

HABILITATION A DIRIGER DES RECHERCHES

Spécialité : **Informatique**

Arrêté ministériel : 23 novembre 1988

Présentée par

Agnès FRONT

préparée au sein du **Laboratoire d'Informatique de Grenoble**

**Réutilisation et Socialisation dans
l'Ingénierie des Méthodes**

Habilitation soutenue publiquement le **25 novembre 2011**,

devant le jury composé de :

Mme Brigitte PLATEAU

Professeur, Grenoble-INP, Grenoble, Présidente

Mme Corine CAUVET

Professeur, Université Paul Cézanne, Aix-Marseille, Rapporteur

Mme Camille ROSENTHAL-SABROUX

Professeur, Université Paris-Dauphine, Paris, Rapporteur

Mme Bernadette SHARP

Full professor, Staffordshire University, Stafford, Rapporteur

Mme Florence SEDES

Professeur, Université Toulouse 3, Toulouse, Examineur

Mr Jean-Pierre GIRAUDIN

Professeur, Université Pierre Mendès-France, Grenoble, Examineur

Mme Dominique RIEU

Professeur, Université Pierre Mendès-France, Grenoble, Examineur



Remerciements

Cette Habilitation à Diriger des Recherches représente à peu de choses près treize ans de ma carrière universitaire, et, plus globalement, de ma vie. Treize années riches et bien remplies, mais aussi parfois difficiles et douloureuses... Treize années malgré tout où la vie a fait son chemin, et qui m'ont donné l'occasion de rencontrer et de côtoyer des collègues et des amis que je tiens tout particulièrement à remercier aujourd'hui.

Je souhaite tout d'abord remercier du fond du coeur mes collègues et amis Dominique, plus connue sous le nom de "Dom", et Jean-Pierre, plus connu sous le nom de "JPG", qui m'ont toujours soutenue et encouragée. J'ai beaucoup appris à vos côtés, et c'est certain, sans vous, je ne serais pas là aujourd'hui... Dom, ta philosophie particulière sur la vie m'a presque autant apporté que tes compétences scientifiques et tes qualités professionnelles... Jean-Pierre, je crois que je n'oublierai jamais la première phrase de l'après-INFORSID'2005 : "Bon, maintenant, il faut penser à ton HDR !". J'ai mis un peu de temps, mais voilà, j'y suis arrivée... Merci à vous deux !

Je tiens ensuite à remercier tous les membres de mon jury d'habilitation.

En premier lieu, j'adresse mes plus profonds remerciements à mes trois rapporteurs, Corine Cauvet, Professeur à l'Université Paul-Cézanne, Camille Rosenthal-Sabroux, Professeur à l'Université Paris-Dauphine, et Bernadette Sharp, Professeur à Staffordshire University. Je vous remercie d'avoir pris du temps, sur un emploi du temps que j'imagine chargé, pour lire mon manuscrit et rédiger un rapport. C'est un grand honneur pour moi que vous ayez accepté d'être rapporteurs de mon HDR.

J'adresse aussi mes sincères remerciements à Florence Sèdes, qui me fait le grand plaisir d'avoir tout entrepris pour pouvoir se libérer aujourd'hui, malgré ses lourdes responsabilités au CNRS, à la direction de l'IRIT, au CNU, et j'en oublie certainement... Merci Florence, garde ton sourire et ton accent de Toulouse !

Enfin, je remercie sincèrement Brigitte Plateau d'avoir accepté de présider ce jury. Brigitte, j'ai pu apprécier ton professionnalisme et ton efficacité lorsque tu étais directrice du Laboratoire d'Informatique de Grenoble. Je suis très honorée que tu sois aujourd'hui présidente de mon jury.

Je voudrais ensuite remercier mes collègues et amis de l'équipe SIGMA, sans qui "venir au boulot" serait quand même bien moins sympathique.

Tout d'abord, merci à notre vaillante chef d'équipe Christine. La tâche était lourde de succéder à Jean-Pierre, mais tu t'en sors très bien ! On n'en voudrait pas une autre ! Merci aussi pour nos restos, nos discussions, nos soirées ciné...

Merci ensuite aux trois autres fidèles permanents de l'équipe, Claudia, Sophie et Cyril, pour votre bonne humeur, pour votre amitié, pour vos encouragements permanents, pour les réunions d'équipe remplies de discussions scientifiques et de

bonbons colombiens... Merci aussi à Nadine, ingénieure méthodes et qualité au sein du LIG, pour ta précieuse aide autour des expérimentations liées à ISEA. C'était une bien belle aventure, j'espère qu'il y en aura d'autres...

