3			<b>0.3840</b>
<b>Eclipse</b>			
Partition 1	<b>0.0250 *</b>	0.0000 ***	<b>0.9308</b>
Partition 2		0.0000 ***	0.0000 ***
Partition 3			0.0000 ***
<b>Company A</b>			
Partition 1	0.0000 ***	0.0000 ***	0.0000 ***
Partition 2		0.0000 ***	0.0000 ***
Partition 3			0.0000 ***

Tableau IV-6. Les valeur-p pour le test post-hoc *Games-Howell* (les valeurs en gras ne sont pas significatives à la limite  $\alpha=0.05$ )

Globalement, les résultats que nous avons obtenus sont tout à fait conformes avec ceux de l'étude initiale. De plus, certains résultats que nous avons obtenus sont encore plus positifs et confirmatoires que ceux de l'étude initiale. En effet, dans l'étude initiale, pour chacun des cinq jeux de données, le test de *Games-Howell* était négatif pour au moins une paire de partitions. Dans notre étude, ce résultat est positif pour toutes les paires de partitions du jeu de données de la Company A (cf. Tableau IV-6, 3<sup>e</sup> ligne).

#### IV.5.4 Conclusion sur la partie 1

Les résultats obtenus avec la réplication confirment sans aucune ambiguïté ceux de l'étude initiale. Ce qui permet de généraliser le résultat initial car il est visiblement indépendant du jeu de données, des prétraitements spécifiques et de la technique de partitionnement des données. C'est le principe de la réplication conceptuelle qui permet d'énoncer une telle généralisation. Ce succès contribue à la **validité externe** des résultats publiés dans l'étude initiale, et confirme que le partitionnement automatique des rapports d'erreurs conduit effectivement à des partitions avec des temps moyens de résolution significativement différents.

Néanmoins, ce résultat en soi est peu intéressant ; ce qui lui confère un certain intérêt, c'est l'éventualité qu'il laisse entrevoir d'une nouvelle technique de prédiction du temps de résolution d'une erreur. Nous nous sommes attaché dans cette étude à explorer cette éventualité. C'est l'objet de la seconde partie de cette étude.

## IV.6 Partie 2 : test opérationnel de l'hypothèse émise par Raja (2013)

Cette section concerne la question de recherche RQ2. L'objectif est d'explorer la possibilité évoquée dans l'étude initiale d'utiliser le partitionnement pour prédire le TRE. La technique de simulation est mise en œuvre pour évaluer empiriquement une telle possibilité.

### IV.6.1 Le protocole expérimental du scénario de simulation

Comme discuté au début de ce chapitre (cf. §IV.4.3), la question clé dans un scénario de simulation est celle de l'afflux de rapports d'erreur. Il est impératif de prendre en compte les ressources nécessaires pour réaliser le partitionnement d'un grand volume de rapports textuel, et qu'il est donc opérationnellement impossible de réaliser le partitionnement à chaque nouvelle déclaration d'une erreur. Nous avons plutôt opté pour un scénario qui simule l'application de la méthode de prédiction à **divers points dans le temps** (cf. Fig. IV-6).

Le scénario d'évaluation que nous mettons en œuvre s'inspire de la technique de simulation appelé « k-queue » dans laquelle les données après un point prédéfini sont utilisés en tant qu'ensemble de test [233]. Dans notre contexte, l'ensemble de test est le sous-ensemble des rapports d'erreur pour lequel TRE sera prédit. Comme notre simulation n'est pas liée à un contexte réel de développement de logiciels, nous ne disposons pas d'un critère pratique pour décider quels points dans le temps seraient appropriés pour définir les ensembles de test. Un tel critère lié au temps pourrait être défini, par exemple, par un jour spécifique de la semaine, ou une heure de la journée. Il pourrait être judicieux, par exemple, d'estimer le temps de résolution pour les nouveaux rapports de d'erreur une fois par semaine. Un autre critère pourrait être lié à la quantité de nouveaux rapports d'erreur, par exemple, chaque fois que 20 nouveaux rapports se sont accumulés.

Dans le scénario de simulation, notre objectif est d'illustrer la mécanique de l'approche de prédiction. Nous avons donc choisi une manière quelque peu arbitraire de diviser les données disponibles afin que nous puissions simuler, de manière répétitive et selon l'axe croissant du temps, l'application de la technique de prédiction à des fragments séquentiels de nouveaux rapports d'erreur. Nous avons simplement divisé chaque jeu de données – trié par le temps de soumission – en dix sous-ensembles, le premier correspondant à une tranche de 10% de l'ensemble de données, le deuxième à 20%, le 3<sup>ème</sup> à 30%, etc. (cf. Fig. IV-6). Ces sous-ensembles représentent différents points dans le temps où les rapports d'erreur auraient été regroupés pour prédire leurs temps de résolution.

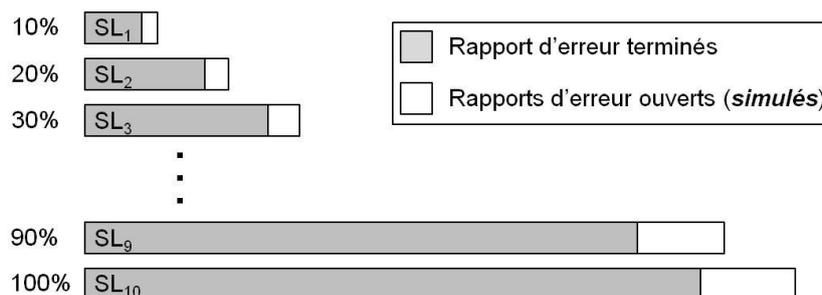


Fig. IV-6. Des tranches de données de taille croissante pour le scénario de simulation

A chaque point dans le temps où une prédiction doit être faite, toute la tranche de données est partitionnée. Les TRE des rapports d'erreur nouvellement arrivés reçoivent uniformément la valeur moyenne du TRE de la partition à laquelle ils appartiennent comme prédiction. Ceci est suivi par une étape d'évaluation dans laquelle on compare le TRE prédit avec le TRE réel pour calculer la puissance prédictive de l'approche. Rappelons que dans notre simulation, nous faisons comme si les rapports d'erreur étaient nouvellement arrivés (c.-à-d. ouverts) ; en réalité, ils sont terminés et nous connaissons leurs temps réels de résolution. Le protocole expérimental de test est décrit dans la Fig. IV-7.

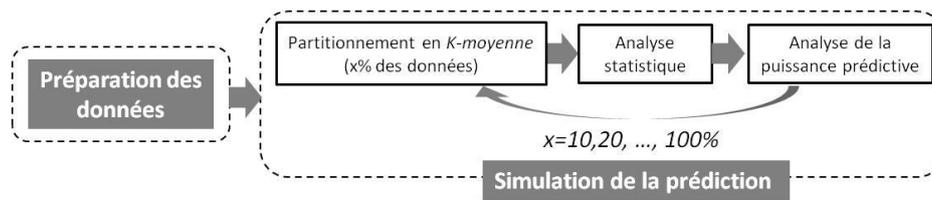


Fig. IV-7. Synopsis de la partie 2 test de l'étude (cf. section §IV.5.2 pour la partie préparation des données)

Le nombre *simulé de rapports d'erreurs ouverts* (désigné par le paramètre *SimREO*) est défini par défaut comme étant égale à 10% d'une tranche de données (cf. les zones blanches dans la Fig. IV-6). Par exemple, pour le jeu de données Android (cf. Tableau IV-4), la tranche 40% contient 1874 rapports d'erreur ordonnés en fonction de leur jour et heure d'arrivée. Le sous-ensemble simulé de rapports d'erreurs ouverts contient alors 187 rapports d'erreurs.

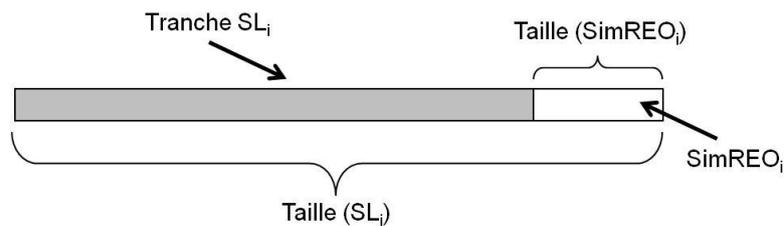


Fig. IV-8. Définition de la portion de rapports d'erreur ouverts utilisés pour la simulation dans une tranche de données  $SL_i$

En réalité, la taille relative des rapports simulés ouverts dans une tranche doit aussi être considérée comme un paramètre dans le scénario de simulation. Nous ferons référence à ce paramètre en tant que la *Taille du Facteur de Simulation* ou TFS. Il est possible d'envisager des valeurs différentes pour ce paramètre. Pour l'ensemble de données Android et la tranche de 40% par exemple et  $TFS = 0,3$ , le nombre simulé de rapports d'erreur ouverts (SimREO) est égal à 562 rapports d'erreurs ; pour  $TFS = 0,5$ , il sera égal à 937 rapports d'erreur. Nous considérons 0,5 comme la plus grande valeur possible pour le paramètre TFS. Ceci est résumé dans les définitions suivantes :

Taille ( $SL_i$ ) = Taille de la  $i^{\text{ème}}$  tranche  $SL_i$ , ( $i = 1, 2, 3, \dots, 10$ )

TFS = *Taille du Facteur de Simulation*, [un paramètre pour définir la taille de la portion des rapports d'erreur utilisés pour la simulation,  $0,1 \leq TFS \leq 0,5$ ]

Taille (SimREO $_i$ ) = Taille ( $SL_i$ ) \* TFS, [taille de la portion SimREO des *Rapports d'Erreur Ouverts* issus de la tranche  $SL_i$  et utilisé pour la simulation]

Pour exécuter le scénario de simulation, à chaque point dans le temps et pour chaque tranche de données, nous définissons **un nombre fixe** de rapports d'erreurs en tant qu'ensemble de test (cf. Fig. IV-9). Ce choix découle du besoin d'avoir des résultats comparables lorsqu'on considère les tranches successives de données utilisées à chaque itération du processus de simulation. Le nombre de rapports d'erreur prévus dans l'ensemble de test doit être **constant** pour toutes les tranches de données, quelque soit leurs tailles. En conséquence, nous avons fixé ce nombre en considérant la plus petite tranche de données (c.-à-d. la tranche 10%). Ainsi, à  $TFS = 0,1, 0,3$  et  $0,5$ , ce nombre correspond à 1%, 3% et 5% respectivement de l'ensemble du jeu de données.

nb\_PR = Taille (SimREO $_1$ ) \* TFS, [nombre d'erreurs prédites, c.-à-d. la taille de l'ensemble de test ; ce nombre est fixe pour un jeu de données et une valeur de TFS]

Dans le cas par exemple du jeu de données Android et avec  $TFS = 0,1$ , nb\_PR est égal à 46. Pour les jeux de données d'Eclipse et de Company A et avec  $TFS = 0,1$ , nb\_PR est égal respectivement à 42 et 68 rapports d'erreurs. Pour les plus grandes valeurs du paramètre TFS, le nombre d'erreurs prédites augmente proportionnellement. Avec  $TFS = 0,5$  et le jeu de données Android, nb\_PR est égal à 234, c.-à-d. 5% de la taille du jeu de données dans sa totalité.

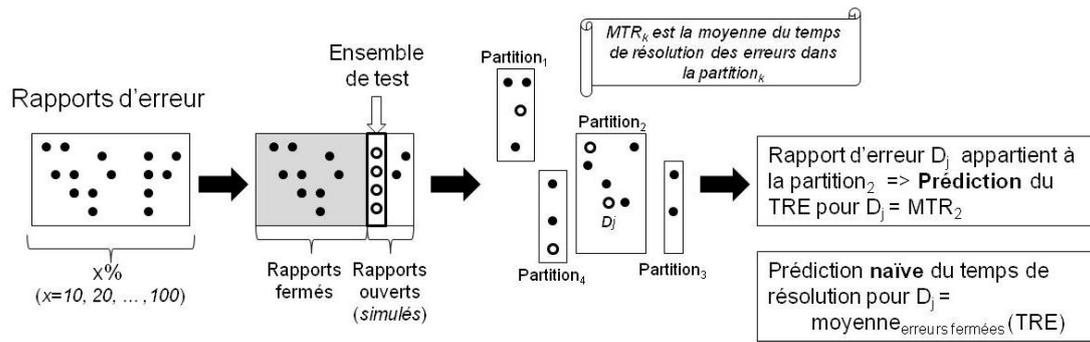


Fig. IV-9. Synopsis du scénario de test pour la prédiction du TRE

A chaque point dans le temps, nous utilisons les informations obtenues à l'aide du partitionnement pour prédire le TRE d'un rapport d'erreur nouvellement arrivé (cf. Fig. IV-9). Nous rejouons ainsi l'historique du système de gestion des erreurs, et nous simulons un afflux de rapports d'erreurs. Notre scénario de test comporte 10 itérations expérimentales. Pour chaque itération, un nombre fixe de rapports d'erreurs récents sont simulés comme étant ouverts (1% de la totalité des données à TFS = 0,1), et sont utilisés pour évaluer l'approche de prédiction.

Pour mesurer la qualité de prédiction de l'approche à chaque point dans le temps, nous utilisons comme indicateur de précision la *Magnitude Relative de l'Erreur*, défini par Shepperd & Kadoda (2001) comme suit [210]:

$$MRE(d) = \frac{|e - \hat{e}|}{e} \quad \text{où } e = \text{valeur réelle de la variable, et } \hat{e} = \text{valeur estimée de la variable}$$

Le **pouvoir prédictif** de l'approche est évalué en calculant la fraction des TRE prédits où les MRE sont inférieures à un certain seuil, 25% et 50% respectivement. Ce qui correspond à la fraction des TRE prédits qui se situent à  $\pm 25\%$  et  $\pm 50\%$  des TRE réels. Ces deux indicateurs du pouvoir prédictif sont appelés respectivement **Pred (0,25)** et **Pred (0,5)**. On remarquera que nous considérons les différences *absolues* entre la valeur réelle et celle prédite, c.-à-d. que la surestimation et la sous-estimation sont tout aussi erronées, conformément à la définition du MRE et à la mesure pratiquée dans l'étude de Weiss et al. (2007) [242]. Pred (x) est calculé de la manière suivante:

$$\begin{aligned} N(k) &= \text{Nombre d'erreurs à prédire appartenant à la partition } k \\ \sum N(k) &= \text{Le nombre total d'erreurs à prédire dans un jeu de données, ce nombre est fixe pour toutes les tranches du jeu de données (pour une certaine valeur du paramètre TFS)} \\ TR(d, k) &= \text{Le temps } \textit{réel} \text{ de résolution pour l'erreur } d \text{ dans la partition } k \\ TRpred(d, k) &= \text{Le temps de résolution prédit pour l'erreur } d \text{ dans la partition } k \\ Res(d, k, x) &= 1 \text{ si } \frac{|TRpred(d,k) - TR(d,k)|}{TRpred(d,k)} < x, \quad 0 \text{ sinon} \\ Pred(x) &= \frac{\sum Res(d,k,x)}{\sum N(k)}, \quad k=1, \dots, 4 \end{aligned}$$

Mesurer Pred (x) dans dix points dans le temps nous permet d'évaluer la façon dont la disponibilité d'un nombre accru de rapports affecte le pouvoir prédictif. Pour permettre une comparaison avec une approche de prédiction moins complexe, nous estimons également les temps de résolution des rapports d'erreurs nouvellement arrivés en utilisant la moyenne du TRE de *tous* les rapports d'erreurs disponibles, c.-à-d. en prenant la moyenne du temps de résolution pour toutes les erreurs fermées (cf. Fig. IV-9). Nous nous référons à cette approche en tant que **prévision naïve**, en ligne avec l'étude de Weiss et al. (2007). Par conséquent, la même formule Pred (x) pour la mesure de la puissance prédictive sera appliquée pour les prévisions naïves, et sera utilisée à des fins de comparaison, c.-à-d. pour évaluer l'apport de l'approche de prédiction basée sur le partitionnement automatique.

### IV.6.2 Premiers résultats

Comme suggéré dans l'étude initiale, l'approche de prédiction testée ici repose entièrement sur la présomption de **signifiante statistique** des différences entre les moyennes de TRE des partitions de rapports d'erreurs (cf. §IV.3.2). Par conséquent, après l'étape de partitionnement automatique, la deuxième étape est de vérifier cette condition pour tous les jeux de données utilisés dans le scénario de simulation. Nous réalisons les deux tests statistiques les plus importants pour vérifier cette condition : le test ANOVA pour l'analyse de la variance, et le test post-hoc *Games-Howell* pour analyser les variations des moyennes de TRE entre les différentes partitions (cf. §IV.3.2). Le Tableau IV-7 et la Fig. IV-10 présentent respectivement les résultats des tests ANOVA et ceux de l'analyse post-hoc.

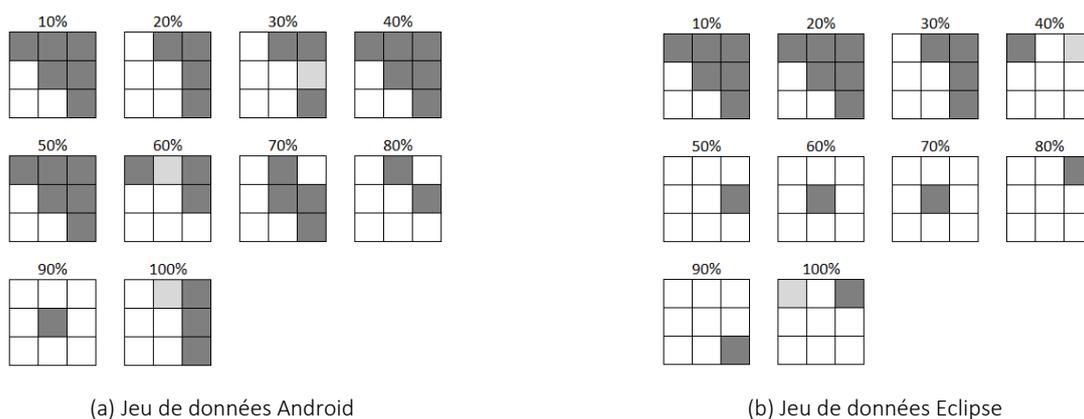
Sous-ensemble	Android		Eclipse		Company A	
10%	(F= <b>1.551</b> )	<b>0.201</b>	(F= <b>0.383</b> )	<b>0.765</b>	(F=11.45)	0.0000 ***
20%	(F=3.903)	0.0087 **	(F= <b>2.392</b> )	<b>0.0673</b>	(F=27.12)	0.0000 ***
30%	(F=2.791)	0.0393 *	(F=8.157)	0.0000 ***	(F=36.6)	0.0000 ***
40%	(F= <b>0.176</b> )	<b>0.912</b>	(F=18.43)	0.0000 ***	(F=28.42)	0.0000 ***
50%	(F= <b>2.441</b> )	<b>0.0625</b>	(F=19.74)	0.0000 ***	(F=33.9)	0.0000 ***
60%	(F=8.381)	0.0000 ***	(F=20.82)	0.0000 ***	(F=30.95)	0.0000 ***
70%	(F= <b>2.745</b> )	0.0416 *	(F=46.72)	0.0000 ***	(F=43.21)	0.0000 ***
80%	(F=9.374)	0.0000 ***	(F=13.8)	0.0000 ***	(F=32.42)	0.0000 ***
90%	(F=10.9)	0.0000 ***	(F=39.58)	0.0000 ***	(F=89.25)	0.0000 ***
100%	(F=13.47)	0.0000 ***	(F=21.44)	0.0000 ***	(F=92.99)	0.0000 ***

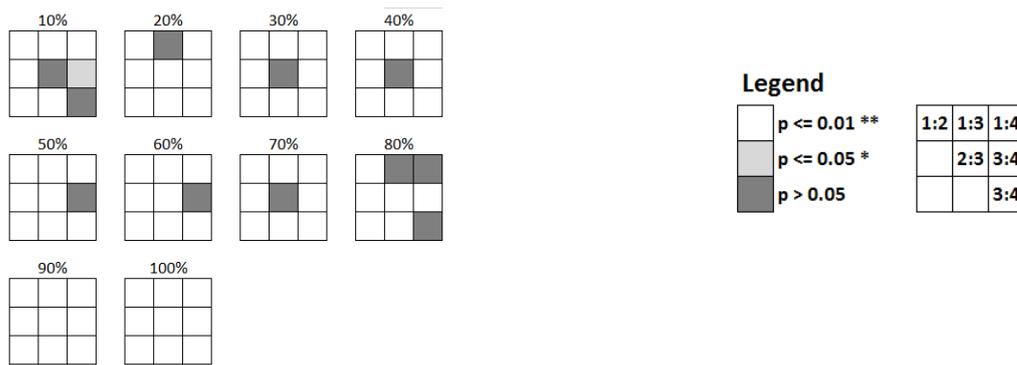
Tableau IV-7. Résultats du test ANOVA pour les partitions obtenues dans chaque tranche de données du scénario de simulation (les valeurs en gras ne sont pas significatives à la limite  $\alpha=0.05$ )

Les résultats sont légèrement différents pour les trois jeux de données : pour celui de Company A, le test ANOVA est positif pour l'ensemble des 10 tranches de données ; pour les deux autres (Android et Eclipse), le test a tendance à être négatif pour les petites tranches. L'analyse post-hoc confirme une certaine différence entre les jeux de données. En termes de variation du TRE inter-partition, les tranches de données de Company A sont plus conformes aux conditions statistiques comparées aux deux autres jeux de données (cf. Fig. IV-10).

Ainsi, les hypothèses théoriques de la méthode de prédiction ne sont pas systématiquement respectées par les tranches de données dans le scénario de simulation. Cependant, bien que le test ANOVA ne soit pas positif pour certains sous-ensembles Android et Eclipse, nous les avons inclus de toute façon dans le scénario de simulation. En effet, il est intéressant de voir si la violation de l'hypothèse statistique a un effet sur les résultats du scénario de simulation, c.-à-d. si des disparités peuvent être observées en termes de précision de la prédiction.

Les figures 11 et 12 visualisent les résultats du scénario de test. Sur ces figures, le trait rouge plein correspond à la précision de prédiction avec l'approche par partitionnement automatique, tandis que le trait bleu pointillé correspond à la précision de la prédiction « naïve » (cf. §IV.6.1). Chaque figure présente trois graphiques de précision, un pour chaque jeu de données : Android, Eclipse et Company A respectivement. De plus, chaque figure correspond à un seuil de précision différent, c.-à-d., Pred (0,25) et Pred (0,5) respectivement.





(c) Jeu de données Company A

Fig. IV-10. Résultat du test post-hoc de *Games-Howell* pour les partitions pour chaque jeu de données

La Fig. IV-11 présente les graphes de précision de  $Pred(0,25)$ , c.-à-d. les fractions des prédictions où la marge d'erreur par rapport au TRE réel ne dépasse pas 25%. Les graphiques semblent très irréguliers et la précision est globalement inférieure à 20%, c.-à-d. que dans seulement 20% des cas, la précision des prédictions du TRE a été dans une la marge d'erreur de 25%. Dans la plus petite tranche du jeu de données Eclipse, c.-à-d. la tranche 10%, la précision est meilleure qu'avec la totalité des données, mais elle tombe à 0 pour d'autres tranches. Enfin, par rapport à l'approche « naïve », la prédiction du TRE par partitionnement fournit rarement des résultats de meilleure qualité. Cela est vrai pour les trois jeux de données ; de plus, la prévision « naïve » est parfois un peu plus précise.

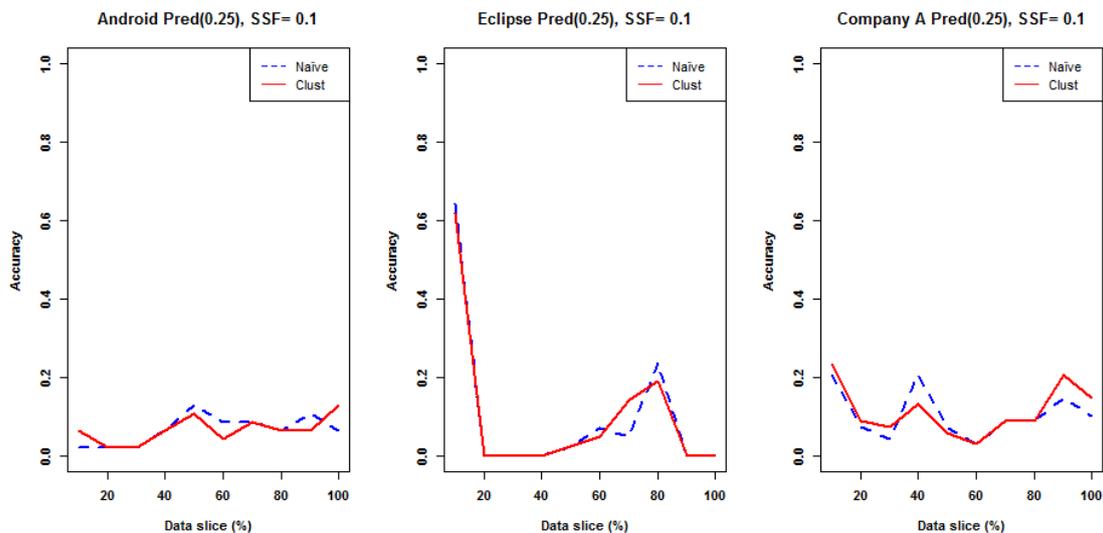


Fig. IV-11. La précision de la prédiction du TRE pour les trois jeux de données, seuil d'erreur  $x = 25\%$

Les graphes de précision à  $Pred(0,5)$ , c.-à-d. avec un seuil d'erreur de 50%, sont présentés dans la Fig. IV-12. A ce niveau relativement élevé de la marge d'erreur, la précision s'améliore. Elle atteint 40 à 60% pour les tranches 10 et 20% pour le jeu de données Eclipse. Néanmoins, on observe encore les mêmes motifs irréguliers. La précision pour le jeu de données de la Company A paraît de nouveau légèrement plus régulière que pour Eclipse ; mais l'approche naïve est, dans tous les cas, presque aussi précise que l'approche par partitionnement, une observation qui vaut également pour le jeu de données Android.