Je ne manquerais bien sûr pas de remercier les différents étudiants doctorants et stagiaires que j'ai pu encadrer pour des durées plus ou moins longues, et sans qui, chacun à sa manière, je ne serais pas là... Je pense en particulier à "mes" doctorants, Ibtissem, Oualid, Nicolas, Charlotte, Rajaa, Luz-Maria, Marco et Ansem, à "mes" stagiaires CNAM, Bruno, Cathy, Laurent, Manu, Nicolas et Yannick ainsi qu'à tous les autres étudiants que j'ai pu encadrer sur des durées plus courtes, et pour lesquels je suis ravie d'avoir participé à un bout de votre chemin...

Pour clore mes remerciements « côté recherche », j'adresse un grand merci à l'équipe administrative du LIG, en particulier à toutes les secrétaires qui ont participé avec un grand intérêt aux expérimentations autour d'ISEA. Et surtout, tous mes remerciements à Sylvie Dupas, assistante de l'équipe SIGMA, qui a très efficacement mené à bien les aspects administratifs qui ont permis à tous les membres du jury d'être présents aujourd'hui.

Comme pour tout enseignant-chercheur, ma vie est doublement rythmée. Une grande partie de ma vie se fait également au Département Informatique de l'IUT2 de Grenoble, dont je tiens à remercier chaleureusement tous les collègues. C'est un grand plaisir pour moi d'enseigner au sein du Département Informatique où règnent toujours sérieux et professionnalisme dans l'intérêt des étudiants. J'adresse des remerciements particuliers à Françoise et Sophie, pour la joie de partager votre bureau, mais également à Laetitia, Cédric et Sylvie pour nos échanges pédagogiques parfois interminables au sujet du cours d'OMGL-1. Un grand merci également à l'équipe administrative du département, et tout particulièrement à Christine et Caroline pour votre efficacité et votre bonne humeur.

D'un point de vue plus personnel, cette HDR est l'occasion pour moi de remercier tous les amis et proches qui me sont les plus chers.

Tout d'abord, je souhaite remercier les amis qui m'ont supportée (dans tous les sens du terme !) ces treize dernières années, en particulier Cathy et Jean-Phi, Delphine et Thierry, Sylvie et Laurent, Claudia et Yves, Van-Dat, Laurence, Olivier, François et Philippe. Plus particulièrement, je remercie Cathy du fond du coeur : nous nous sommes connus autour d'AGAP, tu m'as souvent écoutée, et ta patience et ton amitié me sont d'un grand secours... Merci d'être là...

Je remercie également ma famille. En particulier mes frères, Michel et Joël, pour votre présence et votre aide quand j'en ai besoin. Mais aussi et surtout ma maman, qui croit en moi bien plus que moi-même.

Enfin, ces treize années, si elles ont été riches sur le plan professionnel, m'ont surtout apporté le grand bonheur d'être maman. Clément, Justine, votre découverte de la vie, vos rires, vos larmes, vos joies, vos peines, sont mon bonheur et ma fierté à la fois. Vous étiez des bébés, vous êtes maintenant des enfants, bientôt des adolescents... Restez comme vous êtes ! Je vous aime...

TABLE DES MATIÈRES

Table des matières

1.Introduction.....	11
2.L'ingénierie de systèmes d'information par réutilisation	15
2.1.Les enjeux liés à la réutilisation.....	17
2.1.1.Documentation complète et unifiée	17
2.1.2.Spécification complète des solutions.....	18
2.1.3.Spécification variable des solutions.....	18
2.1.4.Garde-fous et traçabilité.....	19
2.1.5.Processus et outils de réutilisation adaptés.....	19
2.2.P-SIGMA : un formalisme de représentation unifié	20
2.2.1.Présentation générale du formalisme P-SIGMA.....	20
2.2.1.1.La partie Interface.....	21
2.2.1.2.La partie Réalisation.....	21
2.2.1.3.La partie Relations.....	23
2.2.2.Adaptations de P-SIGMA pour documenter des composants métier.....	24
2.2.3.Adaptations de P-SIGMA pour documenter des frameworks.....	24
2.2.4.AGAP : Un outil de gestion de patrons.....	27
2.3.Complétude et variabilité des spécifications.....	30
2.3.1.Concept de variabilité.....	30
2.3.2.Complétude et variabilité pour les patrons de conception.....	30
2.3.3.Complétude et variabilité pour les composants métier.....	35
2.4.Généricité des spécifications.....	38
2.5.Des processus de réutilisation adaptés.....	40
2.6.Des outils pour supporter les processus de réutilisation.....	42
2.6.1.Un système de recherche de composants.....	42
2.6.2.Les transformations dans les outils IDM	45
2.6.2.1.Transformation de modèles pour la réduction des composants métier.....	45
2.6.2.2.Transformation de modèles pour la réduction des patrons.....	48
2.7.Adaptation de modèles de processus	49
2.7.1.Préambule : la méthode Symphony.....	50
2.7.1.1.Le modèle conceptuel d'objets métier Symphony.....	51
2.7.1.2.Le processus Symphony.....	52
2.7.2.Formalisation d'un processus sous forme de patrons.....	55
2.7.3.Guidage méthodologique pour l'exécution et le suivi du processus.....	57
2.8.Synthèse.....	59
3.L'ingénierie de méthodes.	61
3.1.Une approche par métamodélisation pour la génération de métamodèles de processus spécifiques.....	62
3.1.1.Une méthode pour guider la construction de métamodèles de processus.....	63
3.1.2.Un outil support.....	65