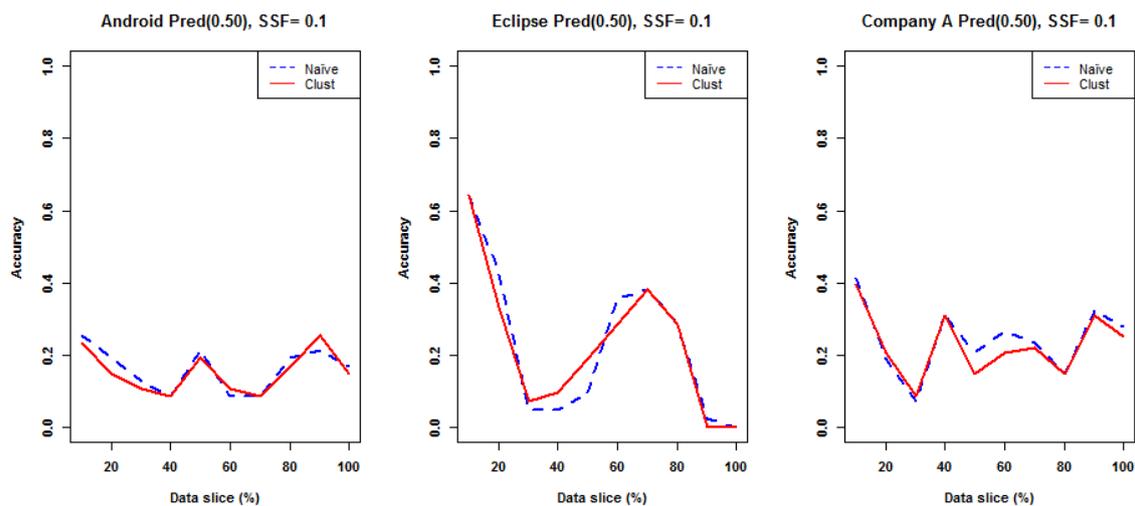


Fig. IV-12. La précision de la prédiction du TRE pour les trois jeux de données, seuil d'erreur  $x = 50\%$

Dans l'analyse des graphes de précision et la relation possible avec les résultats du test ANOVA et les tests post-hoc (cf. Tableau IV-7, Fig. IV-11 et Fig. IV-12), aucune tendance claire ne se dessine. Pour les tranches de données où le test ANOVA est négatif, c.-à-d. les tranches 10%, 40%, 50% d'Android et les tranches 10% et 20% d'Eclipse (cf. Tableau IV-7), la précision de la prédiction est aussi irrégulière que pour le reste. Pour Android, la précision de la prédiction pour les tranches 40% et 50% est proche de la moyenne, c.-à-d.  $\sim 10\%$  à Pred (0,25) et  $\sim 15\%$  à Pred (0,50) ; de plus, la précision de la prédiction ne semble pas être affectée par le résultat négatif du test ANOVA. Le cas d'Eclipse paraît particulièrement chaotique : pour la tranche 10%, la précision est élevée à la fois pour les seuils 0,25 et 0,50, bien que le test ANOVA soit négatif, et la précision est nulle pour les tranches 90% et 100%, même si le test ANOVA est positif. Bref, on ne constate aucune corrélation entre le résultat du test ANOVA et la précision de prédiction.

### IV.6.3 Modification de la taille de l'ensemble de test

Comme indiqué précédemment (cf. §IV.6.1), pour permettre la comparaison des résultats obtenus avec des partitions de taille différentes, le scénario de simulation utilise un nombre fixe de points de test (cf. Fig. IV-9). Toutefois, la taille des sous-ensembles d'erreur qui sont considérés pour la prédiction peut être modifiée. Ce changement dans la procédure de simulation est mis en œuvre à travers le paramètre TFS (la Taille du Facteur de Simulation, cf. §IV.6.1). Les résultats concernant la précision des prédictions ont été obtenus jusqu'à présent avec TFS = 0,1, qui fixe le nombre d'erreurs prévus égale à 1% de l'ensemble de données, c.-à-d.  $0,1 * \text{Taille}(SL_1)$ . Dans ce qui suit, nous présentons les résultats de simulations avec des valeurs **plus grandes pour TFS**. Les valeurs de TFS plus grandes impliquent qu'un plus grand nombre de prédictions sont faites, augmentant ainsi la puissance statistique des étapes d'évaluation dans notre procédure de simulation.

Nous avons réalisé la procédure de simulation avec des valeurs de TFS égales 0,3 et 0,5. Dans la Fig. IV-13, nous avons combiné dans le même graphique les résultats obtenus précédemment avec TFS = 0,1 avec les nouveaux obtenus avec TFS = 0,3 et TFS = 0,5. Cependant, pour des raisons de lisibilité, la prédiction naïve n'est pas cette fois-ci présentée. En regardant séparément chaque jeu de données, il est intéressant de voir que pour Android et Company A, les formes des courbes ne divergent pas et ont tendance à s'aplatir quand TFS augmente. Néanmoins, pour le jeu de données Eclipse, les résultats sont toujours aussi irréguliers pour toutes les valeurs de TFS.

De cette exploration des données, nous pouvons établir que la précision de la prédiction se situe, approximativement, entre 10 et 15% pour Pred(0,25) et entre 15 et 20% pour Pred(0,5) avec les trois jeux de données. On constate des variations particulières pour ce résultat, avec le jeu de données d'Eclipse qui est **moins homogène** que les deux autres, et le jeu des données de Company A qui est légèrement **plus homogène**. Pour cette raison, nous allons dans la prochaine section nous concentrer sur le jeu de données de Company A et explorer l'effet potentiel du nombre de partitions sur la précision de prédiction.

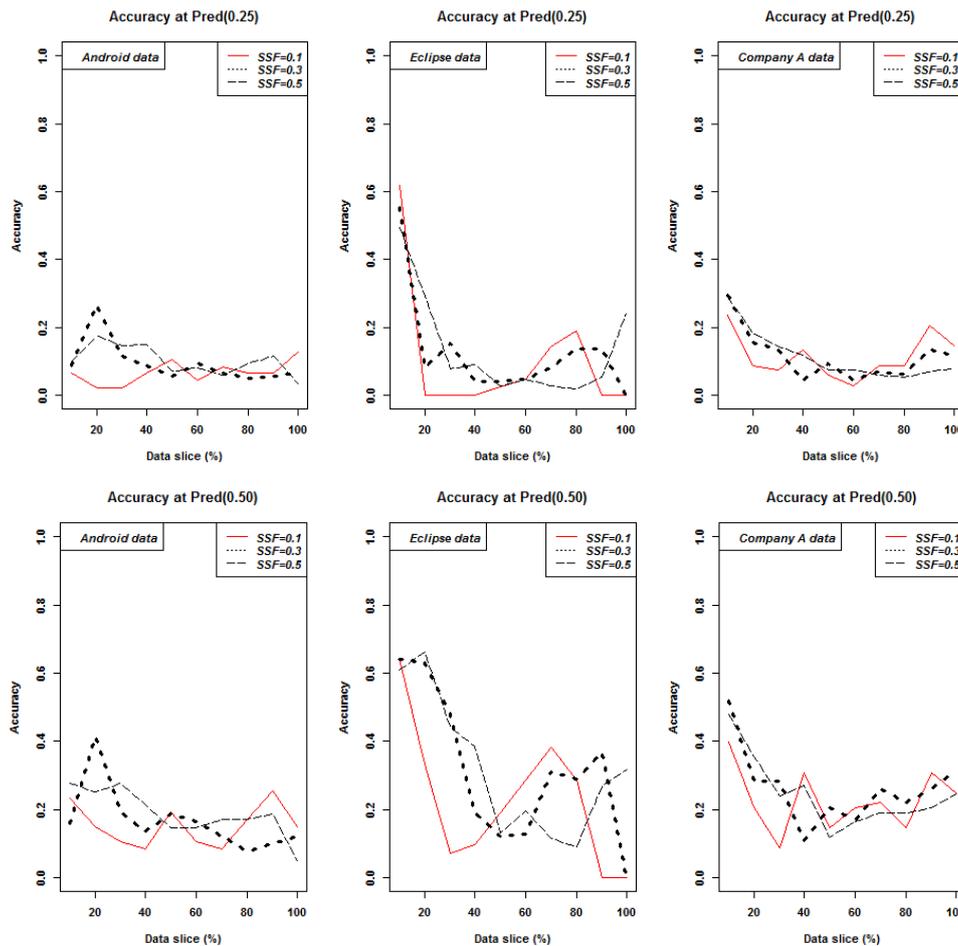


Fig. IV-13. La précision de prédiction du TRE pour les trois jeux de données avec des valeurs différentes du paramètre *Taille du Facteur de Simulation* (0.1, 0.3 et 0.5), et avec un seuil d'erreur de 25% et 50%

#### IV.6.4 Augmentation du nombre de partitions (K = 6, 8 et 10)

Une question importante, à la fois dans l'étude initiale et dans la notre, est le **nombre K de partitions**. Comme indiqué précédemment (cf. §IV.3.2), l'expérience initiale de Raja (2013) s'est appuyée sur une technique de partitionnement sophistiquée, une technique qui comprend plusieurs itérations et interventions humaines. L'auteure a indiqué qu'elle a commencé avec un nombre maximal par défaut de partitions, et que « *plusieurs itérations conduisent à la solution optimale* » (Raja 2013, p.126), avec un nombre final de partitions entre 3 et 5.

La technique de partitionnement que nous utilisons dans la réplcation et dans le scénario de simulation est destinée à être entièrement automatisée et sans intervention humaine ; ceci pour des raisons pratiques, c.-à-d. son utilisation quotidienne pour faire des prédictions du TRE (cf. §IV.6.1). Dans notre réplcation conceptuelle, nous avons délibérément fixé le nombre de partitions à 4 (cf. §IV.5.2). Toutefois, le nombre de partitions peut éventuellement être un facteur confondant qui impacterait la précision de l'approche de prédiction.

Pour explorer cette question, nous exécutons l'étape de partitionnement avec un nombre de partitions plus élevé. Pour définir l'intervalle de variation de ce nombre, nous nous sommes appuyés sur la description de l'expérience initiale dans Raja (2013), et qui indique qu'au cours de l'étape itérative de partitionnement, « *aucun des projets n'a abouti à plus de dix partitions* » (p. 126). Nous avons donc considéré **K = 10** comme une valeur significativement grande pour le partitionnement des rapports d'erreurs, et nous avons décidé d'explorer trois valeurs supplémentaires pour K : 6, 8 et 10. Nous avons choisi d'explorer l'effet potentiel de K sur le jeu de données de Company A uniquement. C'est le plus grand ensemble de données, et il a fourni la meilleure précision de prédiction avec K = 4 pour Pred (0,25). Par ailleurs, les conditions statistiques du test ANOVA et du test post-hoc ont toutes les deux été satisfaites pour toutes les tranches de ce jeu de données.

Nous avons réalisé le test ANOVA sur le jeu de données de la Company A pour chaque tranche, avec K fixé à 6, 8 et 10. Les résultats de ces tests ont été positifs dans tous les cas, et ils ne sont pas présentés ici. Cependant, les résultats de l'analyse post-hoc ne sont pas aussi nets que pour le partitionnement initial avec K = 4 (cf. Fig. IV-10). La Fig. IV-14 présente les résultats du test post-hoc pour K = 6 et K = 8. Les résultats s'améliorent dans les plus grandes tranches de données, et pour les tranches 80% à 100%, le test est presque entièrement positif. Nous ne montrons pas le test post-hoc pour K = 10, les résultats étaient très similaires au cas K = 8.

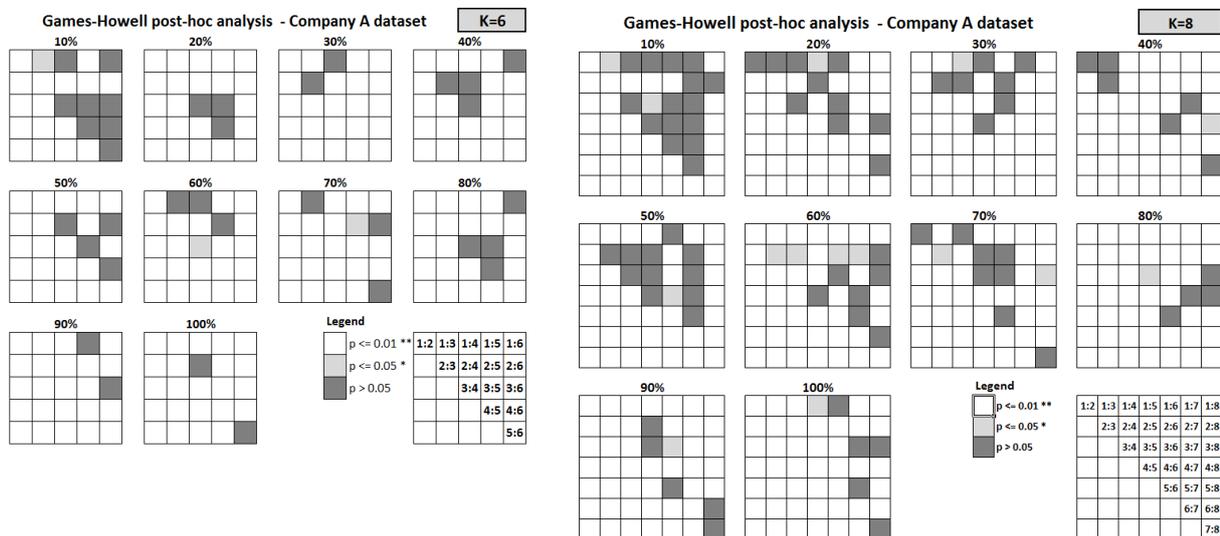


Fig. IV-14. Résultats du test post-hoc *Games-Howell* pour les tranches de données de Company A avec K=6 et K=8

Pour la simulation, comme les prédictions se font à partir d'un plus grand nombre de partitions, l'ensemble de test doit être suffisamment grand. Ainsi, nous réalisons le scénario de simulation avec le paramètre TFS (*Taille du Facteur de Simulation*) = 0,3 et 0,5. Ce qui aboutit à des ensembles de test de taille 204 et 340 rapports respectivement.

Les résultats du scénario de simulation sont présentés dans la Fig. IV-15. Au seuil d'erreur de 25%, c.-à-d. Pred (0,25), la courbe de la précision de prédiction montre une tendance claire et homogène qui passe de ~30% au début où les tranches de données sont relativement petite (soit 30% des prévisions au sein de la plage définie par le seuil), à ~20% plus tard. En comparant la forme de la courbe de précision pour K = 4 (cf. Fig. IV-13) et K = 6, 8 et 10 (cf. Fig. IV-15), en particulier pour TFS = 0,5, les résultats sont très similaires et semblent donc être indépendants du nombre de partitions. Au seuil de 50% d'erreur, c.-à-d. Pred (0,5), les courbes sont moins régulières, mais la précision de prédiction est plus élevée, à environ 45% dans les premiers points du processus de simulation.

### IV.6.5 Conclusion sur la partie 2

Dans la section précédente (cf. §IV.5), le succès de la réplication conceptuelle soutient la suggestion de Raja (2013) que « *la classification textuelle des rapports d'erreurs peut être un indicateur précoce et utile du temps de résolution d'une erreur* » (p. 135). L'idée derrière question de recherche RQ2 est d'explorer une méthode basée sur le partitionnement automatique pour prédire le TRE. Les résultats obtenus montrent que, non seulement la qualité prédictive est globalement faible (même pour Pred (0,5)), mais aussi, qu'on ne constate aucun motif visible qui indiquerait dans quelle situation les prédictions sont de qualité supérieure ou inférieure. En particulier, on ne trouve aucune indication qu'un **plus grands ensembles de rapports fermés** donnent de **meilleurs résultats** pour prédire le TRE. Fait intéressant par ailleurs, dans de nombreux cas, le partitionnement n'a pas amélioré la qualité prédictive par rapport une approche de prédiction naïve qui prend la moyenne des TRE de tous les rapports d'erreurs fermés (cf. Fig. IV-11 et Fig. IV-12). De plus, nous avons vérifié l'éventuelle influence de certains paramètres de l'expérience tels que la taille des ensembles de prédiction ou le nombre de partitions ; les résultats sont globalement inchangés (cf. Fig. IV-13 et Fig. IV-15), la précision moyenne pour Pred(0,25) est comprise entre 10 et 15%, et pour Pred (0,50) entre 15 et 30%.

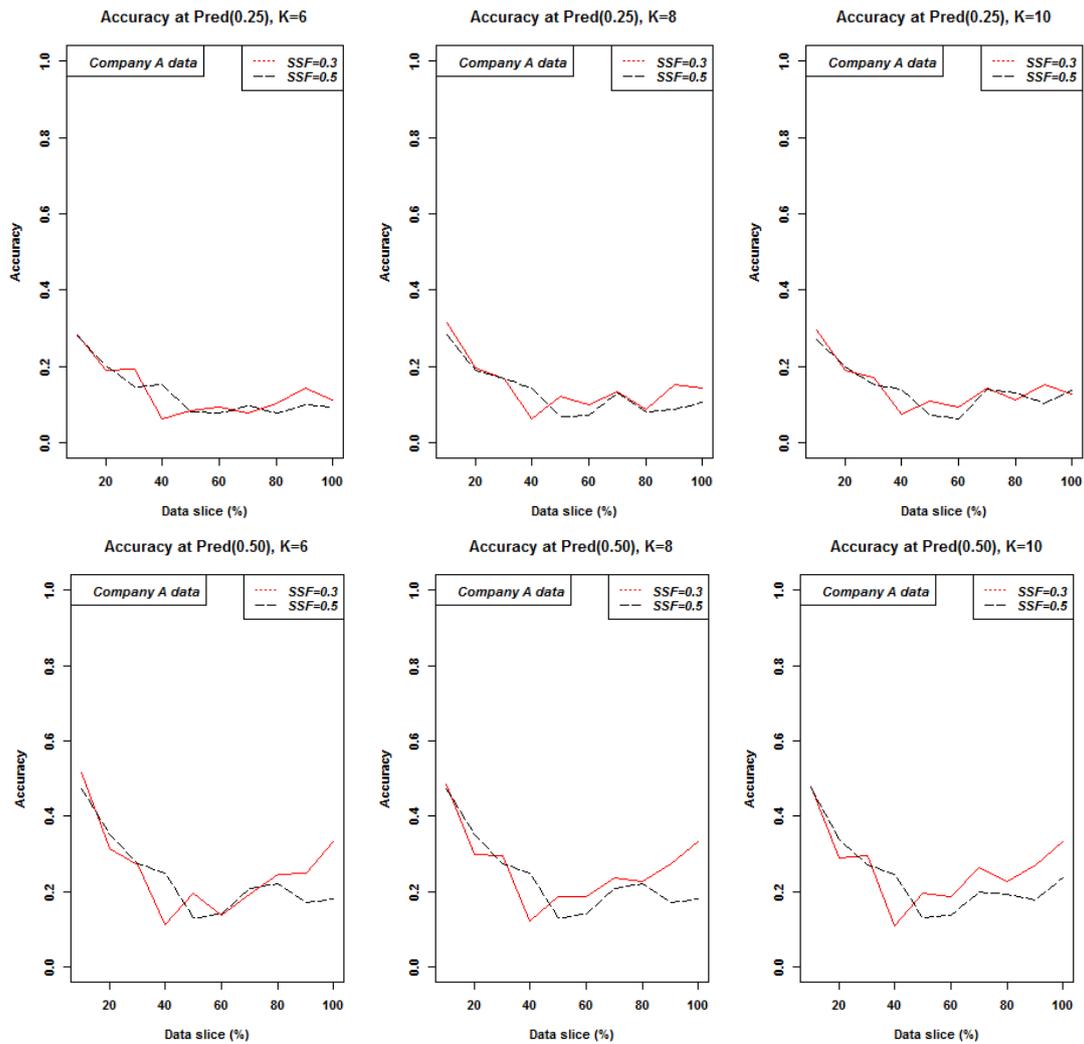


Fig. IV-15. La précision de prédiction du TRE pour des valeurs différentes du nombre K de partitions, en utilisant deux valeurs du paramètre *Taille du Facteur de Simulation* (0.3 et 0.5), et avec un seuil d'erreur de 25% et 50%

En outre, les résultats ne semblent pas être affectés par l'hypothèse théorique des différences statistiquement significatives entre les moyennes des TRE entre partitions (test ANOVA et test post-hoc). Dans le cas du jeu de données Eclipse, lorsque le test ANOVA est négatif (cf. Tableau IV-7), la précision de la prédiction est supérieure à la moyenne, et lorsque le test est positif, la prédiction est très faible, voire nulle. Ceci est également confirmé lors de l'exploration des données Company A avec un plus grand nombre de partitions, il n'y a pas de corrélation visible entre les résultats du test post-hoc (cf. Fig. IV-14) et la précision de la prédiction (cf. Fig. IV-15). Nous aurions ainsi tendance à en conclure que la suggestion de Raja sur l'utilité du partitionnement textuel comme un indicateur précoce du TRE, ne peut pas être appliquée dans la pratique, et surtout, ne semble pas produire une prédiction précise de TRE.

Enfin, on peut s'interroger sur la pertinence d'une approche de prédiction basée uniquement sur le texte d'un rapport d'erreur. L'opinion prévaut dans la littérature que la prédiction du TRE doit être basée sur davantage que la description textuelle de l'erreur (p.ex. [125]). Cependant, il n'est pas clair quelles sont les caractéristiques d'une d'erreur qui sont les plus pertinentes. Dans Giger et al. (2010) par exemple, un algorithme d'arbre de décision est appliquée pour classer les rapports d'erreurs entrants en deux catégories : temps de résolution rapide (moins de la médiane du TRE) ou lent (plus de la médiane du TRE). Les meilleurs modèles de prédiction sont obtenus lors de l'utilisation d'informations supplémentaire (p. ex. un commentaire ou un jalon de projet). De plus, un autre problème pourrait être une qualité insuffisante du rapport d'erreur lui-même. En effet, les approches de prédiction basées sur le texte du rapport d'erreur reposent sur une hypothèse forte : un rapport est une description correcte et complète

d'une erreur; c.-à-d. qu'une même erreur signalée par deux utilisateurs différents utiliserait les mêmes termes, et le texte des deux rapports auraient un degré élevé de similarité. La validité de cette hypothèse est à priori discutable.

## IV.7 Conclusion et perspectives de recherche

En conclusion, notre réplique a confirmé les résultats obtenus par Raja (2013). La valeur méthodologique d'une réplique réside dans l'analyse des résultats en relation avec l'expérience initiale et au vu des variations apportées au contexte expérimental. Si les résultats sont conformes, en particulier si un ou plusieurs facteur(s) ont été changés, les résultats deviennent cumulatifs et augmentent la confiance dans l'hypothèse de départ [159].

Cependant, comme l'ont suggéré nos résultats, la **valeur pratique** de ce résultat en ce qui concerne la prédiction de TREs est plutôt **limitée**. Compte tenu des résultats d'autres chercheurs, ce constat n'a rien de surprenant. Alors que, selon Weick (1995), « *la réponse différenciée des données à des changements dans un traitement est fréquemment une information annonciatrice d'une théorisation à venir* » ([241], p.388), nous devons admettre que dans notre cas, B. Kitchenham a raison lorsqu'elle affirme ne pas être convaincue « *qu'on puisse trouver des théories en faisant simplement varier les conditions expérimentales* » [126], et que – comme ce fut le cas dans notre cas – « *une réplique, qu'elle soit dépendante ou indépendante sans un cadre théorique est, de loin, le type le plus risqué de réplique* ». On peut remarquer en effet que notre étude empirique repose sur **deux principes théoriques** : le premier est le principe de **similitude**, à savoir, des erreurs similaires nécessitent des temps de résolution similaires ; le second est le principe de **partitionnement**, à savoir, des erreurs similaires sont regroupées dans la même partition. Ces principes semblent insuffisants pour expliquer pleinement nos résultats empiriques, et d'autres investigations sont nécessaires pour mieux comprendre les relations entre les multiples facteurs qui peuvent affecter la similarité d'erreurs (p.ex. la gravité de l'erreur) et le temps de résolution (p.ex. l'expertise du développeur ou les spécificités d'un projet).

Les perspectives de travaux futurs seront guidées par ces observations. Une première piste serait de remettre en question **le principe de similitude** en utilisant une expérience contrôlée : est-ce que des individus différents décrivent des erreurs similaires – ou même identiques – avec des rapports textuels similaires? En outre, une seconde piste serait d'explorer **les facteurs confondants** dans les projets logiciels ayant un impact potentiel sur la similarité de l'erreur, au-delà de la similarité purement textuelle. Cela renforcerait la compréhension théorique sur la meilleure manière de décrire une erreur et comment cette information peut être exploitée dans les référentiels de logiciels pour la navigation, la recommandation et la prédiction [25]. Une dernière piste est d'explorer la **similarité sémantique** de deux rapports d'erreur en utilisant une connaissance du domaine, c.-à-d. une connaissance sur le logiciel lui-même, ses fonctionnalités et ses finalités. Cette connaissance peut prendre la forme d'une **ontologie**, et créer des catégories liées au domaine du logiciel avec lesquels l'erreur est liée. Le partitionnement sémantique à l'aide d'ontologie a en effet été expérimenté récemment dans d'autres domaines de recherche et a montré des résultats prometteurs [14].

# V. Chapitre V : Enseignement en ligne et prise en compte des styles d'apprentissage

## Sommaire

<b><u>V.1</u></b>	<b><u>Préambule</u></b>	<b>80</b>
<b><u>V.2</u></b>	<b><u>Problématique</u></b>	<b>80</b>
<u>V.2.1</u>	<u>Caractéristiques des apprenants, styles cognitifs et styles d'apprentissage</u>	81
<u>V.2.2</u>	<u>Constats empiriques de l'effet des styles d'apprentissage</u>	82
<u>V.2.3</u>	<u>Personnalisation de l'enseignement et systèmes adaptatifs</u>	82
<b><u>V.3</u></b>	<b><u>Intégration des styles d'apprentissage dans un dispositif d'enseignement</u></b>	<b>83</b>
<u>V.3.1</u>	<u>Le modèle de Felder &amp; Silverman</u>	83
<u>V.3.2</u>	<u>Proposition d'une grille d'adaptation des contenus multimédia aux styles d'apprentissage</u>	84
<u>V.3.3</u>	<u>Evaluation empirique de la grille d'adaptation</u>	86
<u>V.3.4</u>	<u>Conclusion</u>	86
<b><u>V.4</u></b>	<b><u>Impact des styles d'apprentissage dans l'adoption de l'apprentissage mobile</u></b>	<b>87</b>
<u>V.4.1</u>	<u>Le Mobile Learning</u>	87
<u>V.4.2</u>	<u>Les modèles d'acceptation de la technologie</u>	87
<u>V.4.3</u>	<u>Le modèle des styles d'apprentissage de Kolb</u>	88
<u>V.4.4</u>	<u>Proposition de nouveaux modèles théoriques</u>	90
<u>V.4.5</u>	<u>Enquête quantitative – phase 1</u>	90
<u>V.4.6</u>	<u>Enquête quantitative – phase 2</u>	91
<u>V.4.7</u>	<u>Analyse des données et modélisation PLS</u>	92
<b><u>V.5</u></b>	<b><u>Conclusion et perspectives de recherche</u></b>	<b>94</b>

## V.1 Préambule

Les travaux présentés dans ce chapitre relèvent du thème 2 (cf. chapitre I), c.-à-d. des usages et de l'adoption des artefacts informatiques et sociotechniques ; la perspective d'étude est celle des sciences des faits. Le domaine d'activité concerné est celui de l'enseignement en général, et porte sur l'enseignement en ligne en particulier. En tant que champ d'utilisation des TIC, l'enseignement possède des spécificités dues au fait que c'est une activité fortement cognitive, qui inclut des interactions sociales multiples, et où des facteurs liés à la psychologie et la personnalité de l'apprenant et son contexte d'apprentissage peuvent fortement influencer sur le déroulement de ce processus et sur son résultat. L'informatique et les TIC, par leur capacité à transformer l'écrit et les moyens d'interaction et de communication, apportent beaucoup d'éléments nouveaux aux activités d'enseignement et d'apprentissage.

Les travaux présentés ont été réalisés dans le cadre de deux thèses de doctorat co-encadrées à 80% et 50% respectivement; elles ont donné lieu à la publication de 2 articles dans des revues internationales, 4 articles dans des conférences internationales avec comité et actes, et 2 communications dans des ateliers ou conférences sans actes. Par ailleurs, j'ai été invité à rédiger un article d'encyclopédie sur le thème des TIC dans l'éducation [publi#15].

## V.2 Problématique

L'enseignement et l'apprentissage sont des sujets de recherche bien établis en sciences humaines. Les questions fondamentales qui sont posées concernent, d'une part, la description et la compréhension des processus sous-jacents d'un point de vue cognitif, psychologique et communicationnel, et d'autre part, les facteurs et modalités qui influent sur ces processus et susceptibles d'être des leviers pour les maîtriser et les améliorer. L'informatique et les technologies de l'information rentrent dans le champ d'étude de ces problématiques et y contribuent car, d'une part l'informatique s'est fortement intéressée aux processus cognitifs d'apprentissage et d'interaction entre l'homme et la machine, à travers notamment l'Intelligence Artificielle. D'autre part, grâce aux possibilités offertes par le réseau internet et les supports multimédias numériques, l'informatique et les TIC modifient en profondeur les modalités de communication et d'échange entre l'enseignant et l'apprenant.

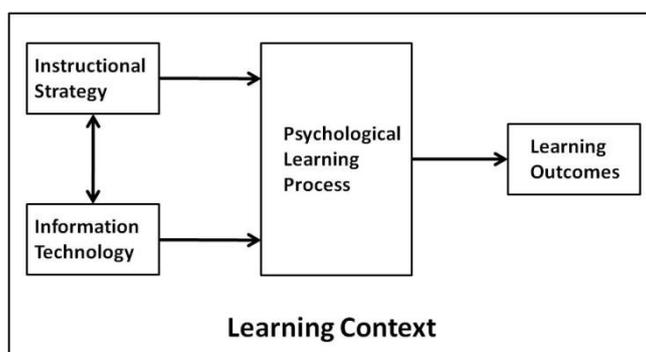


Fig. V-1. Cadre d'analyse pour la recherche sur l'apprentissage médiatisé par les TIC ([3], p.5)

Le cadre d'analyse d'Alavi et Leidner clarifie les éléments interagissant dans une action de recherche relative à l'introduction de l'informatique et des TIC dans l'univers de l'enseignement et de l'apprentissage ([3], p.5). Dans ce cadre, le résultat de l'activité d'apprendre est impacté par le processus psychologique d'apprentissage, lui-même est impacté par les TIC et par la stratégie – ou pédagogie – d'enseignement (cf. Fig. V-1).

Ce cadre d'analyse met en évidence deux points importants pour la suite de cet exposé : la dimension psychologique du processus d'apprentissage, et l'interdépendance de la technologie avec la stratégie d'enseignement. Les travaux présentés dans le reste de ce chapitre font appel à des théories en psychologie qui visent à identifier des catégories de comportement chez les apprenants, des « *styles d'apprentissage* », pour mieux maîtriser le processus pédagogique d'enseignement et garantir son succès au plus grand nombre d'apprenants, quelle que soit la catégorie à laquelle ils appartiennent.

### V.2.1 Caractéristiques des apprenants, styles cognitifs et styles d'apprentissage

La majorité des praticiens de l'enseignement reconnaissent que chaque apprenant a une manière, plus ou moins spécifique et particulière, d'apprendre et d'assimiler l'information et la connaissance. Cette manière d'apprendre peut avoir un certain impact sur la performance de l'apprenant et ses résultats scolaires. Ce constat se nourrit par ailleurs de recherches en psychologie de l'éducation et dans les sciences cognitives, que le psychologue C. Jung a initié dans les années 1920 avec ses travaux sur les traits de personnalité et les profils psychologiques [120]. Ainsi, certains apprenants sont satisfaits avec des théories et des abstractions ; d'autres sont plus à l'aise avec des faits et des phénomènes observables ; certains préfèrent l'apprentissage actif et d'autres penchent vers l'introspection ; certains préfèrent la présentation visuelle de l'information et d'autres préfèrent des explications verbales ([74], p.58). Néanmoins, la possibilité d'identifier et caractériser ces différentes manières d'apprendre soulève beaucoup de difficultés et il y a une multitude d'approches pour le faire. De plus, du côté de l'enseignant, la tendance générale est d'enseigner selon des schémas génériques, parfois adaptés selon sa propre vision et pédagogie, alors que la prise en compte du profil de l'apprenant reste rare ([74], p.58).

Les concepts de **style d'apprentissage** (« Learning Style ») et **style cognitif** (« Cognitive Style ») ont été introduits pour décrire et aider à comprendre ce phénomène complexe. La distinction entre ces deux concepts est importante, même s'ils partagent plusieurs propriétés et sont parfois confondus. Le style cognitif est centré sur le fonctionnement de l'esprit humain en général, c'est un ensemble cohérent et constant de caractéristiques relatives à sa manière de traiter l'information et de résoudre des problèmes. Le style d'apprentissage est un concept similaire, mais le focus est accentué sur l'acquisition de connaissance et la mémorisation dans un objectif rattaché à l'enseignement scolaire, tout âge confondu ([186], p.7). Un style d'apprentissage n'est ni préférable ni inférieur à un autre, il est simplement différent, avec des forces et des faiblesses qui le caractérisent en terme de mode d'assimilation et de compréhension. Le but idéal d'un enseignement serait de donner aux apprenants les compétences associées avec chaque catégorie de style d'apprentissage, quel que soit leurs préférences personnelles ([74], p.58).

Pour illustrer ces idées, je vais présenter brièvement le travail précurseur de H.A. Witkin, datant des années 60 et dénommé la « dépendance/indépendance au champ », c.-à-d. *Field Dependence* et *Field Independence* [248]. La notion de « champ » ici renvoie au **contexte** dans lequel une information va être traitée. Les travaux de Witkin ont porté initialement sur des expérimentations sensorielles concernant la capacité d'un individu à identifier le sens horizontal. Ce qui a permis de mettre en évidence deux catégories d'individus : ceux qui arrivent à identifier le sens horizontal malgré la distorsion introduite par l'expérimentateur (c.-à-d. indépendant du champ), et ceux qui restent sous l'influence de cette distorsion (c.-à-d. dépendant au champ). Ces expérimentations ont été reproduites et généralisées à d'autres types de situations, et ont permis ainsi d'établir cette dichotomie importante. **L'indépendance au champ** correspondrait ainsi à un profil *analytique* qui traite les éléments d'un problème sans l'interférence d'éléments adjacents, qui est peu porté sur l'interaction sociale et la prise en compte du contexte social des relations humaines. **La dépendance au champ** correspondrait à un profil plus *holistique*, porté vers l'interaction sociale et à sa prise en compte dans sa manière de raisonner et d'agir. L'auteur cite ainsi le cas typique des infirmières chirurgicales qui, selon les résultats de ses tests, ont montré une nette orientation vers l'indépendance au champ, ce qui correspond à des compétences nécessaires pour analyser un problème et réagir indépendamment d'éléments contextuels ; alors que les infirmières psychiatriques avaient une nette orientation vers la dépendance au champ, ce qui coïncide avec leurs capacités avancées à interagir avec les malades et être réceptives à leurs détresses.

Plusieurs styles cognitifs et d'apprentissage ont été introduits par la suite, 23 modèles sont cités dans une étude en 2004 à ce sujet [38] ; et publié la même année, une revue systématique de la littérature évoque 71 modèles identifiés, parmi lesquels 13 modèles qui sont analysés en profondeur [44]. Deux éléments ressortent de ces deux études :

- La définition d'un style relève généralement d'une construction synthétique basée sur des théories psychologiques ou cognitives, ou encore sur des constats empiriques (p.ex. le modèle de Witkin). Néanmoins, le style nécessite à chaque fois la définition d'un instrument de mesure, généralement sous forme d'un questionnaire, pour identifier la catégorie à laquelle appartient un sujet. L'intérêt et la pertinence d'un modèle dépendra de la validité et la précision de l'instrument défini avec.

– Un grand nombre de controverses existent à propos des styles, et ces controverses portent autant sur la définition du modèle lui-même que sur l'instrument de mesure qui va avec. Néanmoins, même les critiques les plus sévères envers la théorie des styles (p.ex. [187]), reconnaissent que les catégorisations qu'elles mettent en évidence constituent une base utile pour réfléchir sur les processus d'apprentissage.

Dans les deux thèses que j'ai co-encadrées, nous avons utilisé deux modèles différents de styles d'apprentissage. Il s'agit du modèle de Felder & Soloman [75], développé à partir d'observations initiales dans les écoles d'ingénieurs ; et le modèle de Kolb [130], développé à partir d'observations dans l'apprentissage par expériences (« *experiential learning* »). Ces deux modèles seront présentés dans le reste de ce chapitre.

### **V.2.2 Constats empiriques de l'effet des styles d'apprentissage**

Un grand nombre d'études empiriques ont été menées pour constater l'effet supposé des styles d'apprentissage. La diversité des contextes dans lesquels ces études ont été menées, l'apparente contradiction de certains résultats, ainsi que la multiplicité des modèles utilisés alimentent la controverse sur l'intérêt et la valeur des styles d'apprentissage.

Si on se restreint au modèle de Kolb (ce modèle sera présenté un peu plus loin, cf. §V.4.3), nous avons identifié, dans le cadre d'une revue de la littérature dans la thèse de Y. Cruz, des dizaines d'études empiriques qui utilisent ce modèle ([50] p.109). Ces études analysent la corrélation statistique entre les résultats scolaires des apprenants et les styles d'apprentissage, généralement à l'aide de tests statistiques usuels (t-test ou ANOVA et ses variantes). Trois études en particulier, réalisées sur des échantillons d'assez grande taille ( $N > 300$ ), constatent une corrélation significative entre le style d'apprentissage de l'apprenant et les résultats scolaires de celui-ci ([238], [111], [160]). Et plusieurs autres études obtiennent des résultats négatifs en termes de corrélation statistique (utilisant les mêmes tests), mais avec des échantillons de taille un peu plus faible ( $N < 160$ ) (p.ex. [99], [133]).

### **V.2.3 Personnalisation de l'enseignement et systèmes adaptatifs**

Le principe de l'enseignement personnalisé est simple, il s'agit d'apporter les connaissances à chaque apprenant selon ses caractéristiques propres. Ces caractéristiques peuvent être liées à des facteurs contextuels concrets, tels que l'âge ou l'expérience antérieure, ou à des facteurs psychologiques et cognitifs tels que les styles d'apprentissage. Si les apprenants assimilent et mémorisent les connaissances selon des procédés qui leur sont propres et spécifiques, il est préférable pour l'enseignant de fournir une pédagogie adaptée au profil de chacun. Cette pédagogie différenciée est connue sous le vocable "*Aptitude-Treatment Interaction*" (ATI) et remonte aux années 1960 [252].

Avec l'irruption de l'informatique et des TIC dans le champ des activités d'enseignement, et les possibilités multiples qu'offre la technologie en tant que support au matériel pédagogique (texte, son, image, etc.), l'approche ATI devient le vecteur de ce qu'on peut appeler « un enseignement adapté » [215]. Et c'est dans ce contexte que les styles d'apprentissage trouvent un terrain d'application intéressant. S'il est en effet concrètement infaisable de prendre en compte le style d'apprentissage des apprenants dans une salle de classe, il est plausible de l'envisager dans le contexte d'un système informatisé [57].

Cette idée d'une correspondance intelligente, un « matching » personnalisé, entre la pédagogie et les spécificités de l'apprenant est le fondement des approches adaptatives dans l'enseignement en ligne. Elle constitue le sujet central de la thèse que j'ai co-encadré de A.-L. Franzoni.

### V.3 Intégration des styles d'apprentissage dans un dispositif d'enseignement

Dans cette recherche menée dans le cadre de la thèse de Ana-Lidia Franzoni de l'ITAM<sup>30</sup>, nous nous sommes intéressés à la pédagogie adaptative et aux méthodes personnalisées d'apprentissage. L'idée principale sous-jacente à ce travail est d'exploiter les possibilités offertes par les TIC pour construire un système d'enseignement adaptatif.

Notre objectif est de combiner les médias électroniques et les stratégies d'enseignement selon les styles d'apprentissage de chaque étudiant. Ceci soulève plusieurs questions : Comment prendre en compte le profil d'apprentissage de l'apprenant pour développer une pédagogie adaptée ? Comment utiliser et tirer profit des nouveaux e-média et des outils disponibles basés sur les TIC ? Comment combiner dans un cours ou dans un système d'apprentissage une pédagogie adaptée avec les e-média adéquats ?

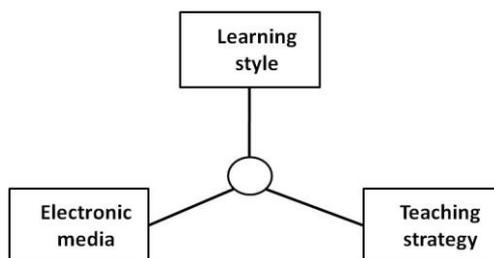


Fig. V-2. Intégration des stratégies d'enseignement avec les styles d'apprentissage et les e-média

L'idée principale de ce travail est de combiner selon un modèle tripartite les e-médias avec les stratégies d'enseignement et les styles d'apprentissage (cf. Fig. V-2). Pour chaque style d'apprentissage, on associe une ou plusieurs stratégie(s) d'enseignement d'une part, et d'autre part, un ou plusieurs média(s) électronique(s). Le résultat est une grille qui intègre les trois éléments. La démarche adoptée pour réaliser cette intégration est la suivante : après avoir sélectionné un style d'apprentissage et recensé les stratégies d'enseignement et les e-média, on fait appel à un groupe d'expert dans un atelier de travail pour construire collectivement une grille et la valider. Ensuite, cette grille est incorporée dans un système expérimental d'enseignement en ligne et tester sur un échantillon d'apprenants à l'institut ITAM au Mexique dans le cadre d'enseignements de programmation informatique.

#### V.3.1 Le modèle de Felder & Silverman

Pour cette recherche, nous utilisons le modèle des styles d'apprentissage de Felder & Silverman [75]. Les raisons de ce choix sont un peu d'ordre personnel et liées à la familiarité avec ce modèle de la personne principale qui a mené cette recherche. Le modèle de F&S comporte quatre dimensions (cf. Tableau V-1) :

- Dimension « Perception » : le type d'information que l'apprenant préfère pour percevoir et comprendre un sujet, sensoriel (p.ex. sons, sensations physiques) ou intuitif (p.ex. idées, intuitions, etc.).
- Dimension « Input » : le canal d'information que l'apprenant considère comme le plus efficace pour assimiler une information, visuel (p.ex. photos, diagrammes, graphiques, démonstrations) ou verbales (p.ex. mots, sons).
- Dimension « Processing » : comment l'apprenant préfère traiter l'information, activement (par l'engagement dans une activité physique ou une discussion) ou par l'intermédiaire d'une réflexion (p.ex. introspection, quizz, etc.).
- Dimension « Understanding » : comment l'apprenant progresse vers la compréhension, séquentiellement (étape par étape) ou globalement (de manière globale).

<sup>30</sup> ITAM : *Instituto Tecnológico Autónomo de México*

Dimension	Valeur
Perception	Sensory – Intuitive
Input	Visual – Verbal
Processing	Active – Reflective
Understanding	Sequential – Global

Tableau V-1. Les quatre dimensions du style d'apprentissage de Felder &amp; Silverman ([75])

Un instrument de mesure<sup>31</sup> sous forme d'un questionnaire comprenant 41 questions a été développé par Solomon et Felder et disponible librement sur internet. La validité de cet instrument a été évaluée statistiquement par le concepteur du modèle lui-même [76], ainsi que par d'autres chercheurs (p.ex. [180]).

### V.3.2 Proposition d'une grille d'adaptation des contenus multimédia aux styles d'apprentissage

Une stratégie d'enseignement désigne une approche pédagogique pour transmettre des informations et des connaissances. Les stratégies d'enseignement doivent être conçues de manière à ce que les apprenants soient encouragés à observer, analyser, exprimer une opinion, formuler une hypothèse, trouver une solution et découvrir la connaissance par eux-mêmes. Une stratégie d'enseignement fait référence à une séquence organisée et systématisée des activités et des ressources que l'enseignant met en œuvre. En se basant sur notre propre expérience de l'enseignement, et en consultant différents manuels de pédagogie, nous avons identifié neuf stratégies d'enseignement, elles sont présentées dans le tableau Tableau V-2.

Code	Nom stratégie
TS1	Games and simulations
TS2	Learning based on problem solving
TS3	Role playing
TS4	Presentation
TS5	Discussion panel
TS6	Brainstorming
TS7	Case study
TS8	Question and answer method
TS9	Project design method

Tableau V-2. Les neuf stratégies d'enseignement identifiées dans la littérature ([82], p.104)

Pour ce qui est des e-média, nous avons analysé la documentation disponible sur internet dans les domaines de l'enseignement en ligne et du Web 2.0. Le résultat est un ensemble de 29 supports multimédias regroupés en huit catégories (cf. Tableau V-3).

Pour construire la grille d'intégration, nous adoptons la méthode de recherche qualitative **Delphi**. Delphi est une démarche structurée de communication de groupe pour solliciter des avis d'experts sur des problèmes complexes ou pour générer de nouvelles idées, à l'aide d'une série de questionnaires et une forme de rétroaction contrôlée [148] ; on fait appel à cette méthode dans certaines recherches en SI où des opinions divergentes existent sur des questions précises (p.ex. [207]). La méthode Delphi est particulièrement adaptée pour la production de nouvelles connaissances théoriques, selon une démarche exploratoire (cf. chapitre II), plutôt que de tester ou confirmer des théories existantes [109].

<sup>31</sup> Index of Learning Styles Questionnaire : <http://www.engr.ncsu.edu/learningstyles/ilsweb.html>

Catégorie	Média
Audio media	Audio recording; Audio conference; Oral lecture
Read media	Digital magazines; Digital newspapers; eBooks; Hypertext (web pages); Readings; Slideshows
Video media	Podcast; Recorded live events; Videos; Web seminars
Diagram media	Animations; Graphics; Movies; Pictures; Simulations
Search media	Internet search
Collaboration media	Forums; Online learning communities; Webblog or blog; Wikis
Communication media	Chat (Messenger); E-mail
Tutoring media	Student Response System; Tutorial systems; WebQuest; Course Legacy System

Tableau V-3. Liste des 29 e-média recensés dans la littérature et sur le web ([82], p. 46–49)

Pour mener l'étude Delphi, nous avons d'abord constitué un panel d'experts pour participer à l'exercice. Notre choix s'est porté sur 10 enseignants de l'ITAM ayant une expérience dans l'enseignement de l'informatique, et 10 autres appartenant à différentes universités avec des expériences dans l'e-learning et l'enseignement de l'informatique. Nous avons ensuite développé un questionnaire électronique, testé avec un petit groupe d'enseignants volontaires à l'ITAM (autres que ceux du panel Delphi). Le test a permis de corriger certaines ambiguïtés et imprécisions. Le questionnaire est ensuite transmis électroniquement aux panélistes. Les réponses de cette première itération sont analysées et commentées et sont envoyées aux panélistes à nouveau pour une deuxième itération du questionnaire. Les résultats sont ensuite compilés et synthétisés dans un rapport et sont discutés dans une réunion plénière avec les panélistes. Cette réunion fut organisée au cours de la conférence "III Congreso de Estilos de Aprendizaje" à Cáceres, Espagne, à laquelle six panélistes et l'auteur principal de cette étude ont participé [publi#55].

Les résultats du panel Delphi sont une évaluation quantitative de l'adéquation entre les styles d'apprentissage et les stratégies d'enseignement d'une part (cf. Fig. V-3), et d'autre part, entre les styles d'apprentissage et les e-médias (cf. Tableau V-3 **Erreur ! Source du renvoi introuvable.**). Ces résultats sont ensuite compilés et rassemblés dans trois tableaux :

- le premier tableau indique quelle stratégie d'enseignement est jugée adéquate pour chaque style d'apprentissage ([82], p.110),
- le second indique quel e-media est jugé pertinent pour chaque style d'apprentissage ([82], p.111),
- et un troisième tableau de synthèse comportant des indications et recommandations pour la mise en œuvre, pour chaque style d'apprentissage et selon les deux tableaux précédents, les stratégies d'enseignement et les e-médias correspondants ([82], p.112–115).

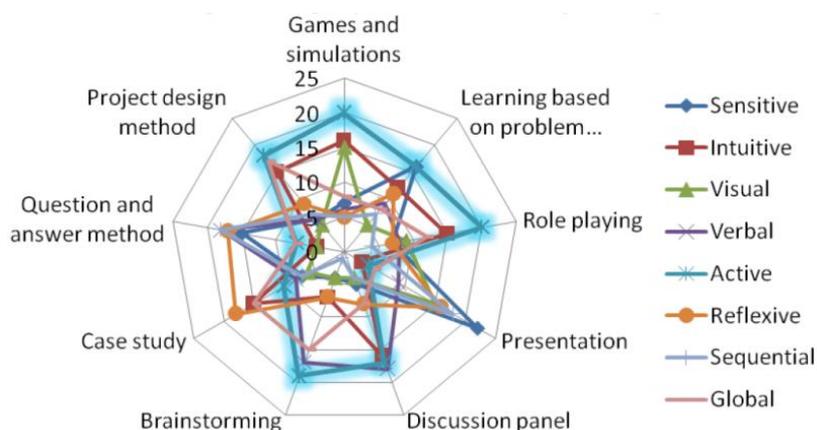


Fig. V-3. Correspondance entre les styles d'apprentissage et les stratégies d'enseignement ([82], p.108)

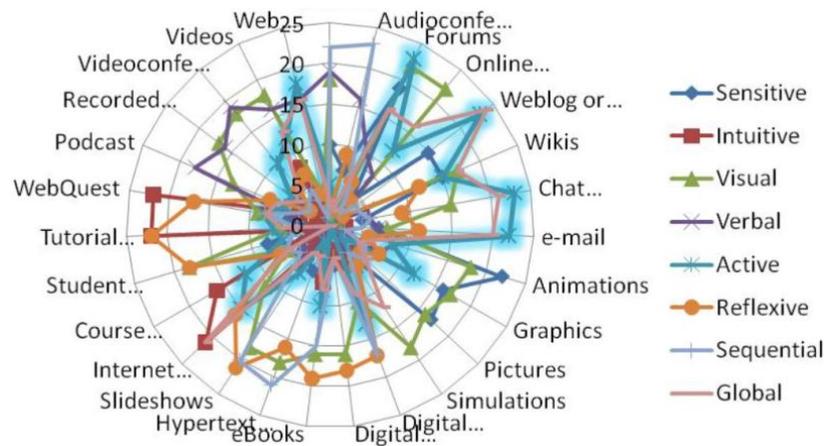


Fig. V-4. Correspondance entre les styles d'apprentissage et les médias électroniques ([82], p.109)

### V.3.3 Evaluation empirique de la grille d'adaptation

Pour expérimenter cette approche, un système expérimental pour l'enseignement en ligne de l'informatique a été développé à l'ITAM. A l'aide de ce système, trois cours différents avec des populations différentes, ont été réalisés. Ces cours portaient sur des thèmes standards en informatique, et concernaient un total de 726 apprenants (cf. Tableau V-4). Pour ce qui est du style d'apprentissage, il était mesuré à l'aide du questionnaire en ligne ; chaque apprenant avait l'obligation de le faire et de rapporter le résultat.