3.2. Une méthode de développement pour les systèmes d'information des organisations ouvertes	67
3.2.1. Cinq aspects pour caractériser le niveau intentionnel	68
3.2.2. L'aspect Identification.....	70
3.3. Une méthode de conception pour des systèmes d'information ubiquitaires personnalisables.....	73
3.3.1. Un domaine d'application privilégié.....	73
3.3.2. Architecture technique visée.....	75
3.3.3. La méthode E-CARE.....	75
3.3.4. Des métamodèles spécifiques.....	76
3.3.5. Les différents prototypes techniques réalisés.....	77
3.3.5.1. Un système d'assistance au voyageur avec Cayuga.....	77
3.3.5.2. Portage sur téléphone mobile.....	79
3.4. Une méthode participative pour la modélisation des processus métier.....	79
3.4.1. La phase de simulation.....	81
3.4.2. Phase d'évaluation.....	83
3.4.3. Phase d'amélioration.....	84
3.4.4. Une méthode outillée.....	84
3.4.5. La construction de la méthode par expérimentations.....	85
3.5. Synthèse.....	87
4. Bilan et perspectives de recherche.....	89
4.1. Bilan.....	89
4.2. Perspectives.....	90
4.2.1. Vers des méthodes de conception de systèmes d'information sociaux.....	91
4.2.2. Vers des méthodes de conception sociales de systèmes d'information.....	92
4.2.3. Vers une ingénierie sociale de méthodes de conception de systèmes d'information.....	95
<i>Bibliographie.....</i>	97
<i>Annexe A : Thèses encadrées.....</i>	113
<i>Annexe B : DEA, M2R et mémoires d'ingénieurs CNAM encadrés.....</i>	115
<i>Annexe C : Participation à des projets.....</i>	117

LISTE DES FIGURES

<i>Figure 1 : Le cube, ou trois critères de classification des patrons.....</i>	<i>17</i>
<i>Figure 2 : AGAP, diagramme simplifié des cas d'utilisation.....</i>	<i>28</i>
<i>Figure 3 : AGAP, création d'un nouveau formalisme.....</i>	<i>29</i>
<i>Figure 4 : AGAP, consultation d'un patron.....</i>	<i>29</i>
<i>Figure 5 : Le métamodèle UML-PI pour l'expression de la variabilité dans les patrons de conception (Arnaud, 2008).....</i>	<i>31</i>
<i>Figure 6 : La solution originale du patron Observateur (Gamma, 1995).....</i>	<i>32</i>
<i>Figure 7 : Vue fonctionnelle avec variantes du patron Observateur.....</i>	<i>33</i>
<i>Figure 8 : Un extrait de la vue dynamique avec variantes du patron Observateur.</i>	<i>34</i>
<i>Figure 9 : Fragments statiques pour le patron Observateur.....</i>	<i>35</i>
<i>Figure 10 : Vue métier du sous-composant « Demande Allocation Abstraite »</i>	<i>37</i>
<i>Figure 11 : Vue fonctionnelle avec variantes du composant métier « Gestion Allocation Ressource ».....</i>	<i>37</i>
<i>Figure 12: Exemple d'un fragment structurel variant.....</i>	<i>38</i>
<i>Figure 13 : Propriétés génériques dans le métamodèle UML-PI.....</i>	<i>39</i>
<i>Figure 14 : Deux instanciations de la métaclasse ClasseImitable.....</i>	<i>39</i>
<i>Figure 15: Processus de réutilisation de composants métier.....</i>	<i>41</i>
<i>Figure 16: Processus d'imitation de composants métier intégré au processus Symphony.....</i>	<i>42</i>
<i>Figure 17 : Capture d'écran de l'AGL ArgoUML comme éditeur de requêtes.....</i>	<i>44</i>
<i>Figure 18 : Capture d'écran du moteur d'évaluation de requêtes.....</i>	<i>45</i>
<i>Figure 19 : Architecture technique du prototype RCMP (Saidi, 2009).....</i>	<i>46</i>
<i>Figure 20 : Réduction d'un composant métier - Approche IDM.....</i>	<i>47</i>
<i>Figure 21 : Extrait de la vue métier après réduction.....</i>	<i>47</i>
<i>Figure 22 : Une réduction de la vue fonctionnelle du patron Observateur.....</i>	<i>48</i>
<i>Figure 23 : Une réduction de la vue dynamique du patron Observateur.....</i>	<i>49</i>
<i>Figure 24 : Une réduction de la vue statique du patron Observateur.....</i>	<i>49</i>
<i>Figure 25 : Exemple de deux objets métier liés par un lien d'utilisation.</i>	<i>52</i>