Cours	Taille population
Algorithme et programmation	140
Outils de calcul et algorithmes	499
Introduction aux ordinateurs et à l'informatique	87
TOTAL	726

Tableau V-4. Aperçu de l'échantillon utilisé dans l'évaluation empirique

Dans ce système, de multiples supports étaient présentés aux apprenants, sans toutefois couvrir l'ensemble des e-média que nous avons recensés. De même, plusieurs stratégies d'enseignement étaient utilisées durant le cours. Les apprenants étaient ensuite invités à répondre à des questionnaires pour évaluer leurs préférences en fonction des e-médias et des stratégies d'enseignement. Les résultats obtenus, en termes de préférence des apprenants selon le style d'apprentissage, tendent à confirmer globalement les correspondances définies dans la grille d'adaptation.

Concernant les résultats scolaires, une corrélation significative a été constatée pour le style Visual – Verbal, c.-à-d. que les résultats des apprenants appartenant à ce style avaient des résultats scolaires meilleurs que les autres, et que cette différence était statistiquement significative. Elle permet d'émettre l'hypothèse que les enseignements de l'informatique étaient plus adaptés à ce style.

### V.3.4 Conclusion

Ce travail de nature fortement empirique aborde des questions difficiles. La grille d'adaptation en est sa principale contribution, la publication de cette grille a attiré un grand nombre de citations en quelques années (plus de 185 citations). L'évaluation expérimentale a permis de démontrer sa pertinence, mais elle n'est pas allée suffisamment loin pour en confirmer la validité totale. De même, le principe de l'enseignement en ligne adaptatif n'a été que très partiellement implémenté (adaptation au style Visual – Verbal uniquement), et n'a pas permis de démontrer ni la faisabilité ni la pertinence d'un système d'enseignement adaptable basé sur les styles d'apprentissage de F&S.

## V.4 Impact des styles d'apprentissage dans l'adoption de l'apprentissage mobile

Ce travail a été réalisé dans le cadre de la thèse de Yaneli Cruz, enseignante à l'ITAM, et que j'ai co-encadrée à 50%. Cette recherche porte sur ce qu'on appelle le « Mobile Learning » (ML), c.-à-d. l'usage des technologies mobile pour supporter des activités d'enseignement et d'apprentissage. Elle se propose d'étudier la problématique de l'acceptation et l'adoption du ML auprès de population d'apprenants universitaires, et surtout, d'explorer l'effet potentiel des styles d'apprentissage sur ce processus d'adoption. De plus, cette recherche adopte une vision de l'adoption selon les modèles théoriques les plus récents, qui tentent d'expliquer non seulement l'intention d'usage, mais aussi la continuité d'usage, c.-à-d. la confirmation du sujet de son intention initiale d'utiliser l'artéfact. La principale question de recherche est la suivante : Y a-t-il un effet significatif des styles d'apprentissage sur l'intention d'usage et la continuité d'utilisation de l'apprentissage mobile ?

### V.4.1 Le Mobile Learning

Grâce aux développements technologiques des dix dernières années, le Mobile Learning (ML) a émergé comme une forme d'apprentissage amélioré qui permet aux apprenants, avec une grande facilité, de communiquer, d'interagir, et d'accéder à des supports électroniques, à tout moment et dans pratiquement n'importe quel endroit, y compris en classe [218]. Le ML est considéré comme une forme particulière de l'enseignement en ligne (e-learning), son développement est du en grande partie aux principales caractéristiques des appareils mobiles telles que la portabilité et la disponibilité.

Ainsi, des établissements d'enseignement supérieur développent de applications mobiles pour encourager la participation en classe, pour donner accès à des ressources multimédias (p.ex. des conférences enregistrées sous forme de Podcast), ou encore à des fins administratives en donnant accès aux calendriers et notes de la classe [157]. Certains enseignants explorent l'usage des réseaux sociaux et des téléphones mobiles pour instaurer une communication et une interaction instantanées dans les grands amphis du premier cycle universitaire [27]. D'autres universités ont lancé des projets mobiles pour enseigner une discipline de manière innovante comme à l'Université de Yale, où à l'aide d'appareils numériques mobiles et un réseaux sans fil, les étudiants analysent les images d'un microscope numérique projetées sur un écran à l'entrée de la salle de classe et diffusées en direct sur leurs iPads [163]. De plus, certains établissements commencent à exiger des étudiants l'achat d'un appareil mobile, tandis que d'autres les subventionnent [37]. Enfin, dans le contexte des pays en voie de développement, le ML peut s'avérer particulièrement pertinent pour permettre à des populations rurales et ayant peu d'accès aux moyens éducatifs, de suivre des formations en ligne ou des cours universitaires [91].

La prémisse de l'adoption du ML dans les milieux éducatifs est que cela devrait améliorer le processus d'apprentissage. Dans la mesure où le ML implique une nouvelle configuration de l'enseignement et de l'apprentissage, certains facteurs peuvent compromettre son efficacité. Par conséquent, plusieurs études ont été réalisées à propos du ML pour identifier les facteurs qui influent sur son acceptation [149]. Cependant, il y a un manque de compréhension des facteurs qui motivent les apprenants à accepter et à utiliser les appareils mobiles pour l'apprentissage. Dans le cadre de cette recherche, notre focus est sur les différences individuelles en général, et les styles d'apprentissage en particulier. A notre connaissance, aucune étude n'a abordé ces questions sous cet angle là.

### V.4.2 Les modèles d'acceptation de la technologie

Quel modèle peut expliquer l'acceptation et l'adoption du ML ? L'acceptation et l'adoption des artéfacts informatiques et sociotechniques sont des questionnements issus du champ des SI. Le modèle TAM, introduit en 1989 [55], a eu un impact considérable grâce à sa puissance explicative et sa facilité de compréhension. Développé à partir de théories plus générales en psychologie sur l'action raisonnée des individus, le modèle TAM postule que l'intention d'accepter un artéfact va dépendre de deux facteurs : **l'utilité** et la **facilité d'usage**. Par la suite, une version plus élaborée a été proposée pour prendre en comptes divers facteurs contextuels susceptibles d'influer sur l'acceptation. C'est le modèle de la *Théorie Unifiée d'Acceptation et d'Utilisation de la Technologie* (UTAUT) [228].

Plusieurs travaux ont utilisé le modèle TAM pour expliquer l'acceptation de l'enseignement en ligne (p.ex. [191], [176]), et aussi pour l'acceptation du ML [150]. Quant au modèle UTAUT, il a été utilisé pour comprendre l'acceptation des terminaux mobiles (p.ex. [36]).

Cependant, même si l'utilisateur perçoit positivement un artéfact et confirme son intention de l'utiliser, ses attentes et ses appréciations sont susceptibles d'évoluer par la suite. Les modèles TAM et UTAUT ne fournissent pas des construits pour **capturer cette évolution** et prendre en compte la confirmation (ou non) des intentions initiales de l'utilisateur ; c'est ce qu'on appelle la « post-adoption ». A contrario de la recherche sur l'adoption de la technologie qui est très mature, la post-adoption a reçu moins d'attention. Une piste de recherche a été de faire appel à l'*Expectation Disconfirmation Theory* (EDT) qui explique l'origine de la satisfaction et de l'insatisfaction des consommateurs au rachat de produits et au contexte de rétention de service [166]. L'EDT a été intégrée par Battacherjee et Premkumar (2004) dans un modèle théorique en deux étapes qui prend en compte les changements sur la conviction cognitive, c.-à-d. l'utilité perçue et l'infirmité (non-confirmation), et sur l'affect, c.-à-d. l'attitude et la satisfaction ; les niveaux d'infirmité ayant le potentiel d'influer sur les comportements successifs [19]. De même, Venkatesh (2011) a étendu le modèle UTAUT avec un modèle prolongé en deux phases pour mieux décrire et expliquer la continuité et la discontinuité d'usage des SI dans le contexte des systèmes transactionnels [229]. Ce ne sont que quelques exemples sur la façon dont différents projets de recherche ont émergé pour comprendre les comportements de post-acceptation des innovations dans la technologie.

Cette recherche se propose d'étudier le ML depuis deux perspectives : celle de l'acceptation de la technologie avec prise en compte de facteurs de pré-acceptation, et celle de la continuité d'utilisation avec prise en compte de facteurs de post-adoption. Ensemble, l'intégration des deux perspectives dans un seul modèle à deux étapes pourrait enrichir les mesures de l'adoption initiale et de l'utilisation continue du ML.

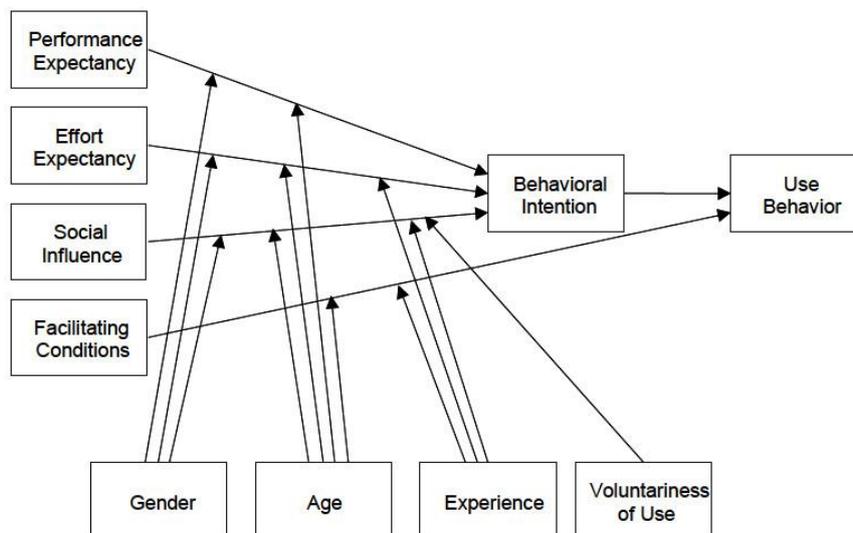


Fig. V-5. Le modèle théorique UTAUT sur lequel s'est basée cette recherche (extrait de [228], p.477)

Le modèle théorique qui va être utilisé dans cette recherche est basé sur le modèle UTAUT (cf. Fig. V-5). Le modèle UTAUT considère que le genre, l'âge, l'expérience ultérieure dans l'artéfact considéré, ainsi que la motivation intrinsèque de l'utilisateur vont modérer les facteurs qui déterminent l'intention d'usage (p.ex. la performance attendue). Le modèle que nous avons dérivé de UTAUT sera présenté dans une prochaine section de ce chapitre (cf. §V.4.4).

### V.4.3 Le modèle des styles d'apprentissage de Kolb

L'autre élément de ce travail de recherche est la prise en compte du style d'apprentissage de l'apprenant. Cet élément instaure une continuité entre le travail de thèse d'A.-L. Franzoni présenté dans la section précédente et le travail développé ici. Cependant, nous avons choisi un autre modèle des styles d'apprentissage, celui de Kolb [130]. La raison principale derrière ce choix est le constat que le modèle de F&S utilisé dans le travail d'A.-L. Franzoni est trop com-

plexe avec ses 16 combinaisons possibles, et qu'il n'a pas de fondement théorique clair. Nous avons constaté par ailleurs que le modèle de Kolb est très utilisé dans diverses recherches dans les domaines des sciences de l'éducation (p.ex. [151]). Enfin, le modèle de Kolb fait partie de la sélection des 13 modèles choisis dans l'étude extensive des styles d'apprentissage de Coffield et al. en 2004 (alors que le modèle de F&S n'en fait pas partie) [44].

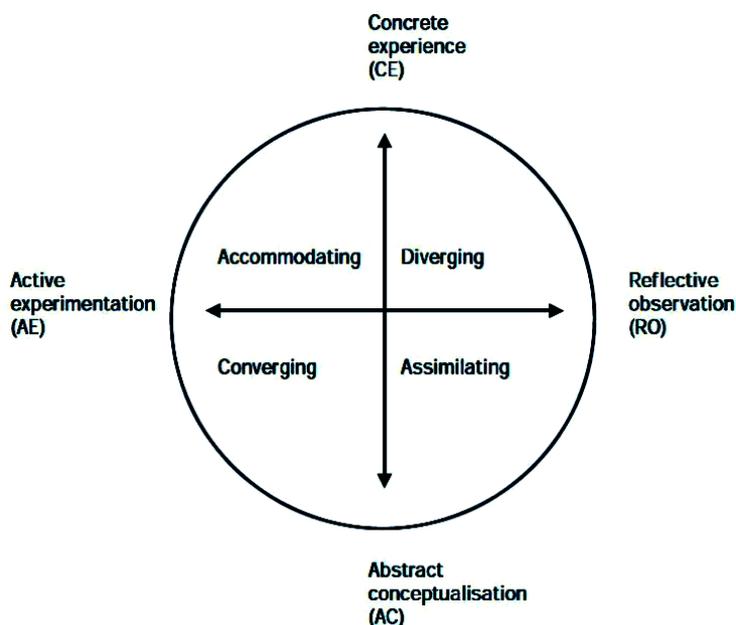


Fig. V-6. Le modèle descriptif des styles d'apprentissage de Kolb (d'après [44], p.62).

D'après Kolb, l'apprentissage nécessite la résolution de conflits entre des modes d'adaptation au monde qui sont *dialectiquement* opposés. L'apprentissage est, par sa nature même, une activité pleine de *tension*, parce qu'une nouvelle connaissance est construite par les apprenants en faisant le choix de capacités particulières dont ils ont disposé. Un apprenant efficace a besoin de quatre types de capacité à apprendre (cf. Fig. V-6) :

- Capacité à apprendre à partir d'expériences concrètes (CE);
- Capacité à apprendre à partir d'observations réflexives (RO);
- Capacité à apprendre à partir de conceptualisations abstraites (AC);
- Capacité à apprendre à partir d'expérimentations actives (AE).

Ces quatre capacités sont les pôles de deux axes indépendants comme le montre la Fig. V-6, avec *le concret versus l'abstrait* d'un côté (axe vertical) et *l'action versus la réflexion* de l'autre (axe horizontal). Les conflits sont résolus par le choix de l'un de ces modes d'adaptation, et au fil du temps, l'individu développe des moyens privilégiés de choisir. Cette typologie définit ainsi quatre styles d'apprentissage :

- Le style « Converging » est abstrait et actif; il est bon à la résolution de problèmes, à la prise de décision et l'application pratique des idées, et préfère traiter des problèmes techniques plutôt que des problèmes interpersonnels.
- Le style « Diverging » met l'accent sur l'expérience concrète et l'observation réfléchie, il est imaginaire et préfère l'observation plutôt que l'action, il est intéressé par les autres et par les rapports sociaux.
- Le style « Assimilating » préfère la conceptualisation abstraite et l'observation réfléchie, il préfère les raisonnements inductifs et la création de modèles théoriques, plus porté sur le côté logique des idées que leur côté pratique.
- Le style « Accomodating » préfère l'action et l'expérience concrète, il préfère réaliser des plans et se lancer dans de nouvelles expériences, avec une tendance à résoudre les problèmes de manière intuitive.

Au delà de cette typologie qui peut sembler rigide, Kolb insiste sur le fait que « les styles d'apprentissage représentent des préférences pour un mode d'adaptation par rapport à d'autres; mais ces préférences ne sont pas toujours utilisées à l'exclusion des autres, et peuvent varier dans le temps et d'une situation à une autre » ([130], p.290).

#### V.4.4 Proposition de nouveaux modèles théoriques

Pour comprendre l'adoption et la continuation d'usage du ML, nous avons construit des modèles et nous les avons testés empiriquement avec plusieurs échantillons d'utilisateurs. Les modèles proposés sont construits à partir du modèle UTAUT et s'inspirent de travaux similaires ayant les mêmes objectifs (c.-à-d. comprendre la continuité d'usage).

La Fig. V-7 présente le premier modèle qui porte uniquement sur la phase d'acceptation. Ce modèle considère les trois variables initiales du modèle UTAUT (performance attendue, effort attendu, et influence sociale), et introduit deux variables modératrices, l'expérience antérieure et le style d'apprentissage.

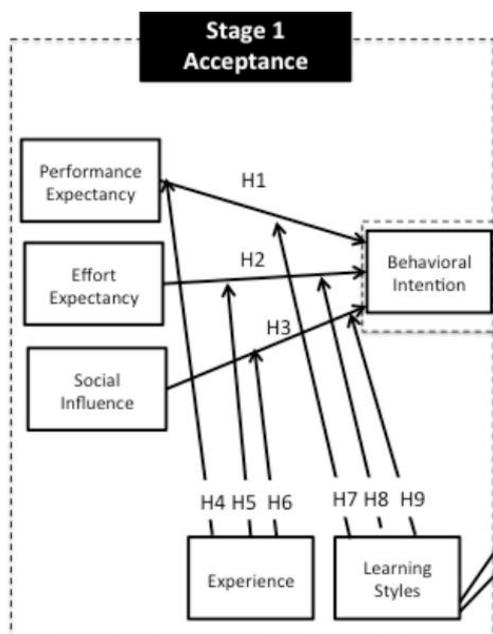


Fig. V-7. Le premier modèle théorique proposé pour la compréhension du ML (1ère phase)

Le second modèle est un modèle intégré qui porte sur les deux phases d'acceptation et d'usage actuel (c.-à-d. la poste acceptation). Il a été élaboré après avoir testé le premier (cf. Fig. V-7), et de ce fait, il n'inclut plus la variable modératrice « expérience antérieure » car celle-ci s'est révélée sans effet lors du test. On distingue dans ce modèle la distinction claire entre les deux phases de pré-acceptation et de poste acceptation (cf. Fig. V-8). Ce modèle fait de nouveau l'hypothèse que le style d'apprentissage joue le rôle de variable modératrice dans les deux phases.

#### V.4.5 Enquête quantitative – phase 1

La population cible était les étudiants de premier cycle inscrits dans une spécialisation de gestion des affaires. L'étude a été menée à l'ITAM dans le cadre d'une expérience avec des terminaux mobiles dans deux cours de comptabilité pour le semestre d'automne 2013. De multiples types de terminaux mobiles étaient utilisées tels que tablettes, Smartphones et ordinateurs portables. Nous avons invité 80 étudiants à participer à cette étude, et nous avons reçu 42 réponses au questionnaire des styles d'apprentissage et un total de 39 réponses au questionnaire concernant l'acceptation de la technologie. L'étude s'est donc basée sur l'échantillon de 39 réponses aux deux questionnaires. Parmi les répondants, 38% étaient des hommes et 62% des femmes. En moyenne, les participants ont 21 ans et utilisent leur téléphone 74 minutes par jour pour les appels et les SMS, 59 minutes sur les applications mobiles, 20 minutes sur le courrier électronique, 20 minutes sur les jeux vidéos et 69 minutes sur les plateformes de réseaux sociaux. Au début du semestre, les étudiants ont été informés du projet de recherche et de la nouvelle activité d'apprentissage basée sur les dispositifs mobiles. On a demandé aux étudiants de signer une charte de confidentialité mobile pour participer à l'étude, cette politique permet l'utilisation d'appareils mobiles dans la salle de classe à des fins académiques.

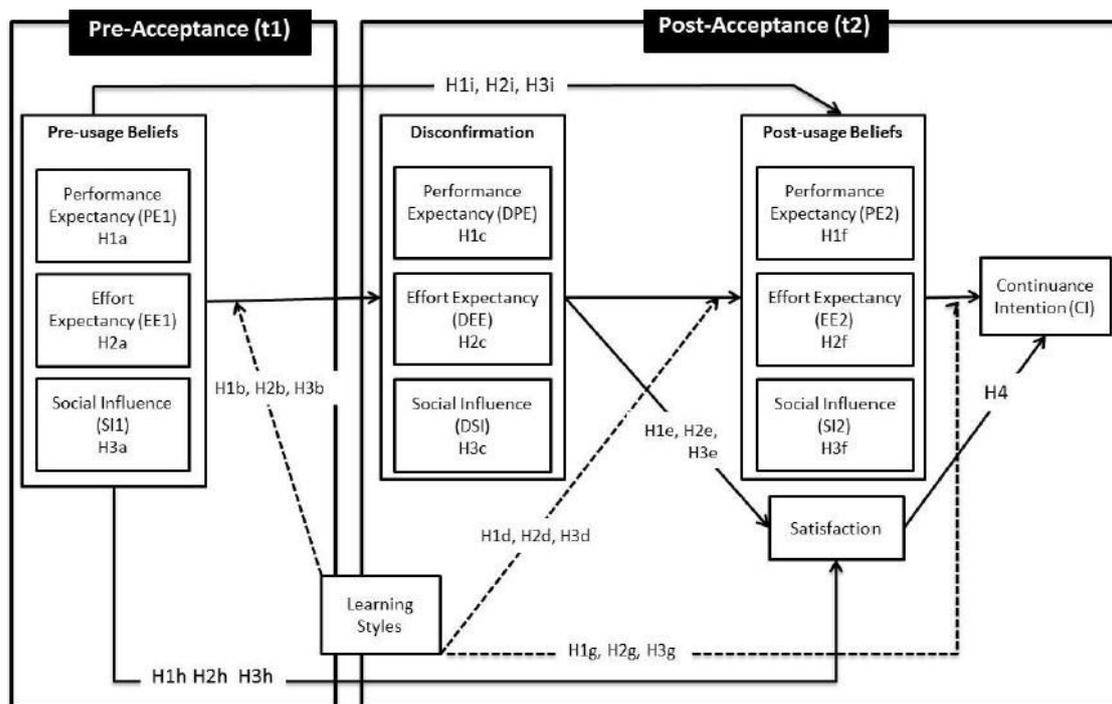


Fig. V-8. Le second modèle théorique proposé pour la compréhension du ML (2<sup>nd</sup>e phase)

#### V.4.6 Enquête quantitative – phase 2

Cette deuxième étude a été réalisée dans des conditions similaires à la première. D'autres outils logiciels ont été utilisés cette fois-ci, les étudiants ont dû télécharger les applications requises, c.-à-d. Google Docs pour charger les rubriques créées sur Google Forms, Exittix App<sup>32</sup> utilisé comme système de réponse des élèves, et Nearpod App<sup>33</sup> utilisé pour collaborer à l'aide de vidéos et de présentations animées. Les activités d'apprentissage comprenaient des quiz, des vidéos courtes et des rubriques pour faire des commentaires et des retours d'expériences à d'autres étudiants à travers les dispositifs mobiles.

Conformément aux protocoles appliqués par Bhattacharjee et Premkumar dans [19] et par Venkatesh dans [229], nous avons mesuré les variables du modèle à deux points dans le temps. L'étude a été organisée en deux phases: pré-acceptation dans la phase t1 et post-acceptation dans la phase t2. Dans la phase t1, deux questionnaires ont été appliqués: l'un pour explorer les perceptions et le second pour reconnaître le style d'apprentissage selon le modèle de Kolb<sup>34</sup>. Au milieu du semestre, un autre questionnaire a été appliqué pour recueillir des données après l'expérience avec l'enseignement mobile. L'instrument de pré-acceptation recueillait des informations descriptives et des éléments nécessaires pour mesurer les trois variables *performance attendue*, *effort attendu*, et *influence sociale*.

Un mois après avoir répondu aux questionnaires de la phase t1, les répondants ont été invités à participer à une enquête de suivi pour indiquer leur perception post-acceptation et le maintien de leur intention d'utiliser le ML. Dans le questionnaire de la phase t2, nous avons mesuré les convictions concernant la non confirmation, la satisfaction, les convictions post-acceptation, et l'intention de continuité d'usage du ML.

Nous avons examiné un échantillon de 90 étudiants, et un total de 51 réponses pour les deux étapes ont été retournées, y compris le questionnaire du style d'apprentissage. L'étude a donc exploité l'échantillon de 51 réponses. Les participants étaient âgés de 21 ans en moyenne et passent au téléphone 98 minutes par jour pour les appels et les SMS, 55 minutes avec des applications mobiles, 20 minutes sur le courrier électronique, 20 minutes sur les jeux vidéo et 66 minutes sur les plates-formes de réseaux sociaux ; 60% des élèves possèdent plus de 3 appareils mobiles.

<sup>32</sup> <http://exittix.com/>

<sup>33</sup> <https://nearpod.com/>

<sup>34</sup> Instrument de mesure pour le modèle de Kolb : <http://learningfromexperience.com/tools/>

### V.4.7 Analyse des données et modélisation PLS

L'analyse des données a été effectuée avec la méthode PLS (« Partial Least Squares »), ou *régression des moindres carrées partielles*. C'est une méthode d'analyse statistique et de recherche de corrélation qui emprunte sa démarche à la fois à l'analyse en composantes principales et à la régression linéaire. Cette méthode très sophistiquée est capable d'identifier des variables latentes dans un modèle, et elle a la particularité de pouvoir s'appliquer sur des problèmes avec beaucoup de variables et des échantillons de données de taille modeste.

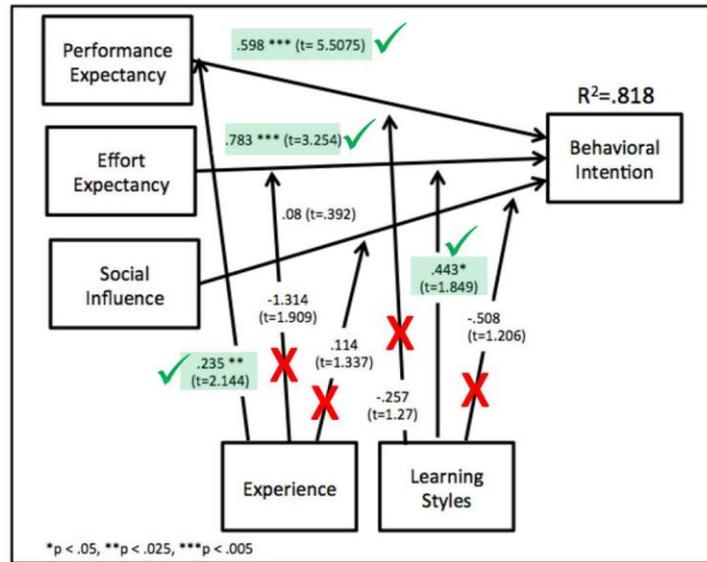


Fig. V-9. Modèle structurel pour la phase 1

La Fig. V-9 présente les résultats obtenus pour la phase 1. Comme on peut le voir, les relations initiales du modèle UTAUT sont confirmées. Ainsi, l'intention d'utilisation le ML est bien impacté par la performance et l'effort attendus, et par l'influence sociale. Par contre, l'expérience antérieure est modératrice uniquement de la relation PE -> BI ; de même le style d'apprentissage de l'apprenant ne modère que la relation EE -> BI. On peut expliquer cette modération par le fait que le style d'apprentissage de Kolb capture une manière d'agir et de réfléchir tournées vers l'action ou la réflexion, ce qui peut donner lieu à des perceptions différentes de l'effort à faire pour apprendre à travers un dispositif de type ML.

La **Erreur ! Source du renvoi introuvable.** présente les résultats pour la phase 2 de l'étude, celle où l'on cherche à comprendre non seulement l'adoption, mais aussi la continuité d'usage. Ce modèle montre uniquement les liens de corrélation qui ont été validés, c.-à-d. qui satisfont la contrainte statistique de signifiante ( $p < 0.05$ ). Ce modèle est la version générale, il ne prend pas en compte la modération potentielle du style d'apprentissage. Comme on peut le voir, la majorité des liens n'ont pas été validés. Les trois variables initiales du modèle UTAUT restent corrélées entre elles entre les deux phases de l'étude. Néanmoins, la seule qui influe sur la variable finale de continuité d'usage est encore une fois la performance attendue (lien PE2 -> CI). Quant à l'explication de la variation ( $R^2$ ), elle est moyenne (58%).

Bref, ce que ces deux modèles indiquent est que l'intention d'adopter le ML est influencée par les trois facteurs du modèle, alors que pour continuer à utiliser le ML, c'est uniquement la performance attendue, c.-à-d. l'utilité. Ce résultat n'a rien de nouveau en soi, il est néanmoins conforme à l'intuition et à la pratique des systèmes.

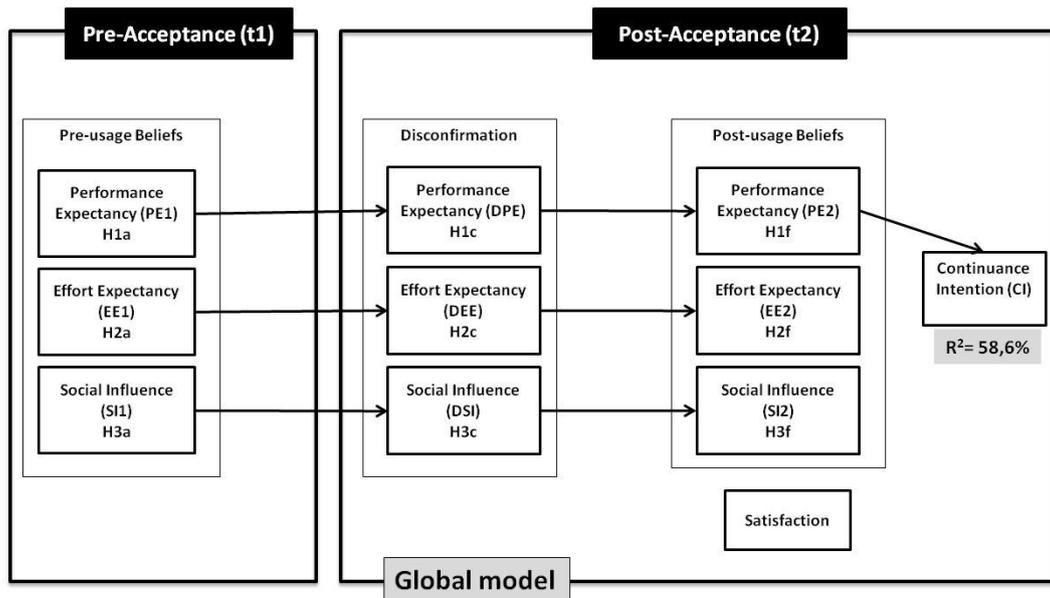


Fig. V-10. Le modèle obtenu après analyse PLS, cas général (uniquement les liens significatifs sont montrés)

Pour analyser l'impact potentiel du style d'apprentissage, nous avons considéré la paire de style AE/RO (cf. Fig. V-6). Le résultat apparaît dans la figure Fig. V-11. Les liens initiaux du modèle général (cf. figure précédente) restent bien entendu valides ; néanmoins, on constate que le style d'apprentissage modère le lien issu de l'influence sociale, et aussi celui de l'effort attendu. Par ailleurs, l'effort attendu devient un facteur déterminant pour la continuité d'usage. Enfin, l'explication de la variation est nettement plus élevée avec des valeurs de 71% pour le style AE et 67% pour le style RO.

Que peut-on conclure de ce second modèle ? Le style d'apprentissage semble bien avoir une certaine influence sur les relations entre les variables du modèle, et que l'effort attendu de l'utilisation du ML semble être perçu différemment par les individus qui, selon le modèle de Kolb, sont actifs et tournés vers l'expérimentation par rapport au reste de la population.

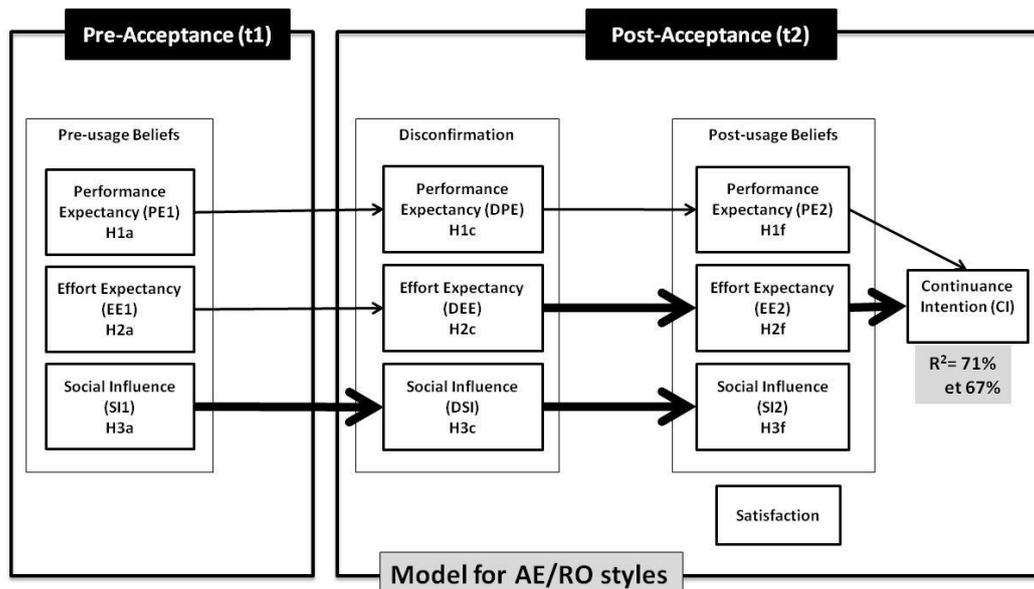


Fig. V-11. Le modèle obtenu après analyse PLS, cas des styles AE/RO (les liens en gras indique une modération positive)

## V.5 Conclusion et perspectives de recherche

Ce chapitre a été consacré à un volet important de mes travaux, dans un domaine que je pratique quotidiennement en tant qu'enseignant, mais que je connaissais très peu en termes de recherche. Les problématiques de pédagogie et d'enseignement en ligne sont des domaines de recherche très actifs, et j'étais étonné de découvrir une communauté très dynamique avec une discipline très proche de la psychologie. Néanmoins, je constate que les recherches sur l'enseignement en ligne, de par **l'importance de l'artéfact technologique**, semblent être dominées par des chercheurs au profil informatique. Et certaines revues de référence, p.ex. *Computer and Education*, ont une connotation informatique forte.

J'ai découvert la théorie des styles d'apprentissage avec la thèse d'A.-L. Franzoni qui était passionnée par le sujet et souhaitait l'inclure dans sa problématique de recherche. C'est un peu plus tard que j'ai pris la mesure de la complexité des questions qui y sont attachés, des **divergences d'opinion** à leur propos, et qu'en fin de compte, il est très difficile de catégoriser clairement les capacités cognitives et psychologiques des individus. Comme le souligne à juste titre le rapport de Coffield et al., il ne faut pas trop attendre d'un instrument de mesure assez simple et basé uniquement sur le *self-reporting* ([44], p.141). Néanmoins, l'étude empirique sur le Mobile Learning montre que les styles d'apprentissage sont susceptibles d'influer sur la manière avec laquelle un individu appréhende un nouveau dispositif d'enseignement.

Je dois signaler par ailleurs que la question de la **validité de ces théories** (sur les styles d'apprentissage) ne rentre pas dans le champ des recherches menées ici. On a pris ces théories telles qu'elles, en essayant d'en savoir le maximum sur le pour et le contre, et en considérant qu'elles constituent un socle théorique suffisamment pertinent et intéressant pour le considérer comme une base pour nos recherches.

D'un point de vue méthodologique, ces deux travaux m'ont permis de m'initier aux méthodes de recherche en sciences des faits et **l'utilisation des théories explicatives** pour mener la recherche (cf. chapitre II). Le rôle des mathématiques statistiques est essentiel, et la validité des résultats en dépend en grande partie. Le rôle des modèles théoriques, tel que le modèle UTAUT, et comment les exploiter pour construire un montage théorique susceptible d'être testé expérimentalement, est une autre facette de cette découverte et de cet apprentissage.

Dans l'immédiat, un sujet important que je compte développer est la synthèse de résultats quantitatifs à l'aide d'une technique de méta-analyse. Ce sujet est lié à la réflexion que j'ai menée sur les méthodes de revue de la littérature (cf. Chapitre 2, §II.6.5). Pour la revue de la littérature de la thèse de Y. Cruz, on a sélectionné 38 articles qui étudient **l'impact** des styles d'apprentissage sur l'enseignement en ligne ; des études assez différentes en termes de style d'apprentissage, et aussi très variables en termes de taille d'échantillon. Comment tirer une conclusion significative et pertinente à partir de cette masse de données, à priori hétérogène ? Les techniques de méta-analyse permettent de combiner des données, ayant un niveau contrôlable d'hétérogénéité, pour en tirer des indicateurs synthétiques capables de renseigner de manière rigoureuse sur l'orientation globale des résultats sous-jacents [202].

Sur plus long terme, et pour accompagner les transformations actuelles que déclenche le numérique, je souhaite continuer à observer et à étudier le domaine de l'enseignement en ligne en tant que praticien qui peut être amené à s'interroger sur le choix des pédagogies les plus pertinentes, sur la place du jeu et de la simulation en tant que pédagogies innovantes, et sur l'impact des MOOC sur le système éducatif traditionnel.

## VI. Chapitre VI : Conclusion générale

J'ai commencé ce mémoire en indiquant que **l'analyse ontologique** de l'informatique en tant que sujet scientifique d'étude aboutit à considérer **trois perspectives** : science du formel, science de conception et science des faits. Le besoin de cette analyse a émergé au fil d'une décennie, suite aux différents travaux que j'ai menés avec des équipes de recherche appartenant à des champs disciplinaires multiples. Ces trois perspectives se retrouvent dans les travaux présentés dans le chapitre III (sciences de conception, science du formel), le chapitre IV et V (science des faits).

Suivant le même besoin, j'ai ensuite défini le système d'information (SI) en tant qu'artéfact sociotechnique, distinct de l'artéfact informatique par la prédominance des facteurs humains et organisationnels et par son fort impact sur l'environnement ; un impact qui influe et façonne les contextes d'usage (et vice-versa). Le SI amène des interrogations différentes qu'il est convenu de classer en deux catégories : celles qui relèvent des usages et celles qui relèvent de la conception et développement. **Mes travaux ont porté sur ces deux problématiques**, les chapitre III et IV pour la conception et développement, et le chapitre V pour les usages.

Néanmoins, l'analyse ontologique est insuffisante pour saisir la multidisciplinarité d'une activité ou d'un domaine de recherche. En effet, **la dimension méthodologique** est tout aussi importante, et j'ai consacré une partie significative de mes efforts de recherche ces dernières années pour mieux la comprendre et la maîtriser. Cet aspect a été longuement décrit dans le chapitre II où, après un rappel synthétique de la littérature sur le sujet et quelques brefs aperçus des questions épistémologiques sous-jacentes, j'ai présenté une série de travaux publiés ces quatre dernières années et qui relèvent – directement ou indirectement – de problématiques liées à la notion de « méthode » dans les domaines de l'ingénierie et de la science.

En introduisant dans le chapitre I cette perspective épistémologique d'étude, j'ai soulevé un certain nombre de questionnements. Le Tableau VI-1 reprend ces questionnements et tente d'y apporter quelques éléments de réponses en se basant sur ma propre expérience de recherche. Ces réponses ne doivent pas être prises comme des affirmations définitives, mais comme des réflexions personnelles sujettes à débat (cf. Tableau VI-1). Il est important de préciser que les quatre chapitres de ce mémoire **ne couvrent pas l'intégralité de mes travaux**. Plusieurs de ces travaux (dont certains ont été mentionnés dans la section §1.4 de l'introduction), correspondent à des collaborations dans l'encadrement de thèses au sein du laboratoire CRI, d'autres à des collaborations ponctuelles avec des chercheurs aux niveaux national et international.

Plusieurs **perspectives** et voies de recherche se dessinent :

- Formalisation des modèles : le travail réalisé sur l'exécution de modèles (cf. chapitre III) reste ouvert car il y a encore beaucoup à faire pour implémenter, expérimenter et valider l'approche de modélisation événementielle de la sémantique d'un langage.
- Modélisation des SI : elle reste mon thème de prédilection, et suite à l'émergence de ce qu'on appelle communément « la transformation numérique des entreprises », je m'intéresse à la modélisation des entreprises (appelée aussi « cartographie du SI » ou « Enterprise Modeling ») et à son impact sur la conduite et la maîtrise de ces processus de transformation.
- Evaluation empirique des méthodes d'ingénierie : je travaille actuellement avec un doctorant au niveau international pour définir une méthode d'ingénierie des besoins qui inclut des techniques créatives, et pour l'évaluer expérimentalement.
- Application de la méthode de méta-analyse aux résultats quantitatifs des recherches sur l'impact des styles d'apprentissage dans le contexte de l'enseignement en ligne.
- Poursuite du travail entrepris sur l'analyse des problématiques de construction de théories dans les sciences des faits, et élargissement aux questions de développement de théories dans les sciences de conception.

Questionnement	Tentative de réponse
<i>L'étude du SI et de l'artéfact logiciel sur lequel il repose relève-t-elle des sciences des faits ou des sciences de conception ?</i>	Elle relève des deux. Un artéfact logiciel amène toujours un contexte et une problématique d'usage pour lesquels des faits peuvent être constatés, récoltés et analysés.
<i>Dans quelle mesure l'étude de l'un éclaire-t-elle l'étude de l'autre ?</i>	La démarche des sciences de conception repose sur la confrontation entre la structure interne des artéfacts et les contextes d'usage ; l'une et l'autre sont indispensables au concepteur.
<i>Dans quelle mesure est-il possible d'étudier l'un (c.-à-d. le SI) en ignorant l'autre (c.-à-d. le logiciel) ?</i>	On peut effectivement considérer l'artéfact technologique comme une boîte noire, et ne prendre en compte que son apport au contexte d'usage et son impact sur celui-ci. Ce type d'approche est typique dans les recherches en <i>Management des SI</i> .
<i>Est-il nécessaire et/ou pertinent de combiner l'étude de l'un et de l'autre, et si oui, comment faire et quel est l'impact sur les méthodes de recherche ?</i>	Néanmoins, l'artéfact logiciel possède des caractéristiques spécifiques liées à son architecture et sa construction interne. Il peut donc être pertinent d'étudier l'un et l'autre, ce qui permettrait d'expliquer l'usage et l'impact par la connaissance détaillée de l'artéfact logiciel.
<i>Comment se distinguent les critères d'évaluation des contributions scientifiques lorsque l'on étudie l'un ou l'autre ?</i>	De multiples critères sont envisageables. Ce qu'on peut dire ici est que, dans le cas des sciences de conception, c'est d'une part la <b>nouveauté</b> et l' <b>originalité</b> qui priment, et d'autre part, la <b> faisabilité technique</b> et la <b>satisfaction de besoins</b> . La formalisation mathématique reste un outil de choix pour démontrer la validité conceptuelle de l'artéfact conçu. Dans le cas des sciences des faits, les critères d'évaluation examinent l' <b>apport théorique</b> et sa contribution, d'abord à la <b>description</b> , et ensuite à l' <b>explication</b> du phénomène étudié. L'usage de théories d'ordre générale est fortement encouragé ; ces théories sont souvent issues de domaines disciplinaires connexes (p.ex. gestion, sociologie, psychologie).
<i>Est-il possible de faire de l'ingénierie des SI sans faire – serait-ce un peu – de génie logiciel, et vice-versa ?</i>	L'ingénierie des SI étant une science de conception, elle reste fondamentalement très proche du génie logiciel, et plusieurs sujets et techniques sont communs.
<i>Dans quelle mesure les recherches et contributions scientifiques en ingénierie des SI sont-elles distinctes et s'évaluent-elles différemment de celles en génie logiciel, et vice-versa ?</i>	Les contributions dans les deux domaines s'évaluent de manières assez proches. On peut néanmoins dire que dans le cas des SI, l' <b>impact</b> et l' <b>apport</b> de l'artéfact acquièrent une importance plus grande. Cette distinction peut se constater en analysant les publications des deux conférences principales : CAISE et ICSE.
<i>Y aurait-il des différences significatives dans les fondements épistémologiques de la recherche en ingénierie des SI et celle en génie logiciel ?</i>	Non, je ne le pense pas et je ne l'ai pas constaté.
<i>En quoi l'étude du développement du SI peut-elle renseigner l'étude des usages (et vice-versa) ? Ces deux domaines d'étude sont-ils disjoints, ou y a-t-il des intersections et des points de vue communs qui pourraient être exploités ?</i>	Ces deux problématiques de recherche restent assez disjointes, même si on peut facilement constater un impact potentiel des choix d'implémentation d'un logiciel sur les possibilités qu'offre ce logiciel en termes d'usage et de fonctionnalités.

Tableau VI-1. Questionnements soulevés dans l'introduction de ce manuscrit

## VII. Production et rayonnement scientifique

### VII.1 Synthèse bibliométrique

Citations (Google Scholar, 13 janvier 2017)

- H-index = 11
- i10-index = 13
- Total des citations = 484

Publications (x70 – dont 4 en tant que seul auteur)

- 11 articles dans revues et magazines (dont 1 classée A, 3 classées B ou C)
- 3 ouvrages coédités
- 10 chapitres d'ouvrage (dont 1 article d'encyclopédie)
- 40 publications dans conférences et ateliers de recherche
- 6 autres publications

Co-encadrement de thèses

- 4 thèses co-encadrées à Paris 1 La Sorbonne (50% pour la 3 premières, 75% la dernière)
- 2 thèses co-encadrées à Telecom Ecole de Management (75% l'une, 50% l'autre)

Encadrement de mémoires Master

- Plus de 50 mémoires et thèses professionnels niveau M2 depuis 1998

Enregistrement en ligne

- Mon tutorial sur la Meta-modélisation est publié en ligne sur [IEEE Education](#)

### VII.2 Publications

#### VII.2.1 Revues avec comités de lecture

1. Jaramillo Franco, A., Assar, S.: **Leveraging Creativity Techniques in Requirements Elicitation**, *Requirement Engineering Magazine*, Issue 2016, no 2, 2016 [<https://re-magazine.ireb.org/issues/2016-2-take-the-broader-view/leveraging-creativity-techniques-in-requirements-elicitation-a-literature-review/>]
2. Assar, S., Borg, M., Pfahl, D.: **Using text clustering to predict defect resolution time: a conceptual replication and an evaluation of prediction accuracy**, *Empirical Software Engineering Journal*, 22(3), pp. 1437–1475, Springer, 2016, [<http://link.springer.com/article/10.1007/s10664-015-9391-7>]
3. Assar, S.: **Méthodes de recherche empirique en ingénierie des SI. Principes et applications**, *Ingénierie des Systèmes d'Information*, VOL 20/6, pp. 11–33, Lavoisier, 2015, [<http://isi.revuesonline.com/article.jsp?articleId=35599>]
4. Assar, S., Souveyet, C., Mallouli, S.: **Vers une sémantique orientée événement des modèles de processus : Démarche de type IDM**, *Ingénierie des Systèmes d'Information*, VOL 20/2, pp.93–117, Lavoisier, 2015, [<http://isi.revuesonline.com/article.jsp?articleId=20762>]
5. Mazo, R., Assar, S., Salinesi, C., Hassen, N. B.: **Using Software Product Line to improve ERP Engineering: Literature Review and Analysis**. *Latin-American Journal of Computing*, 1(1), 2014, [<http://lajc.epn.edu.ec/index.php/LAJC/article/view/56>]
6. Latif, R., Abbas, H., Assar, S.: **Distributed Denial of Service (DDoS) Attack in Cloud- Assisted Wireless Body Area Networks: A Systematic Literature Review**. *Journal of Medical Systems*, 38(11): pp. 1–10, 2014, [<http://link.springer.com/article/10.1007/s10916-014-0128-8>]
7. Franzoni, A-L., Cervantes-Pérez, F., Assar, S.: **A Quantitative Analysis of Student Learning Styles and Teacher Teachings Strategies in a Mexican Higher Education Institution**, *Journal of Applied Research and Technology (JART)*, Vol.10, n°3, pp.289–308, 2012 [[http://www.jart.ccadet.unam.mx/jart/volumen10\\_3.htm](http://www.jart.ccadet.unam.mx/jart/volumen10_3.htm)]

8. Assar, S., El Amrani, R., Watson, R.T.: **ICT and education: A critical role in human and social development**, *Information Technology for Development*, Vol.16, n°3, pp. 151–158, 2010  
[<http://www.tandfonline.com/toc/titd20/16/3>]
9. Franzoni, A-L., Assar, S.: **Student Learning Styles Adaptation Method Based on Teaching Strategies and Electronic Media**; *Journal of Educational Technology & Society*, 12(4), pp.15–29, 2009  
[<http://www.ifets.info/issues.php?id=45>]
10. Assar, S., Boughzala, I.: **Empirical evaluation of public e-procurement platforms in France**, *International Journal of Value Chain Management*, vol.2, n°1, pp.90–108, Inderscience Enterprises Ltd., 2008  
[<http://www.inderscience.com/info/inarticle.php?artid=16120>]
11. Assar, S., Beauvallet, G., Boughzala, I.: Editorial : L'administration électronique, un point de rencontre entre la pratique des projets et la recherche en management des SI , *Système d'Information et Management (SIM)*, vol.10, n°1, pp. 3–13, 2005 [<http://revuesim.org/sim/issue/view/41>]

### VII.2.2 Co-édition d'ouvrage

12. I. Boughzala, M. Janssen, S. Assar (editors): **Case Studies in e-Government 2.0: Changing Citizen Relationships**, Springer, 2014, ISBN 978-3-319-08080-2. [<http://link.springer.com/book/10.1007/978-3-319-08081-9>]
13. S. Assar, I. Boughzala, I. Boydens (editors): **Practical Studies in E-Government: Best Practices from Around the World**, Springer, 2010, ISBN 978-1441975324 [<http://link.springer.com/book/10.1007/978-1-4419-7533-1>]
14. S. Assar, I. Boughzala (Editors): **Administration électronique : constats et perspectives** Hermès Lavoisier editions, 2007, ISBN 978-2-7462-1546-7. [<http://www.lavoisier.fr/livre/informatique/administration-electronique-constats-et-perspectives/assar/descriptif-9782746215467>]

### VII.2.3 Chapitres d'ouvrage

15. Assar, S.: **Information and Communications Technology in Education**. International Encyclopedia of the Social & Behavioral Sciences, Second Edition, Elsevier, 2015, pages 66–71  
[<http://www.sciencedirect.com/science/article/pii/B9780080970868921044>]
16. Boughzala, I., Janssen, M., S. Assar: **E-Government 2.0: Back to Reality, a 2.0 Application to Vet**, dans "Case Studies in e-Government 2.0: Changing Citizen Relationships", by I. Boughzala, M. Janssen, S. Assar (editors), Springer, 2014.
17. S. Assar, C. Cauvet, A. Front, C. Hug, J. Ralyté: **Vers une ingénierie de l'adaptation de méthodes**, dans P. Lopisteguy, D. Rieu, P. Roose (editors): "L'adaptation dans tous ses états" Cépaduès-éditions, France, 2012, p. 227-255 [<http://www.cepadues.com/index.aspx?p=notice&id=772>]
18. S. Assar, C. Cauvet, A. Front, C. Hug, J. Ralyté: **Les méthodes adaptables**, dans P. Lopisteguy, D. Rieu, P. Roose (editors): "L'adaptation dans tous ses états" Cépaduès-éditions, France, 2012, pp. 203-225  
[<http://www.cepadues.com/index.aspx?p=notice&id=772>]
19. S. Assar: **Système d'Information et fonctions de l'entreprise**, dans J. Thevenot (Editeur) : "MASTER Systèmes d'information" EDITIONS AGPA-ESKA, France, 2011, ISBN 978-2-7472-1765-1, pp. 219-251.
20. G. Beauvallet, Y. Boughzala, S. Assar: **E-Procurement, from Project to Practice: Empirical Evidence from the French Public Sector**, dans S. Assar, I. Boughzala, I. Boydens (editors): "Practical Studies in E-Government: Best Practices from Around the World", Springer, 2010, pp. 13-28.
21. S. Assar, I. Boughzala, I. Boydens: **Back to Practice, a Decade of Research in E-Government**, dans S. Assar, I. Boughzala, I. Boydens (editors): "Practical Studies in E-Government: Best Practices from Around the World", Springer Science+ Business Media, New York, 2010, pp. 1-12.
22. S. Assar, I. Boughzala: **Évaluation des plates-formes d'achat public**, dans S. Assar, I. Boughzala (editors): "Administration électronique - Constats et perspectives", Hermès Lavoisier Editions, January 2007.
23. Boughzala, S. Assar: **Modélisation et analyse des procédures d'achat public**, dans S. Assar, I. Boughzala (editors): "Administration électronique - Constats et perspectives", Hermès Lavoisier Editions, 2007.
24. S. Assar, I. Boughzala, Y. Boughzala: **Collaborative features in French public e-procurement**, dans F. Feltz, B. Otjacques, A. Oberweis, N. Poussing (Editors): "AIM 2006: Information Systems and Collaboration: State of the Art and Perspectives", Lecture Notes in Informatics, vol. p-92, pp. 83-103, 2006. ISBN 978-3-88579-186-7  
[<https://www.gi.de/service/publikationen/lni/gi-edition-proceedings-2006/gi-edition-lecture-notes-in-informatics-lni-p-92.html>]

## VII.2.4 Conférences et ateliers avec comité de programme

25. S. Assar, R. El Amrani: **The fragmentation of theoretical contributions in ERP research: an exploratory study**, Proceedings *49th Hawaii Int. Conf. on System Sciences* (HICSS 2016), January 5-8, 2016  
[<https://www.computer.org/csdl/proceedings/hicss/2016/5670/00/index.html>]
26. S. Assar: **Model Driven Requirements Engineering: Mapping the Field and Beyond**, Proceedings of the *4th Model Driven Requirements Eng. workshop* (MoDRE'14), August 26, 2014, Karlskrona, Sweden  
[<http://ieeexplore.ieee.org/document/6890820/>]
27. R. Latif, H. Abbas, S. Assar, S. Latif: **Analyzing Feasibility for Deploying Very Fast Decision Tree for DDoS Attack Detection in Cloud-Assisted WBA**, *Intelligent Computing Theory*, Proceedings of the *10th Int. Conf. ICIC 2014*, Taiyuan, China, August 3-6, 2014, LNCS vol.8588, Springer, pp 507-519  
[[http://link.springer.com/chapter/10.1007/978-3-319-09333-8\\_57](http://link.springer.com/chapter/10.1007/978-3-319-09333-8_57)]
28. Cruz, I. Boughzala, S. Assar: **Technology Acceptance and Actual Use with Mobile Learning: First Stage for Studying the Influence of Learning Styles on the Behavioral Intention**, Proceedings of the *European Conf. on IS* (ECIS'14), 9-11 June 2014, Tel-Aviv, Israel [<http://aisel.aisnet.org/ecis2014/proceedings/track16/12/>]
29. S. Assar, R. El Amrani: **A preliminary analysis of theorizing effort in ERP research**, Proceedings of the *8th IEEE Int. Conf. on Research Challenges in Information Science* (RCIS'14), May 28-30, 2014, Marrakesh, Morocco  
[<http://ieeexplore.ieee.org/document/6861032/>]
30. S. Mallouli, S. Assar, C. Souveyet: **Process behavior meta-modeling for the derivation of enactment engines**, Proceedings of the *8th IEEE Int. Conf. on Research Challenges in Information Science* (RCIS'14), May 28-30, 2014, Marrakesh, Morocco [<http://ieeexplore.ieee.org/document/6861088/>]
31. S. Mallouli, S. Assar, C. Souveyet: **Proposition d'une démarche de type IDM pour la construction d'outils d'exécution de processus**, *INFORSID 2014*, 20-23, 2014, Lyon, France.
32. S. Assar, R. El Amrani: **Théories et théorisation : esquisse d'une analyse avec la recherche sur les ERP**, *AIM Research Methods in Information Systems Workshop* (RMiIS), 2<sup>nd</sup> edition, April 4<sup>th</sup>, 2014, Nantes, France.
33. R. Latif, H. Abbas, S. Assar, Q. Ali : **Cloud Computing Risk Assessment: A Systematic Literature Review**, Proceedings of the *7th Int. Symposium on Digital Forensics and Information Security* (in conjunction with FutureTech 2013), September 4-6, 2013, Gwangju - Korea, Lecture Notes in Electrical Engineering, Springer, pp. 285-295 [[http://link.springer.com/chapter/10.1007/978-3-642-40861-8\\_42](http://link.springer.com/chapter/10.1007/978-3-642-40861-8_42)]
34. S. Mallouli, S. Assar: **Enacting a Requirement Engineering Process with Meta-Tools: an Exploratory Project**, Proceedings of the *8th Int. Multi-Conference on Computing in the Global Inf. Technology* (ICCGI 2013), July 21-26, 2013, Nice - France, pp. 208-213.  
[[http://www.thinkmind.org/index.php?view=article&articleid=iccg\\_i\\_2013\\_10\\_30\\_10309](http://www.thinkmind.org/index.php?view=article&articleid=iccg_i_2013_10_30_10309)]
35. S. Assar: **Pour une revue rigoureuse et systématique de la littérature en MIS : étude comparative et multidisciplinaire**, *AIM Research Methods in Information Systems Workshop* (RMiIS), 1<sup>st</sup> Edition, April 5<sup>th</sup>, 2013, Nantes, France. [<http://aim.asso.fr/index.php/mediatheque/summary/29-rmiis13/949-pour-une-revue-rigoureuse-et-systematique-de-la-litterature-en-mis-etude-comparative-et-multidisciplinaire>]
36. S. Assar, I. Boughzala : **E-Government Evolution Priorities from a Web 2.0 Perspective: An Exploratory Field Study**, Proceedings *46<sup>th</sup> Hawaii Int. Conf. on System Sciences* (HICSS), January 07-10, 2013, pp.2072-2081  
[<https://www.computer.org/csdl/proceedings/hicss/2013/4892/00/index.html>]
37. K. Wnuk, M. Borg, S. Assar : **Towards Scalable Information Modeling of Requirements Architectures**, *Advances in Conceptual Modeling - ER 2012 Workshops*, MoDIC Workshop, October 15-18 2012, Florence - Italy, LNCS 7518, Springer, pp. 141-150 [[http://link.springer.com/chapter/10.1007/978-3-642-33999-8\\_17](http://link.springer.com/chapter/10.1007/978-3-642-33999-8_17)]
38. Cruz, Y., S. Assar, I. Boughzala: **Exploring Teacher's Perception and Potential Use of Mobile Learning in a Business School**, Proceedings *18<sup>th</sup> American Conference on Information Systems* (AMCIS'12), August 9-11, 2012, Seattle – USA [<http://aisel.aisnet.org/amcis2012/proceedings/ISEducation/26/>]
39. S. Mallouli, S. Assar, C. Souvyet : **A Model-Driven Approach to Process Enactment**, Poster at *7<sup>th</sup> Int. Conf. on Software and Data Technologies* (ICSOFT), July 24-27, 2012, Rome – Italy.
40. S. Assar, C. Souveyet, K. Aljoumaa: **A goal-oriented perspective on approaches to web service discovery**, Proceedings of the *6<sup>th</sup> Int. IEEE Workshop on Requirements Engineering For Services* (REFS 2012), in conjunction with COMPSAC 2012, July 16-20, 2012, Izmir – Turkey [<http://ieeexplore.ieee.org/document/6341600/>]
41. Cruz, Y., I. Boughzala, S. Assar: **Opportunities and Obstacles for Mobile Learning in a Business School**, Proceedings of the *2<sup>nd</sup> Int. Conf. on Comm. and Inf. Tech.* (ICCIT'12), June 26-28, 2012, Hammamet – Tunisia  
[<http://ieeexplore.ieee.org/document/6285824/>]

42. S. Assar, I. Boughzala, Th. Isckia: **E-government Trends in the Web 2.0 Era and the Open Innovation Perspective: An Exploratory Field Study**, Proceedings of the 10<sup>th</sup> conference on Electronic Government (EGOV 2011), August 28-September 2, 2011, Delft – Netherlands [[http://link.springer.com/chapter/10.1007/978-3-642-22878-0\\_18](http://link.springer.com/chapter/10.1007/978-3-642-22878-0_18)]
43. S. Assar, S. Mallouli, C. Souvyet : **A behavioral perspective in meta-modeling**, Proceedings of the 6<sup>th</sup> Int. Conf. on Software and Data Technologies (ICSOFT), July 18-21, 2011, Seville – Spain.
44. K. Aljoumaa, S. Assar: **Matching user requirements with query formulations in intentional service oriented computing**, Proceedings of the 5<sup>th</sup> Int. IEEE Workshop on Requirements Engineering For Services (REFS 2011), in conjunction with COMPSAC 2011, July 18-21, 2011, Munich – Germany [<http://ieeexplore.ieee.org/document/6032284/>]
45. Boughzala, S. Assar, Th. Isckia: **Étude exploratoire sur les perspectives d'évolution de l' e-gouvernement à l'ère de l'innovation ouverte et du Web 2.0** , XX<sup>th</sup> Annual meeting of AIMS, June 7-9, 2010, Nantes – France. [<http://www.strategie-aims.com/events/conferences/4-xxeme-conference-de-l-aims/communications/1359-etude-exploratoire-sur-les-perspectives-devolution-de-l-e-gouvernement-a-l-ere-de-l-innovation-ouverte-et-du-web-2-0>]
46. K. Aljoumaa, S. Assar, C. Souveyet: **Publication des services intentionnels, Processus de guidage d'annotation du descripteur intentionnel**, Proceedings of the 29<sup>th</sup> INFORSID conference, May 24-26, 2011, Lille – France. [<https://hal.archives-ouvertes.fr/hal-00706092/>]
47. S. Mallouli, S. Assar, C. Souvyet : **Pour une perspective comportementale dans les méta-modèles de processus**, Proceedings of 29<sup>th</sup> INFORSID conference, May 24-26, 2011, Lille – France. [<http://liris.cnrs.fr/inforsid/?q=node/502>]
48. K. Aljoumaa, S. Assar, C. Souveyet: **Publishing Intentional Services using extended semantic annotation**, Proceedings of the 5<sup>th</sup> IEEE International Conference on Research Challenges in Information Science (RCIS), May 19-21, 2011, Guadeloupe, French West Indies – France [<http://ieeexplore.ieee.org/document/6006842/>]
49. K. Aljoumaa, S. Assar, C. Souveyet: **Reformulating user's queries for Intentional Services Discovery using an Ontology-based Approach**, Proceedings 4<sup>th</sup> IFIP Int. Conf. on New Technologies, Mobility and Security (NTMS'2011), 7-10 February 2011, Paris – France [<http://ieeexplore.ieee.org/document/5721075/>]
50. K. Aljoumaa, S. Assar, C. Souveyet: **Publishing intentional services using new annotation for WSDL**, Proceedings of the 12<sup>th</sup> Int. Conf. on Information Integration and Web-based Applications & Services (iiWAS '10), ACM, 08-10 November 2010, Paris – France [<http://dl.acm.org/citation.cfm?id=1967640>]
51. I. Boughzala, S. Assar, Nicholas C. Romano Jr.: **An E-government Field Study of Process Virtualization Modeling**, Proceedings of the 11<sup>th</sup> Annual meeting of Group Decision & Negotiation 2010 (GDN 2010), June 21-23, 2010, Delft – The Netherlands.
52. H. El Ghazi, S. Assar: **Evaluation orientée coût/valeur du besoin de traçabilité des exigences dans un projet**, Proceedings of the 27<sup>th</sup> INFORSID conference, May, 25-27, 2009, Toulouse – France. [<http://liris.cnrs.fr/inforsid/?q=node/427>]
53. J. Heili, S. Assar: **An Empirical Enquiry into the adoption of Open Source Software by individual users in France**, IADIS International Conference on Information Systems, February, 25–27, 2009, Barcelona – Spain. [<http://www.iadisportal.org/is-2009-proceedings>]
54. A.-L. Franzoni Velázquez, S. Assar, B. Defude, J. Rojas, C.P. Francisco: **Taxonomía de enseñanza adaptativa basada en los estilos de aprendizaje del estudiante, las estrategias didácticas y los medios electrónicos**, (en Espagnol) Proceedings of the III Congreso de Estilos de Aprendizaje 2008, pp. 37-46, July 7–9, 2008, Cáceres – Spain.
55. A.-L. Franzoni Velázquez, S. Assar, B. Defude, J. Rojas: **Student Learning Styles Adaptation Method Based on Teaching Strategies and Electronic Media**, Proceedings The 8<sup>th</sup> IEEE International Conference on Advanced Learning Technologies, July 1–5, 2008, Santander – Spain [<http://ieeexplore.ieee.org/document/4561832/>]
56. H. El-Ghazi, S. Assar: **A Multi View based Traceability Management Method**, Proceedings IEEE Int. Conf. on Research Challenges in Information Science (RCIS'08), pp. 393-400, June 03-06, 2008, Marrakech – Morocco, ISBN: 978-1-4244-1677-6 [<http://ieeexplore.ieee.org/document/4632129/>]
57. A.-L. Franzoni Velázquez, S. Assar: **Using Learning Styles to Enhance an e-Learning System**, Proceedings 6<sup>th</sup> European Conference on e-Learning, October 4–5, 2007, Copenhagen - Denmark.
58. J. Heili, S. Assar: **Enquête empirique sur l'adoption des logiciels libres par l'utilisateur individuel**, Proceedings of 12<sup>th</sup> AIM conference 2007, June 17–19, 2007, Lausanne – Switzerland.
59. S. Assar: **Estimation of Business Process System Adequacy**, Proceedings BPMDS'07 workshop, CAISE'07 conference, June 10<sup>th</sup>-15<sup>th</sup>, 2007, pp. 385–388, Trondheim – Norway.
60. S. Assar, I. Boughzala, Y. Boughzala: **Aspects collaboratifs dans la dématérialisation des achats publics en France** , Proceedings of the 11<sup>th</sup> AIM conference 2006, June 7–9, 2006, Luxembourg.

61. S. Assar, I. Boughzala: **E-procurement platforms in the French public sector**, Proceedings of 2<sup>nd</sup> IEEE International Conference on Information & Communication Technology – ICTTA'06, Volume 1, 24–28 April 2006, Page(s):906 – 910, Damascus – Syria [<http://ieeexplore.ieee.org/document/1684494/>]
62. S. Assar, C. Ben-Achour, S. Si-Saïd: **Un Modèle pour la Spécification des Processus d'analyse des Systèmes d'Information**, Proceedings of INFORSID 2000 conference – Lyon, May 2000.
63. S. Assar, C. Rolland: **Traduction d'un schéma conceptuel Remora sur un SGBD relationnel**, Proceedings of INFORSID 1992 conference, Clermont-Ferrand – France, 1992.
64. Y. Lingat, S. Assar, P. Colignon and C. Rolland: **PROQUEL: a PROgramming QUery Language**, in R. Hull, R. Morrison, D. W. Stemple (Eds.): Proceedings 2<sup>nd</sup> International Workshop on Database Programming Languages, 4–8 June, 1989, Oregon – USA.

### VII.2.5 Autres publications

65. Cruz, Y., I. Boughzala, S. Assar : **Understanding the Influence of Learning Styles in Mobile Learning Usage from the Perspective of Technology Adoption and Continuance Use**, *pre-ICIS AIM workshop*, Dec. 15, 2013, Milano, Italy.
66. S. Assar : **Diversity of literature review approaches: illustration with TAM secondary studies**, In “Building up or Piling Up? The Literature Review in IS research”, *a pre-ECIS workshop*, June 5, 2013, Utrecht, Netherlands.
67. J. Heili, S. Assar: **An Empirical Enquiry into the adoption of Open Source Software by individual users in France**, *EUROMOT 2008 conference*, September 17-19, 2008, Nice – France.
68. C. Souveyet, S. Assar: **Exécutabilité des modèles : réflexions et expériences**, *MADSI Workshop at INFORSID 2007 conference*, May 22–25 2007, Perros-Guirec – France.
69. S. Assar: **Meta-modeling for schema transformation and code generation**, CAISE\*94 Doctoral Consortium, Utrecht, Holland, 1994.
70. J-Y. Lingat, S. Assar, P. Colignon, C. Rolland: **Apports du langage PROQUEL pour le prototypage d'applications Bases de Données**, 10<sup>th</sup> Franco-Tunisian computer science seminar, May 23–25, 1989, Tunis – Tunisia.

## VII.3 Financement de projets de recherche

- Projet **ProAdmin**, du 1er janvier 2005 au 31 décembre 2005, budget global 30k€, détails sur <http://www-public.tem-tsp.eu/~assar/proadmin/>

## VII.4 Animation de la recherche et travaux d'expertise

### VII.4.1 Membre de comités éditoriaux

- **Éditeur associé**
  - Information Technology for Development Journal (S. Qureshi, [squreshi@ist.unomaha.edu](mailto:squreshi@ist.unomaha.edu)) : depuis 2012
  - Int. J. of Social and Organizational Dynamics in IT (M. Knight, [knightm@uwgb.edu](mailto:knightm@uwgb.edu)) : 2010 - 2015
- **Comité éditorial**
  - J. of Information Technology Management (JITM) (A. Bento, [abento@ubalt.edu](mailto:abento@ubalt.edu))
- **Relecteur occasionnel**
  - Ingénierie des Systèmes d'Information: 2014, 2015
  - Journal of Theoretical and Applied E-commerce Research (JTAER): 2014 - 2016
  - Systèmes d'Information et Management (SIM) : 2014 - 2016
  - Journal of the AIS (JAIS) : 2011
  - Communication of the AIS (CAIS) : 2006
  - Journal of Internet Commerce : 2007, 2014

### VII.4.2 Coordination de numéros spéciaux de revues

- **Co-éditeur** (avec Bénédicte Le Grand) d'un numéro spécial (publié en décembre 2015) de la revue Franco-phone *Ingénierie des Systèmes d'Information* sur le thème des méthodes en ingénierie des SI (contact: Corine Cauvet, [corine.cauvet@isis.org](mailto:corine.cauvet@isis.org)) [<http://isi.revuesonline.com/resnum.jsp?editionId=3045>]
- **Co-éditeur** (avec R. Watson et R. El Amrani) d'un numéro spécial (vol.16, n°3, Sept. 2010) de la revue *Information Technology for Development Journal* sur le thème de l'enseignement à l'aide des TIC dans les pays en voie de développement (contact: Sajda Qureshi, [squreshi@ist.unomaha.edu](mailto:squreshi@ist.unomaha.edu)) [<http://www.tandfonline.com/toc/titd20/16/3>]

### VII.4.3 Comité de programme de conférences et atelier de recherche

- CAISE: 2016
- RCIS (Research Challenge in Information Science) conference: 2015, 2014, 2013, 2012.
- INFORSID conference: 2014, 2012, 2011.
- CAISE Forum: 2015, 2014.
- RMIIS (AIM Workshop on Research Methods in Information Systems): 2015, 2014.
- HICSS (Hawaii Int. Conf. on Systems Science): 2009, 2013, 2014, 2015
- ICIS (Int. Conf. on Information Systems): 2008, 2009, 2010, 2013, 2014
- ECIS (European Conf. on IS): 2007, 2013, 2014
- EGOV (Int. Conf. on e-Government): 2008, 2009, 2010

### VII.4.4 Organisation et animation de la recherche

- **Program Chair** (avec le prof. Oscar Pastor) : RCIS 2017
- **Proceedings chair** : CAISE'16, RCIS 2013
- **Track co-chair** (avec K. Mohan and C. Amrit) on *Methods, Tools and Human Factors in IS/IT Management*, at the European Conference on IS (ECIS) : 2016, 2015, 2013
- **Program chair** for the *Posters & Demos track* at IEEE RCIS conference : 2016, 2015
- **Co-chair** (avec R. El Amrani) of the pre-ICIS'08 Int. Development workshop at ICIS conference : 2008
- **Organization committee and webmaster** for the 13 Int. Conf. on Requirements Engineering (RE'05) : 2005
- **Webmaster** for the IFIP WG8.1 "Working Conference on Engineering IS in the Internet Context" (Septembre 2002, Kanazawa – Japon).
- **Co-organizing chair** of the Doctoral Consortium at the CAISE'95 conference, Jyväskylä, Finlande.

### VII.4.5 Comités scientifiques

- 2005 – 2010 : member externe du comité de spécialistes à l'université Paris 1 – Sorbonne

## VII.5 Direction de recherche

### VII.5.1 Encadrement de thèse

- 2014 Yaneli Cruz: **Examining the Effect of Learning Styles on Mobile Learning Adoption**, , soutenue le 14 décembre 2014, en collaboration avec le prof. Imed Boughzala, Telecom Ecole de Management.
- 2014 Sana MALLOULI: **Méta-modélisation du Comportement d'un Modèle de Processus : Une Démarche de Construction d'un Moteur d'Exécution**, soutenue le 25 juillet 2014, en collaboration avec la prof. C. Souveyet, Université Paris I La Sorbonne.
- 2011 Kadan ALJOUAAA: **Modélisation intentionnelle et annotation sémantique pour la réutilisation de services métiers**, soutenue le 20 octobre 2011, en collaboration avec la prof. Colette Rolland et la prof. C. Souveyet, Université Paris I La Sorbonne.
- 2009 Ana-Lidia FRANZONI: **Exploring the relationships between learning style and e-media in distance learning**, soutenue le 10 décembre 2009, en collaboration avec le prof. Jean-Louis Ermine, TELECOM & Management Sud Paris – Troyes University (France) – L'instituto Tecnologico Autonomo de Mexico (ITAM).
- 2009 Hamid El-GHAZI: **MV-TMM: Une approche multi vues pour la gestion de la traçabilité des exigences**, soutenue le 13 juillet 2009, en collaboration avec le prof. Colette Rolland, Université Paris I La Sorbonne.
- 2005 Marc-Henri EDME: **Proposition pour la modélisation intentionnelle et le guidage de l'usage des systèmes d'information**, soutenue le 9 juillet 2005 en collaboration avec le prof. Colette Rolland, Université Paris I La Sorbonne.

### VII.5.2 Appartenance à des jurys de thèse

- 2005 Iyad ZOUKAR : **Méthode d'Ingénierie des Besoins pour la personnalisation d'un Progiciel de Gestion Intégré**, Université Paris I La Sorbonne, avril 2005.
- 2003 Juan Fernando VELEZ: **Proposition d'un environnement logiciel centré processus pour l'ingénierie des systèmes d'information**, Université Paris I La Sorbonne, janvier 2003.

## VIII. Références bibliographiques

- [1] Akoka, J. et Comyn-Wattiau, I., Éd., *Encyclopédie de l'informatique et des systèmes d'information*. Paris: Vuibert, 2006.
- [2] Akoka, J. et Comyn-Wattiau, I., *L'informatique*. Paris: Presses Universitaires de France, 2002.
- [3] Alavi, M. et Leidner, D. E., « Research Commentary: Technology-Mediated Learning—A Call for Greater Depth and Breadth of Research », *Information Systems Research*, vol. 12, n° 1, p. 1-10, 2001.
- [4] Alter, S., « Defining information systems as work systems: implications for the IS field », *European Journal of IS (EJIS)*, vol. 17, n° 5, p. 448-469, 2008.
- [5] Anbalagan, P. et Vouk, M., « On predicting the time taken to correct bug reports in open source projects », in *Proceedings of the IEEE Int'l Conf. on Software Maintenance (ICSM'09)*, 2009, p. 523-526.
- [6] Andersen, H. et Hepburn, B., « Scientific Method », *The Stanford Encyclopedia of Philosophy*. 2016.
- [7] Assar, S., « Meta-modélisation pour la transformation des schémas et la génération de code », Thèse de doctorat, Université Paris 6 - UPMC, Paris, France, 1995.
- [8] Assar, S., Ben Achour, C., et Si-Saïd, S., « Un modèle pour la Spécification des Processus d'Analyse des Systèmes d'Information », in *Actes 18ème congrès INFORSID*, Lyon, France, 2000, p. 287-301.
- [9] Astrahan, M. M., Blasgen, M. W., Chamberlin, D. D., Eswaran, K. P., Gray, J. N., Griffiths, P. P., King, W. F., Lorie, R. A., McJones, P. R., Mehl, J. W., Putzolu, G. R., Traiger, I. L., Wade, B. W., et Watson, V., « System R: Relational Approach to Database Management », *ACM Trans. Database Syst.*, vol. 1, n° 2, p. 97-137, 1976.
- [10] Bailey, K. D., *Typologies and taxonomies: an introduction to classification techniques*, vol. 102. Sage Publications, Inc, 1994.
- [11] Banville, C. et Landry, M., « Can the Field of MIS be Disciplined? », *Comm. of the ACM*, vol. 32, n° 1, p. 48-60, 1989.
- [12] Barreau, H., *L'épistémologie*, 8e édition. Paris: Presses Universitaires de France, 2013.
- [13] Barrow, R., Carroll, J., et Smith, R., « Towards An Integrated Model Of ISDM Tailoring », présenté à 20th Australasian Conference on Information Systems, 2009.
- [14] Batet, M., « Ontology-based semantic clustering », *AI Communications*, vol. 24, n° 3, p. 291-292, 2011.
- [15] Bem, D. J., « Writing a review article for Psychological Bulletin », *Psychological Bulletin*, vol. 118, n° 2, p. 172, 1995.
- [16] Bendraou, R., Combemale, B., Cregut, X., et Gervais, M.-P., « Definition of an Executable SPEM 2.0 », in *Proceedings 14th Asia-Pacific Software Engineering Conf. (APSEC 2007)*, 2007, p. 390-397.
- [17] Bettin, J., « Model-driven software development », *MDA Journal*, vol. 1, 2004.
- [18] Bhattacharya, P. et Neamtiu, I., « Bug-fix time prediction models: can we do better? », in *Proceedings of the 8th Working Conf. on Mining Software Repositories (MSR'11)*, New York, NY, USA, 2011, p. 207-210.
- [19] Bhattacharjee, A. et Premkumar, G., « Understanding changes in belief and attitude toward information technology usage: a theoretical model and longitudinal test », *MIS quarterly*, p. 229-254, 2004.
- [20] Bichler, M., Heinzl, A., et Aalst, W. van der, « BISE and the Engineering Sciences », *Business & Information Systems Engineering*, vol. 58, n° 2, p. 105-106, 2015.
- [21] Bjørner, D., « Semantics », in *Software Engineering 2*, Berlin/Heidelberg: Springer, 2006, p. 151-172.
- [22] Boehm, B. et Basili, V. R., « Software Defect Reduction Top 10 List », *Computer*, vol. 34, n° 1, p. 135-137, 2001.
- [23] Boell, S. K. et Cecez-Kecmanovic, D., « On being 'systematic' in literature reviews in IS », *Journal of Information Technology*, vol. 30, n° 2, p. 161-173, 2015.
- [24] Borg, M., « Advancing Trace Recovery Evaluation - Applied Information Retrieval in a Software Engineering Context », MSc Thesis, Lund Tekniska Högskola, Lund, Sweden, 2012.
- [25] Borg, M., « Embrace your Issues: Compassing the Software Engineering Landscape using Bug Reports », présenté au Doctoral Symposium de la 29th IEEE/ACM Int'l Conf. on Automated Software Engineering (ASE'14), Västerås, Sweden, 2014.

- [26] Bougie, G., Treude, C., German, D. M., et Storey, M., « A comparative exploration of FreeBSD bug lifetimes », in *Proceedings of the 7th IEEE Working Conference on Mining Software Repositories (MSR'10)*, 2010, p. 106-109.
- [27] Brafman, N., « En amphi de droit, Twitter remplace la main levée », *Le Monde.fr*, 20-avr-2016.
- [28] Bresciani, P., Perini, A., Giorgini, P., Giunchiglia, F., et Mylopoulos, J., « Tropos: An Agent-Oriented Software Development Methodology », *Autonomous Agents and Multi-Agent Systems*, vol. 8, n° 3, p. 203-236, 2004.
- [29] Breton, E. et Bézivin, J., « Towards an understanding of model executability », in *Proceedings 2nd Int. Conf. on Formal Ontology in Information Systems - Vol. 2001*, Ogunquit, ME, USA, 2001.
- [30] Brinkkemper, S., Saeki, M., et Harmsen, F., « Assembly techniques for method engineering », in *CAISE'98*, vol. 1413, B. Pernici et C. Thanos, Éd. Berlin/Heidelberg: Springer, 1998, p. 381-400.
- [31] Brooks, A., Roper, M., Wood, M., Daly, J., et Miller, J., « Replication's Role in Software Engineering », in *Guide to Advanced Empirical Software Engineering*, F. Shull, J. Singer, et D. I. K. Sjøberg, Éd. London, UK: Springer, 2008, p. 365-379.
- [32] Bryant, B., Gray, J., Mernik, M., Clarke, P., France, R., et Karsai, G., « Challenges and Directions in Formalizing the Semantics of Modeling Languages », *Computer Science and Information Systems*, vol. 8, n° 2, p. 225-253, 2011.
- [33] Budgen, D., Burn, A. J., Brereton, O. P., Kitchenham, B. A., et Pretorius, R., « Empirical evidence about the UML: a systematic literature review », *Software: Practice and Experience*, vol. 41, n° 4, p. 363-392, 2011.
- [34] Bunge, M., *Scientific Research 1: The Search for System*, vol. I. New York: Springer, 1967.
- [35] Bunge, M., *Treatise on Basic Philosophy. Volume 3. Ontology I: The furniture of the world*. Dordrecht, Netherlands: D. Reidel Publishing Company, 1977.
- [36] Carlsson, C., Carlsson, J., Hyvonen, K., Puhakainen, J., et Walden, P., « Adoption of Mobile Devices/Services - Searching for Answers with the UTAUT », in *Proceedings of the 39th Annual Hawaii Int. Conf. on System Sciences (HICSS'06)*, 2006.
- [37] Carter, D., « Higher education's best mobile technology programs », *eCampus News*, 01-juin-2010. [En ligne]. Disponible sur: <http://www.ecampusnews.com/technologies/higher-educations-best-mobile-technology-programs/>. [Consulté le: 09-oct-2016].
- [38] Cassidy, S., « Learning Styles: An overview of theories, models, and measures », *Educational Psychology*, vol. 24, n° 4, p. 419-444, 2004.
- [39] Ceri, S., Brambilla, M., et Fraternali, P., « The History of WebML Lessons Learned from 10 Years of Model-Driven Development of Web Applications », in *Conceptual Modeling: Foundations and Applications*, A. T. Borgida, V. K. Chaudhri, P. Giorgini, et E. S. Yu, Éd. Springer Berlin Heidelberg, 2009, p. 273-292.
- [40] Chaudron, M. R. V., Heijstek, W., et Nugroho, A., « How effective is UML modeling ? », *Software & Systems Modeling*, vol. 11, n° 4, p. 571-580, 2012.
- [41] Clark, T., Sammut, P., et Willans, J., *Applied Metamodelling: A Foundation for Language Driven Development.*, 2nd edition. Ceteva, 2008.
- [42] CNN, « Citoyens d'une société numérique », Conseil National du Numérique, 2013. Disponible en ligne: <https://cnnnumerique.fr/wp-content/uploads/2013/12/Rapport-CNNNum-10.12-1.pdf>
- [43] CNRS, « Projet d'établissement du CNRS ». 2002.
- [44] Coffield, F., Moseley, D., Hall, E., et Ecclestone, K., « Learning styles and pedagogy in post-16 learning: a systematic and critical review », Learning & Skills Research Centre, London, 2004.
- [45] Combemale, B., Crégut, X., Garoche, P.-L., et Thirioux, X., « Essay on Semantics Definition in MDE An Instrumented Approach for Model Verification. », *Journal of Software*, vol. 4, n° 9, p. 943-958, 2009.
- [46] Combemale, B., Rougemaille, S., Crégut, X., Migeon, F., Pantel, M., Maurel, C., et Coulette, B., « Towards rigorous metamodelling », in *2nd Int. Workshop on Model-Driven Enterprise IS (MDEIS)*, Paphos, Cyprus, 2006.
- [47] Cross, N., « Designerly ways of knowing », *Design studies*, vol. 3, n° 4, p. 221-227, 1982.
- [48] Cross, N., « Designerly ways of knowing: Design discipline versus design science », *Design issues*, vol. 17, n° 3, p. 49-55, 2001.
- [49] Cross, N., « Science and design methodology: A review », *Research in Engineering Design*, vol. 5, n° 2, p. 63-69, 1993.
- [50] Cruz, Y., « Learning Styles Effect on Mobile Learning Acceptance: A Continuance Intention Approach », PhD thesis, Université d'Evry-Val d'Essonne - Telecom Ecole de Management, Evry, France, 2014.
- [51] Cunningham, P., « A Taxonomy of Similarity Mechanisms for Case-Based Reasoning », *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, n° 11, p. 1532-1543, 2009.

- [52] Czarnecki, K. et Helsen, S., « Feature-based survey of model transformation approaches », *IBM Systems Journal*, vol. 45, n° 3, p. 621-645, 2006.
- [53] D'Ambros, M., Lanza, M., et Robbes, R., « An extensive comparison of bug prediction approaches », in *Proceedings of the 7th IEEE Working Conference on Mining Software Repositories (MSR'10)*, 2010, p. 31-41.
- [54] Daneva, M. et Wieringa, R. J., « Requirements Engineering for Enterprise Systems: What We Know and What We Don't Know? », in *Intentional Perspectives on Information Systems Engineering*, S. Nurcan, C. Salinesi, C. Souveyet, et J. Ralyté, Éd. Springer, 2010, p. 115-136.
- [55] Davis, F. D., « Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology », *MIS Quarterly*, vol. 13, n° 3, p. 319-340, 1989.
- [56] Davis, M. D., Sigal, R., et Weyuker, E. J., *Computability, Complexity, and Languages: Fundamentals of Theoretical Computer Science*, 2nd edition. Academic Press, 1994.
- [57] De Bra, P., Kay, J., et Weibelzahl, S., « Introduction to the Special Issue on Personalization », *IEEE Transactions on Learning Technologies*, vol. 2, n° 1, p. 1-2, 2009.
- [58] Denning, P. J., « Is Computer Science Science? », *Comm. of the ACM*, vol. 48, n° 4, p. 27-31, 2005.
- [59] Dijkman, R. M., Dumas, M., et Ouyang, C., « Semantics and analysis of business process models in BPMN », *Information and Software Technology*, vol. 50, n° 12, p. 1281-1294, 2008.
- [60] Dobing, B. et Parsons, J., « How UML is used », *Comm. of the ACM*, vol. 49, n° 5, p. 109-113, 2006.
- [61] Dowek, G., « Informatics in the classification of sciences », présenté à 2nd Int. Conf. on the History and Philosophy of Computing (HaPoC 2013), École Normale Supérieure (ENS), Paris, 2013.
- [62] Dowson, M. et Fernström, C., « Towards requirements for enactment mechanisms », in *Software Process Technology*, B. C. Warboys, Éd. Springer Berlin Heidelberg, 1994, p. 90-106.
- [63] Dubes, R. C., « Cluster analysis and related issues », in *Handbook of Pattern Recognition & Computer Vision*, C. H. Chen, L. F. Pau, et P. S. P. Wang, Éd. River Edge, NJ, USA: World Scientific Publishing Co., Inc., 1993, p. 3-32.
- [64] Dumez, H., « Eléments pour une épistémologie de la recherche qualitative en gestion », *le Libellio d'AEGIS*, vol. 6, n° 4, p. 3-15, 2010.
- [65] Easterbrook, S., Singer, J., Storey, M.-A., et Damian, D., « Selecting Empirical Methods for Software Engineering Research », in *Guide to Advanced Empirical Software Engineering*, F. Shull, J. Singer, et D. I. K. Sjøberg, Éd. London, UK: Springer, 2008, p. 285-311.
- [66] Edme, M., « Proposition pour la modélisation intentionnelle et le guidage de l'usage des systèmes d'information », Thèse de doctorat, Université Paris 1 La Sorbonne, France, 2005.
- [67] Egger, M. et Smith, G. D., « Meta-analysis: potentials and promise », *BMJ*, vol. 315, n° 7119, p. 1371-1374, 1997.
- [68] Ellis, C. A. et Nutt, G. J., « Modeling and enactment of workflow systems », in *Application and Theory of Petri Nets 1993*, M. A. Marsan, Éd. Berlin/Heidelberg: Springer, 1993, p. 1-16.
- [69] Engels, G., Hausmann, J. H., Heckel, R., et Sauer, S., « Dynamic Meta Modeling: A Graphical Approach to the Operational Semantics of Behavioral Diagrams in UML », in *«UML» 2000 — The Unified Modeling Language*, A. Evans, S. Kent, et B. Selic, Éd. Springer Berlin Heidelberg, 2000, p. 323-337.
- [70] Erwig, M. et Walkingshaw, E., « Semantics First! », in *Software Language Engineering*, vol. 6940, A. Sloane et U. Alßmann, Éd. Berlin/Heidelberg: Springer, 2012, p. 243-262.
- [71] Eugster, P. T., Felber, P. A., Guerraoui, R., et Kermarrec, A.-M., « The many faces of publish/subscribe », *ACM Comp. Surveys*, vol. 35, n° 2, p. 114-131, 2003.
- [72] Favre, J.-M., Gasević, D., Lammel, R., et Winter, A., « Guest Editors' Introduction to the Special Section on Software Language Engineering », *IEEE Transactions on Software Engineering*, vol. 35, n° 6, p. 737-741, 2009.
- [73] Feiler, P. H. et Humphrey, W. S., « Software process development and enactment: concepts and definitions », in *Software Process, 1993. Continuous Software Process Improvement, Second International Conference on the*, 1993, p. 28-40.
- [74] Felder, R. M. et Brent, R., « Understanding Student Differences », *Journal of Engineering Education*, vol. 94, n° 1, p. 57-72, 2005.
- [75] Felder, R. M. et Silverman, L. K., « Learning and teaching styles in engineering education », *Engineering education*, vol. 78, n° 7, p. 674-681, 1988.
- [76] Felder, R. M. et Spurlin, J., « Applications, Reliability and Validity of the Index of Learning Styles », *International Journal of Engineering Education*, vol. 21, n° 1, p. 103-112, 2005.
- [77] Fitzgerald, B., « Formalized systems development methodologies: a critical perspective », *Information Systems Journal*, vol. 6, n° 1, p. 3-23, 1996.

- [78] Fitzgerald, B., « The systems development dilemma: whether to adopt formalised systems development methodologies or not », in *Proceedings of the Second European Conference on Information Systems*, 1994, p. 691–706.
- [79] Fitzgerald, B., « The use of systems development methodologies in practice: a field study », *Information Systems Journal*, vol. 7, n° 3, p. 201–212, 1997.
- [80] Fitzgerald, B., Russo, N., et O’Kane, T., « An empirical study of system development method tailoring in practice », in *8th European Conference on IS*, 2000, p. 187–194.
- [81] France, R. et Rumpe, B., « Model-driven development of complex software: A research roadmap », in *2007 Future of Software Engineering*, 2007, p. 37–54.
- [82] Franzoni-Velázquez, A. L., « A Proposed Method for Adapting and Integrating Student Learning Style, Teaching Strategies and Electronic Media », PhD thesis, Univ. Techn. de Troyes(UTT) - Telecom et Management Sud Paris, Evry, France, 2009.
- [83] Gargantini, A., Riccobene, E., et Scandurra, P., « A semantic framework for metamodel-based languages », *Automated Software Engineering*, vol. 16, n° 3-4, p. 415-454, 2009.
- [84] Giachetti, G., Marín, B., López, L., Franch, X., et Pastor, O., « Verifying goal-oriented specifications used in model-driven development processes », *Information Systems*, vol. 64, p. 41-62, mars 2017.
- [85] Giger, E., Pinzger, M., et Gall, H., « Predicting the fix time of bugs », in *Proceedings 2nd Int. Workshop on Recommendation Systems for Software Eng. (RSSE’10)*, New York, NY, USA, 2010, p. 52–56.
- [86] Giraudin, J. P., « Complexité des systèmes d’information et de leur ingénierie », *e-TI - la revue électronique des technologies d’information*, n° 3, 2007.
- [87] Glass, R. L., « Editor’s corner: How can computer science truly become a science, and software engineering truly become engineering? », *Journal of Systems and Software*, vol. 10, n° 2, p. 75-76, 1989.
- [88] Gómez, O. S., Juristo, N., et Vegas, S., « Understanding replication of experiments in software engineering: A classification », *Information and Software Technology*, vol. 56, n° 8, p. 1033-1048, 2014.
- [89] González-Barahona, J. et Robles, G., « On the reproducibility of empirical software engineering studies based on data retrieved from development repositories », *Empirical Software Engineering*, vol. 17, n° 1, p. 75-89, 2012.
- [90] Gregor, S., « The nature of theory in information systems », *MIS Quarterly*, vol. 30, n° 3, p. 611–642, 2006.
- [91] Grönlund, Å. et Islam, Y. M., « A mobile e-learning environment for developing countries: the Bangladesh Virtual Interactive Classroom », *Information Technology for Development*, vol. 16, n° 4, p. 244-259, 2010.
- [92] Grosz, G., Rolland, C., Schwer, S., Souveyet, C., Plihon, V., Si-Said, S., Achour, C. B., et Gnaho, C., « Modelling and engineering the requirements engineering process: An overview of the NATURE approach », *Requirements Engineering*, vol. 2, n° 3, p. 115–131, 1997.
- [93] Grover, V. et Teng, J. T. C., « An examination of DBMS adoption and success in American organizations », *Information & Management*, vol. 23, n° 5, p. 239-248, 1992.
- [94] Guizzardi, G., Wagner, G., Guarino, N., et Sinderen, M. van, « An Ontologically Well-Founded Profile for UML Conceptual Models », in *Advanced Information Systems Engineering*, A. Persson et J. Stirna, Éd. Berlin/Heidelberg: Springer, 2004, p. 112-126.
- [95] Hannay, J. E., Dybå, T., Arisholm, E., et Sjøberg, D. I. K., « The effectiveness of pair programming: A meta-analysis », *Information and Software Technology*, vol. 51, n° 7, p. 1110-1122, 2009.
- [96] Hannay, J. E., Sjøberg, D. I. K., et Dybå, T., « A Systematic Review of Theory Use in Software Engineering Experiments », *IEEE Transactions on Software Engineering*, vol. 33, n° 2, p. 87-107, 2007.
- [97] Harel, D., « Statecharts: a visual formalism for complex systems », *Science of Computer Programming*, vol. 8, n° 3, p. 231-274, juin 1987.
- [98] Harel, D. et Rumpe, B., « Meaningful modeling: what’s the semantics of “semantics”? », *IEEE Computer*, vol. 37, n° 10, p. 64-72, 2004.
- [99] Harris, R. N., Dwyer, W. O., et Leeming, F. C., « Are Learning Styles Relevant in Web-Based Instruction? », *Journal of Educational Computing Research*, vol. 29, n° 1, p. 13-28, 2003.
- [100] Harrous, D., « Étude d’une approche générique d’ingénierie des modèles », Université Paris 1, Mémoire Master Recherche SID, 2011.
- [101] Henderson-Sellers, B. et Gonzalez-Perez, C., « Standardizing Methodology Metamodeling and Notation: An ISO Exemplar », in *Information Systems and e-Business Technologies*, vol. 5, R. Kaschek, C. Kop, C. Steinberger, G. Fliedl, W. Aalst, J. Mylopoulos, M. Rosemann, M. J. Shaw, et C. Szyperski, Éd. Springer Berlin Heidelberg, 2008, p. 1-12.

- [102] Henderson-Sellers, B., Gonzalez-Perez, C., et Ralyte, J., « Comparison of Method Chunks and Method Fragments for Situational Method Engineering », in *19th Australian Conf. on Software Eng. (ASWEC 2008)*, 2008, p. 479-488.
- [103] Hevner, A. et Chatterjee, S., « Design Science Research in Information Systems », in *Design Research in Information Systems*, vol. 22, A. Hevner et S. Chatterjee, Éd. Boston, MA: Springer US, 2010, p. 9-22.
- [104] Hevner, A. R., March, S. T., Park, J., et Ram, S., « Design Science in Information Systems Research », *MIS Quarterly*, vol. 28, n° 1, p. 75-105, 2004.
- [105] Hinze, A., Sachs, K., et Buchmann, A., « Event-based applications and enabling technologies », in *Proceedings 3rd ACM Int. Conf. on Distributed Event-Based Systems*, New York, USA, 2009, p. 1-15.
- [106] Hirschheim, R. et Klein, H., « A Glorious and Not-So-Short History of the Information Systems Field », *Journal of the AIS*, vol. 13, n° 4, 2012.
- [107] Hirschheim, R., Klein, H. K., et Lyytinen, K., *Information Systems Development and Data Modeling: Conceptual and Philosophical Foundations*. Cambridge: Press Syndicate of the University of Cambridge, 1995.
- [108] Hofmann, M. et Klinkenberg, R., *RapidMiner: Data Mining Use Cases and Business Analytics Applications*. CRC Press, 2014.
- [109] Holsapple, C. W. et Joshi, K. D., « Knowledge manipulation activities: results of a Delphi study », *Information & Management*, vol. 39, n° 6, p. 477-490, mai 2002.
- [110] Hovorka, D. et Lee, A., « Reframing Interpretivism and Positivism as Understanding and Explanation: Consequences for Information Systems Research », *ICIS 2010 Proceedings*, 2010.
- [111] Hu, P. J.-H., Hui, W., Clark, T. H. K., et Tam, K. Y., « Technology-Assisted Learning and Learning Style: A Longitudinal Field Experiment », *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, vol. 37, n° 6, p. 1099-1112, nov. 2007.
- [112] Iivari, J., « A Paradigmatic Analysis of Information Systems As a Design Science », *Scandinavian Journal of IS*, vol. 19, n° 2, p. 39-64, 2007.
- [113] Iivari, J. et Maansaari, J., « The usage of systems development methods: are we stuck to old practices? », *Information and Software Technology*, vol. 40, n° 9, p. 501-510, 1998.
- [114] Jain, A. K., « Data clustering: 50 years beyond K-means », *Pattern Recognition Letters*, vol. 31, n° 8, p. 651-666, 2010.
- [115] Jamshidi, P., Khoshnevis, S., Teimourzadegan, R., Nikravesh, A., et Shams, F., « Toward automatic transformation of enterprise business model to service model », in *ICSE Workshop on Principles of Engineering Service Oriented Systems (PESOS 2009)*, 2009, p. 70-74.
- [116] Jasny, B. R., Chin, G., Chong, L., et Vignieri, S., « Again, and Again, and Again ... », *Science*, vol. 334, n° 6060, p. 1225-1225, 2011.
- [117] Jayaratna, N. et Sommerville, I., « The role of information systems methodology in software engineering [Editorial] », *Software, IEE Proceedings -*, vol. 145, n° 4, p. 93-94, 1998.
- [118] Jouault, F., Allilaire, F., Bézivin, J., et Kurtev, I., « ATL: A model transformation tool », *Science of Computer Programming*, vol. 72, n° 1-2, p. 31-39, 2008.
- [119] Jouault, F. et Kurtev, I., « Transforming Models with ATL », in *Satellite Events at the MoDELS 2005 Conference*, J.-M. Bruel, Éd. Berlin/Heidelberg: Springer, 2006, p. 128-138.
- [120] Jung, C. G., *Psychological types: or the psychology of individuation*. Oxford, England: Harcourt, Brace, 1923.
- [121] Juristo, N. et Gómez, O. S., « Replication of Software Engineering Experiments », in *Empirical Software Engineering and Verification*, B. Meyer et M. Nordio, Éd. Berlin/Heidelberg: Springer, 2012, p. 60-88.
- [122] Juristo, N. et Vegas, S., « Using differences among replications of software engineering experiments to gain knowledge », in *Proceedings of the 3rd Int. Symposium on Empirical Software Eng. and Measurement (ESEM'09)*, 2009, p. 356-366.
- [123] Kelly, S., Lyytinen, K., et Rossi, M., « MetaEdit+ A fully configurable multi-user and multi-tool CASE and CAME environment », in *Advanced Information Systems Engineering (CAISE)*, vol. 1080, P. Constantopoulos, J. Mylopoulos, et Y. Vassiliou, Éd. Berlin/Heidelberg: Springer, 1996, p. 1-21.
- [124] Kelly, S., Lyytinen, K., Rossi, M., et Tolvanen, J. P., « MetaEdit+ at the Age of 20 », in *Seminal Contributions to Information Systems Engineering*, J. Bubenko, J. Krogstie, O. Pastor, B. Pernici, C. Rolland, et A. Sølvberg, Éd. Springer Berlin Heidelberg, 2013, p. 131-137.
- [125] Keung, J. et Kitchenham, B., « Experiments with Analogy-X for Software Cost Estimation », in *Proceedings of the 19th Australian Conf. on Software Engineering (ASWEC'08)*, 2008, p. 229-238.

- [126] Kitchenham, B. A., « The role of replications in empirical software engineering—a word of warning », *Empirical Software Engineering*, vol. 13, n° 2, p. 219-221, 2008.
- [127] Kitchenham, B. A. et Charters, S., « Guidelines for performing Systematic Literature Reviews in Software Engineering », Keele University, EBSE Technical Report EBSE-2007-01, 2007.
- [128] Kleppe, A., *Software language engineering: creating domain-specific languages using metamodels*. Addison-Wesley Professional, 2008.
- [129] Kolb, A. Y. et Kolb, D. A., « Learning styles and learning spaces: Enhancing experiential learning in higher education », *Academy of Management Learning & Education*, vol. 4, n° 2, p. 193-212, 2005.
- [130] Kolb, D. A., « Experiential learning theory and the learning style inventory: A reply to Freedman and Stumpf », *Academy of Management Review*, vol. 6, n° 2, p. 289-296, 1981.
- [131] Kolovos, D. S., Paige, R. F., Kelly, T., et Polack, F. A., « Requirements for domain-specific languages », in *Proc. of ECOOP Workshop on Domain-Specific Program Development (DSPD)*, 2006, vol. 2006.
- [132] Kosar, T., Martínez López, P. E., Barrientos, P. A., et Mernik, M., « A preliminary study on various implementation approaches of domain-specific language », *Information and Software Technology*, vol. 50, n° 5, p. 390-405, 2008.
- [133] Kozub, R. M., « An ANOVA Analysis Of The Relationships Between Business Students' Learning », *American Journal of Business Education*, vol. 3, n° 3, p. 89-98, 2010.
- [134] Kuhrmann, M., Méndez Fernández, D., et Tiessler, M., « A mapping study on the feasibility of method engineering », *J. Softw. Evol. and Proc.*, vol. 26, n° 12, p. 1053-1073, 2014.
- [135] Lamkanfi, A. et Demeyer, S., « Filtering Bug Reports for Fix-Time Analysis », in *Proceedings of the 16th European Conf. on Software Maintenance and Reengineering (CSMR'12)*, 2012, p. 379-384.
- [136] van Lamsweerde, A., « Goal-oriented requirements engineering: a guided tour », in *Fifth IEEE International Symposium on Requirements Engineering, 2001. Proceedings*, 2001, p. 249-262.
- [137] Landrum, T. J. et McDuffie, K. A., « Learning Styles in the Age of Differentiated Instruction », *Exceptionality*, vol. 18, n° 1, p. 6-17, 2010.
- [138] Langefors, B., *Theoretical analysis of information systems*. Lund, Sweden: Studentlitteratur, 1968.
- [139] Larousse, *Dictionnaire de Français*, Édition en ligne. 2015.
- [140] Le Moigne, J. L., « La théorie du système d'information organisationnel », *Informatique et gestion*, vol. 102, p. 28-31, 1978.
- [141] Lecourt, D., *La philosophie des sciences*, 5e édition. Paris: Presses Universitaires de France, 2012.
- [142] Lee, A. S., « Researching MIS », in *Rethinking Management Information Systems: An Interdisciplinary Perspective*, W. L. Currie et B. Galliers, Éd. Oxford University Press, 1999, p. 7-27.
- [143] Lemos, J., Alves, C., Duboc, L., et Rodrigues, G. N., « A systematic mapping study on creativity in requirements engineering », in *Proceedings of the 27th Annual ACM Symposium on Applied Computing*, New York, NY, USA, 2012, p. 1083-1088.
- [144] Léonard, M., « Modèle dans le domaine des systèmes d'information », in *Encyclopédie de l'informatique et des systèmes d'information*, J. Akoka et I. Comyn-Wattiau, Éd. Paris: Vuibert, 2006, p. 1396-1411.
- [145] Léonard, M. et Ralyté, J., « From Sustainable Information System with a Farandole of Models to Services », in *Intentional Perspectives on Information Systems Engineering*, S. Nurcan, C. Salinesi, C. Souveyet, et J. Ralyté, Éd. Springer Berlin Heidelberg, 2010, p. 1-15.
- [146] Lessmann, S., Baesens, B., Mues, C., et Pietsch, S., « Benchmarking Classification Models for Software Defect Prediction: A Proposed Framework and Novel Findings », *IEEE Transactions on Software Engineering*, vol. 34, n° 4, p. 485-496, 2008.
- [147] Lingat, J.-Y., « Rubis : un système pour la spécification et le prototypage d'applications bases de données », Thèse de doctorat, Université Paris 6 - UPMC, 1988.
- [148] Linstone, H. A., Turoff, M., et others, *The Delphi method: Techniques and applications*, vol. 29. Addison-Wesley Reading, MA, 1975.
- [149] Liu, Y., Han, S., et Li, H., « Understanding the factors driving m-learning adoption: a literature review », *Campus-Wide Information Systems*, vol. 27, n° 4, p. 210-226, 2010.
- [150] Liu, Y., Li, H., et Carlsson, C., « Factors driving the adoption of m-learning: An empirical study », *Computers & Education*, vol. 55, n° 3, p. 1211-1219, 2010.
- [151] Loo, R., « A Meta-Analytic Examination of Kolb's Learning Style Preferences Among Business Majors », *Journal of Education for Business*, vol. 77, n° 5, p. 252, 2002.

- [152] Lyytinen, K., « Different perspectives on information systems: problems and solutions », *ACM Computing Surveys*, vol. 19, n° 1, p. 5–46, 1987.
- [153] Mayerhofer, T., Langer, P., Wimmer, M., et Kappel, G., « xMOF: Executable DSMLs Based on fUML », in *Software Language Engineering (SLE'13)*, M. Erwig, R. F. Paige, et E. V. Wyk, Éd. Springer International Publishing, 2013, p. 56-75.
- [154] McCandless, M., Hatcher, E., et Gospodnetic, O., *Lucene in Action, Second Edition*. Greenwich, CT, USA: Manning Publications Co., 2010.
- [155] Menzies, T. et Shepperd, M., « Special issue on repeatable results in software engineering prediction », *Empir Software Eng*, vol. 17, n° 1-2, p. 1-17, 2012.
- [156] Mernik, M., Heering, J., et Sloane, A. M., « When and how to develop domain-specific languages », *ACM Comput. Surveys*, vol. 37, n° 4, p. 316–344, 2005.
- [157] Meurant, R. C., « iPad tablet computing to foster Korean EFL digital literacy », *International Journal of u-and e-Service, Science and Technology*, vol. 3, n° 4, p. 49–62, 2010.
- [158] Meyerovich, L. A. et Rabkin, A. S., « Empirical Analysis of Programming Language Adoption », *ACM SIGPLAN Notices - OOPSLA '13*, vol. 48, n° 10, p. 1–18, 2013.
- [159] Miller, J., « Replicating software engineering experiments: a poisoned chalice or the Holy Grail », *Information and Software Technology*, vol. 47, n° 4, p. 233-244, 2005.
- [160] Mohr, A. T., Holtbrügge, D., et Berg, N., « Learning style preferences and the perceived usefulness of e-learning », *Teaching in Higher Education*, vol. 17, n° 3, p. 309-322, 2012.
- [161] Müller, M. et Pfahl, D., « Simulation Methods », in *Guide to Advanced Empirical Software Engineering*, F. Shull, J. Singer, et D. I. K. Sjøberg, Éd. Springer London, 2008, p. 117-152.
- [162] Muller, P.-A., Fleurey, F., et Jézéquel, J.-M., « Weaving Executability into Object-Oriented Meta-languages », in *Model Driven Eng. Languages and Systems (MODELS'05)*, vol. 3713, L. Briand et C. Williams, Éd. Berlin/Heidelberg: Springer, 2005, p. 264-278.
- [163] Ngim, R., « Shared Visions – Yale Students Use iPads to View Live Microscope Images | Yale ITS », *Yale Information Technology Services*, 01-oct-2012. [En ligne]. Disponible sur: <http://its.yale.edu/news/shared-visions-yale-students-use-ipads-view-live-microscope-images>. [Consulté le: 09-oct-2016].
- [164] Niiniluoto, I., « The aim and structure of applied research », *Erkenntnis*, vol. 38, n° 1, p. 1-21, 1993.
- [165] Nurcan, S. et Edme, M., « Intention-driven modeling for flexible workflow applications », *Software Process: Improvement and Practice*, vol. 10, n° 4, p. 363-377, 2005.
- [166] Oliver, R. L., « Cognitive, affective, and attribute bases of the satisfaction response », *Journal of consumer research*, vol. 20, n° 3, p. 418–430, 1993.
- [167] Olle, T., « IFIP TC8 Information Systems », in *The Past and Future of Information Systems: 1976–2006 and Beyond*, vol. 214, D. Avison, S. Elliot, J. Krogstie, et J. Pries-Heje, Éd. Springer Boston, 2006, p. 1-10.
- [168] Olle, T. W., Hagelstein, J., MacDonald, I. G., Rolland, C., Sol, H. G., Van Assche, F. J. M., et Verrijn-Stuart, A. A., *Information Systems Methodologies: A Framework for Understanding*. Addison-Wesley, 1988.
- [169] OMG, « OMG's MetaObject Facility (MOF) », 2015. [En ligne]. Disponible sur: <http://www.omg.org/spec/MOF/2.5/>. [Consulté le: 01-oct-2016].
- [170] OMG, « Semantics of a Foundational Subset for Executable UML Models (FUML), Version 1.0 », 2011. [En ligne]. Disponible sur: <http://www.omg.org/spec/FUML/1.0/>.
- [171] OMG, « Software & Systems Process Engineering Metamodel specification (SPEM) Version 2.0 », 2008. [En ligne]. Disponible sur: <http://www.omg.org/spec/SPEM/2.0/>. [Consulté le: 01-oct-2016].
- [172] Orlikowski, W. J., « The Duality of Technology: Rethinking the Concept of Technology in Organizations », *Organization Science*, vol. 3, n° 3, p. 398-427, 1992.
- [173] Osterweil, L., « Automated support for the enactment of rigorously described software processes », *ACM SIGSOFT Software Engineering Notes*, vol. 14, n° 4, p. 122–125, 1988.
- [174] Paige, R. F. et Varró, D., « Lessons learned from building model-driven development tools », *Software & Systems Modeling*, vol. 11, n° 4, p. 527-539, 2012.
- [175] Panjer, L. D., « Predicting Eclipse Bug Lifetimes », in *Proceedings of the 4th IEEE Working Conf. on Mining Software Repositories (MSR'07)*, 2007, p. 29.
- [176] Park, S. Y., « An Analysis of the Technology Acceptance Model in Understanding University Students' Behavioral Intention to Use E-Learning », *Educational Technology & Society*, vol. 12, n° 3, p. 150–162, 2009.
- [177] Parnas, D. L., « Software engineering programs are not computer science programs », *IEEE Software*, vol. 16, n° 6, p. 19-30, 1999.

- [178] Pascal, A., « Le design science dans le domaine des systèmes d'information : mise en débat et perspectives », *Systèmes d'Information et Management (SIM)*, vol. 17, n° 3, p. 7-31, 2012.
- [179] Pastor, O. et Molina, J. C., *Model-Driven Architecture in Practice: A Software Production Environment Based on Conceptual Modeling*. Springer, 2007.
- [180] Platsidou, M. et Metallidou, P., « Validity and reliability issues of two learning style inventories in a Greek sample: Kolb's Learning Style Inventory and Felder & Soloman's Index of Learning Styles », *International Journal of Teaching and Learning in Higher Education*, vol. 20, n° 3, p. 324-335, 2009.
- [181] Potter, B., Till, D., et Sinclair, J., *An introduction to formal specification and Z*. Prentice Hall PTR, 1996.
- [182] Raja, U., « All complaints are not created equal: text analysis of open source software defect reports », *Empirical Software Engineering*, vol. 18, n° 1, p. 117-138, 2013.
- [183] Ralyté, J. et Rolland, C., « An Assembly Process Model for Method Engineering », in *CAiSE'01*, vol. 2068, K. R. Dittrich, A. Geppert, et M. C. Norrie, Éd. Berlin/Heidelberg: Springer, 2001, p. 267-283.
- [184] Ralyté, J., Rolland, C., et Plihon, V., « Method Enhancement with Scenario Based Techniques », in *CAiSE'99*, vol. 1626, M. Jarke et A. Oberweis, Éd. Berlin/Heidelberg: Springer, 1999, p. 103-118.
- [185] Ramsin, R., « The Engineering of an Object-Oriented Software Development Methodology. », PhD thesis, University of York, York, UK, 2006.
- [186] Rayner, S. et Riding, R., « Towards a Categorisation of Cognitive Styles and Learning Styles », *Educational Psychology*, vol. 17, n° 1-2, p. 5-27, 1997.
- [187] Reynolds, M., « Learning Styles: A Critique », *Management Learning*, vol. 28, n° 2, p. 115-133, 1997.
- [188] Rivera, J. E. et Vallecillo, A., « Adding Behavioral Semantics to Models », in *11th IEEE International Enterprise Distributed Object Computing Conference (EDOC 2007)*, 2007, p. 169-180.
- [189] Robertson, J., « Requirements analysts must also be inventors », *IEEE Software*, vol. 22, n° 1, p. 48, 50-, 2005.
- [190] Robillard, P. N., « Génie logiciel », in *Encyclopédie de l'informatique et des systèmes d'information*, J. Akoka et I. Comyn-Wattiau, Éd. Paris: Vuibert, 2006, p. 1053-1057.
- [191] Roca, J. C., Chiu, C.-M., et Martínez, F. J., « Understanding e-learning continuance intention: An extension of the Technology Acceptance Model », *International Journal of Human-Computer Studies*, vol. 64, n° 8, p. 683-696, 2006.
- [192] Rolland, C., « A primer for method engineering », in *Proceedings of the INFORSID conference*, Toulouse, France, 1997.
- [193] Rolland, C., « Method engineering: towards methods as services », *Software Process: Improvement and Practice*, vol. 14, n° 3, p. 143-164, 2009.
- [194] Rolland, C., Cauvet, C., Nobecourt, P., Proix, C., Collignon, P., Lingat, J.-Y., et Souveyet, C., « The RUBIS system », in *Computerized Assistance During the Information Systems Life Cycle*, A. A. Verrijn-Stuart, T. W. Olle, et L. Bhabuta, Éd. North-Holland Pub. Co., 1988, p. 193-239.
- [195] Rolland, C., Foucault, O., et Benci, G., *Conception des systèmes d'information: la méthode REMORA*. Paris, France: Eyrolles, 1988.
- [196] Rolland, C. et Prakash, N., « From conceptual modelling to requirements engineering », *Annals of Software Engineering*, vol. 10, n° 1-4, p. 151-176, 2000.
- [197] Rolland, C., Prakash, N., et Benjamen, A., « A Multi-Model View of Process Modelling », *Requirements Engineering*, vol. 4, n° 1, p. 169-187, 1999.
- [198] Rolland, C. et Richard, C., « The REMORA Methodologie for Information System Design and Management », in *Information System Design Methodologies: A Comparative Review*, T. W. Olle, H. G. Sol, et A. A. Verrijn-Stuart, Éd. North-Holland, 1982, p. 369-426.
- [199] Rolland, C., Souveyet, C., et Achour, C. B., « Guiding goal modeling using scenarios », *IEEE Transactions on Software Engineering*, vol. 24, n° 12, p. 1055-1071, 1998.
- [200] Rosenthal, R., « Replication in behavioral research », *Journal of Social Behavior & Personality*, vol. 5, n° 4, p. 1-30, 1990.
- [201] Rosenthal, R., « Replication in Behavioral Research », in *Replication Research in the Social Sciences*, J. W. Neuliep, Éd. Newbury Park: SAGE Publications Inc, 1991, p. 1-30.
- [202] Rosenthal, R. et DiMatteo, M., « Meta-analysis: Recent developments in quantitative methods for literature review », *Annual Review of Psychology*, vol. 52, n° 1, 2001.
- [203] Runeson, P., Alexandersson, M., et Nyholm, O., « Detection of Duplicate Defect Reports Using Natural Language Processing », in *Proceedings of the 29th Int'l Conf. on Software Engineering (ICSE 2007)*, 2007, p. 499-510.

- [204] Russo, N., Wynekoop, J., et Walz, D., « The use and adaptation of system development methodologies », in *Managing Information & Communications in a Changing Global Environment*, Idea Group Publishing, PA, Atlanta, USA, 1995.
- [205] Sawilowsky, S. S. et Blair, R. C., « A more realistic look at the robustness and Type II error properties of the t test to departures from population normality », *Psychological Bulletin*, vol. 111, n° 2, p. 352-360, 1992.
- [206] Schlichter, B. R. et Kraemmergaard, P., « A comprehensive literature review of the ERP research field over a decade », *Journal of Enterprise Information Management*, vol. 23, n° 4, p. 486-520, 2010.
- [207] Schmidt, R., Lyytinen, K., Keil, M., et Cule, P., « Identifying Software Project Risks: An International Delphi Study », *Journal of Management Information Systems*, vol. 17, n° 4, p. 5-36, 2001.
- [208] Schmidt, S., « Shall we really do it again? The powerful concept of replication is neglected in the social sciences », *Review of General Psychology*, vol. 13, n° 2, p. 90-100, 2009.
- [209] Seymour, L., Makanya, W., et Berrangé, S., « End-users' acceptance of enterprise resource planning systems: An investigation of antecedents », in *Proceedings of the 6th Annual ISOnEworld Conference*, 2007, p. 26-1.
- [210] Shepperd, M. et Kadoda, G., « Using simulation to evaluate prediction techniques [for software] », in *Proceedings of the 7th Int'l Software Metrics Symposium (METRICS 2001)*, 2001, p. 349-359.
- [211] Shepperd, M. et Schofield, C., « Estimating software project effort using analogies », *IEEE Transactions on Software Engineering*, vol. 23, n° 11, p. 736-743, 1997.
- [212] Shihab, E., Kamei, Y., et Bhattacharya, P., « Mining challenge 2012: The Android platform », in *Proceedings of the 9th IEEE Working Conference on Mining Software Repositories (MSR'12)*, 2012, p. 112-115.
- [213] Shull, F., Carver, J., Vegas, S., et Juristo, N., « The role of replications in Empirical Software Engineering », *Empirical Software Engineering*, vol. 13, n° 2, p. 211-218, 2008.
- [214] Shull, F., Singer, J., et Sjøberg, D. I. K., *Guide to Advanced Empirical Software Engineering*. London, UK: Springer, 2008.
- [215] Shute, V. et Towle, B., « Adaptive E-Learning », *Educational Psychologist*, vol. 38, n° 2, p. 105-114, 2003.
- [216] Simon, H. A., *The Science of the Artificial*, 3rd Edition., vol. 175. Cambridge: MIT Press, 1996.
- [217] Simon, H. A., « The structure of ill-structured problems », in *Models of discovery*, Netherlands: Springer, 1977, p. 304-325.
- [218] Singh, M., « M-learning: A New Approach to Learn Better », *International Journal of Education & Allied Sciences*, vol. 2, n° 2, p. 65-72, 2010.
- [219] Singhal, A., « Modern Information Retrieval: A Brief Overview », *Data Eng Bull*, vol. 24, n° 2, p. 1-9, 2001.
- [220] Sofaer, S., « Qualitative methods: what are they and why use them? », *Health services research*, vol. 34, n° 5 Pt 2, p. 1101-1118, 1999.
- [221] Souag, A., « Etude comparative des formalismes de métamodélisation », Université Paris 1, Mémoire Master Recherche SID, 2010.
- [222] Sprinkle, J., Rumpe, B., Vangheluwe, H., et Karsai, G., « Metamodelling », in *Model-Based Engineering of Embedded Real-Time Systems*, vol. 6100, H. Giese, G. Karsai, E. Lee, B. Rumpe, et B. Schätz, Éd. Berlin/Heidelberg: Springer, 2011, p. 57-76.
- [223] Strate, J. D. et Laplante, P. A., « A Literature Review of Research in Software Defect Reporting », *IEEE Transactions on Reliability*, vol. 62, n° 2, p. 444-454, 2013.
- [224] Tamim, R. M., Bernard, R. M., Borokhovski, E., Abrami, P. C., et Schmid, R. F., « What Forty Years of Research Says About the Impact of Technology on Learning A Second-Order Meta-Analysis and Validation Study », *REVIEW OF EDUCATIONAL RESEARCH*, vol. 81, n° 1, p. 4-28, 2011.
- [225] Tassef, G., « The economic impacts of inadequate infrastructure for software testing », National Institute of Standards and Technology (NIST), USA, RTI Project Number 7007.011, 2002.
- [226] Ulrich, K. T., *Design: Creation of artifacts in society*. University of Pennsylvania, 2011.
- [227] Velez, F., « Proposition d'un environnement logiciel centré processus pour l'ingénierie des systèmes d'information », Thèse de doctorat, Université Paris 1 La Sorbonne, France, 2003.
- [228] Venkatesh, V., Morris, M. G., Davis, G. B., et Davis, F. D., « User acceptance of information technology: Toward a unified view », *MIS Quarterly*, p. 425-478, 2003.
- [229] Venkatesh, V., Thong, J. Y. L., Chan, F. K. Y., Hu, P. J.-H., et Brown, S. A., « Extending the two-stage information systems continuance model: incorporating UTAUT predictors and the role of context », *Information Systems Journal*, vol. 21, n° 6, p. 527-555, 2011.
- [230] Verrijn-Stuart, A., « Themes and Trends in Information Systems: TC8, 1975-1985 », *The Computer Journal*, vol. 30, n° 2, p. 97-109, 1987.

- [231] Vial, S., *Le Design*. Presses Universitaires de France - PUF, 2015.
- [232] Vogelsang, L. et Kensing, F., « Utilizing systems development methods—a conceptual framework », in *Proceedings of the 14th European Conference on Information Systems*, Göteborg, Sweden, 2006, p. 847-858.
- [233] Walker, R. J. et Holmes, R., « Simulation », in *Recommendation Systems in Software Engineering*, M. P. Robillard, W. Maalej, R. J. Walker, et T. Zimmermann, Éd. Springer Berlin Heidelberg, 2014, p. 301-327.
- [234] Wand, Y. et Weber, R., « An ontological model of an information system », *IEEE Transactions on Software Engineering*, vol. 16, n° 11, p. 1282-1292, 1990.
- [235] Wand, Y. et Weber, R., « On the ontological expressiveness of information systems analysis and design grammars », *Information Systems Journal*, vol. 3, n° 4, p. 217-237, 1993.
- [236] Wand, Y. et Weber, R., « Research Commentary: Information Systems and Conceptual Modeling—A Research Agenda », *Information Systems Research*, vol. 13, n° 4, p. 363-376, 2002.
- [237] Wand, Y. et Weber, R., « Toward a theory of the deep structure of information systems », in *ICIS 1990 Proceedings*, 1990.
- [238] Wang, K. h., Wang, T. h., Wang, W. L., et Huang, S. C., « Learning styles and formative assessment strategy: enhancing student achievement in Web-based learning », *Journal of Computer Assisted Learning*, vol. 22, n° 3, p. 207-217, 2006.
- [239] Watson, R. T., « A Personal Perspective on a Conceptual Foundation for Information Systems », *Journal of the Association for Information Systems*, vol. 15, n° 8, p. 514-535, 2014.
- [240] Wegner, P., « Research paradigms in computer science », in *Proceedings of the 2nd Int. Conf. on Software Eng. (ICSE'76)*, 1976, p. 322-330.
- [241] Weick, K. E., « What Theory is Not, Theorizing Is », *Administrative Science Quarterly*, vol. 40, n° 3, p. 385-390, 1995.
- [242] Weiss, C., Premraj, R., Zimmermann, T., et Zeller, A., « How Long Will It Take to Fix This Bug? », in *Proceedings of the 4th Int. Workshop on Mining Software Repositories (MSR'07)*, Washington, D.C., 2007, p. 1.
- [243] Wieringa, R., « Design science as nested problem solving », in *Proceedings of the 4th Int. Conf. on Design Science Research in Inf. Syst. and Technology (DESIST'09)*, New York, NY, USA, 2009, p. 8:1-8:12.
- [244] Wieringa, R. et Daneva, M., « Six strategies for generalizing software engineering theories », *Science of Computer Programming*, vol. 101, p. 136-152, 2015.
- [245] Wieringa, R., Daneva, M., et Condori-Fernandez, N., « The Structure of Design Theories, and an Analysis of their Use in Software Engineering Experiments », in *2011 Int. Symposium on Empirical Software Engineering and Measurement (ESEM)*, 2011, p. 295-304.
- [246] Winskel, G., *The Formal Semantics of Programming Languages: An Introduction*. MIT press, 1993.
- [247] Winter, R., « Design science research in Europe », *European Journal of IS (EJIS)*, vol. 17, n° 5, p. 470-475, 2008.
- [248] Witkin, H. A., Moore, C. A., Goodenough, D. R., et Cox, P. W., « Field-dependent and field-independent cognitive styles and their educational implications », *Review of educational research*, p. 1-64, 1977.
- [249] Wohlin, C., « Guidelines for Snowballing in Systematic Literature Studies and a Replication in Software Engineering », 2014.
- [250] Wohlin, C., Höst, M., et Henningson, K., « Empirical research methods in software engineering », in *Empirical Methods and Studies in Software Engineering: Experiences from ESERNET*, R. Conradi et A. I. Wang, Éd. Berlin/Heidelberg: Springer, 2003, p. 7-23.
- [251] Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., et Wesslén, A., *Experimentation in Software Engineering*. Berlin/Heidelberg: Springer, 2012.
- [252] Yeh, Y., « Aptitude-Treatment Interaction », *Encyclopedia of the Sciences of Learning*. Springer US, p. 295-298, 2012.
- [253] Zaidi, H., « Etude comparative d'outils de méta-modélisation », Université Versailles Saint Quentin en Yvelines, Mémoire Master Recherche COSY, 2010.
- [254] Zelkowitz, M. V. et Wallace, D. R., « Experimental models for validating technology », *Computer*, vol. 31, n° 5, p. 23-31, 1998.
- [255] Zeller, A., *Why Programs Fail: A Guide to Systematic Debugging*, 2nd ed. Morgan Kaufmann Publishers, 2009.
- [256] Zhang, H., Gong, L., et Versteeg, S., « Predicting Bug-fixing Time: An Empirical Study of Commercial Software Projects », in *Proceedings of the 2013 International Conf. on Software Eng. (ICSE'13)*, Piscataway, NJ, USA, 2013, p. 1042-1051.