<i>Figure 26 : Cycle de vie en Y du processus Symphony.....</i>	<i>53</i>
<i>Figure 27 : Processus Symphony, modèle en flocons</i>	<i>54</i>
<i>Figure 28 : Site Web autonome généré à partir d'AGAP.....</i>	<i>58</i>
<i>Figure 29 : Guidage inter-patrons processus</i>	<i>59</i>
<i>Figure 30: Le graphe conceptuel, regroupant les différents types de métamodèles de processus.....</i>	<i>64</i>
<i>Figure 31 : Les principales étapes de la méthode.....</i>	<i>64</i>
<i>Figure 32: L'outil ProMISE, support de la méthode.....</i>	<i>65</i>
<i>Figure 33 : Exemple de métamodèle de processus spécifique.....</i>	<i>66</i>
<i>Figure 34 : Modèle de processus sous forme de diagramme d'objets UML.....</i>	<i>66</i>
<i>Figure 35 : Modèle de processus avec le formalisme KAOS.....</i>	<i>67</i>
<i>Figure 36 : Cinq aspects pertinents pour l'étude d'une Organisation Virtuelle.....</i>	<i>70</i>
<i>Figure 37 : Le modèle pour l'aspect Identification.....</i>	<i>72</i>
<i>Figure 38 : Une instanciation de l'aspect Identification.....</i>	<i>72</i>
<i>Figure 39 : Représentation graphique de l'OV Union des Eleveurs de Tabasco.....</i>	<i>73</i>
<i>Figure 40 : La démarche E-CARE pour la conception de SI ubiquitaires.....</i>	<i>76</i>
<i>Figure 41 : Le métamodèle générique de contexte utilisé dans la méthode E-CARE</i>	<i>77</i>
<i>Figure 42 : Le métamodèle d'événements utilisé dans la méthode E-CARE.....</i>	<i>77</i>
<i>Figure 43 : Modélisation d'un réseau de transport et géolocalisation des systèmes de transport.....</i>	<i>78</i>
<i>Figure 44 : Recalcul d'itinéraire suite à une perturbation.....</i>	<i>78</i>
<i>Figure 45 : Le prototype final sur téléphone mobile, du projet DÉSIT</i>	<i>79</i>
<i>Figure 46 : La méthode ISEA.....</i>	<i>81</i>
<i>Figure 47 : Déroulement du jeu de rôle de simulation - Des activités avec un document.....</i>	<i>82</i>
<i>Figure 48 : Extrait du résultat de la phase de simulation dans le jeu.....</i>	<i>82</i>
<i>Figure 49 : Diagramme d'affinités des actions d'amélioration et diagramme des relations.....</i>	<i>83</i>
<i>Figure 50 : Copie d'écran de l'outil ISEAsy.....</i>	<i>85</i>
<i>Figure 51 : Un participant au serious game et ses activités.....</i>	<i>87</i>
<i>Figure 52 : Chronologie de mes différents thèmes de recherche.....</i>	<i>89</i>
<i>Figure 53 : Evolution des systèmes d'information.....</i>	<i>90</i>

LISTE DES TABLEAUX

<i>Tableau 1 : Rubriques de la partie Interface du formalisme P-SIGMA</i>	<i>21</i>
<i>Tableau 2 : Rubriques de la partie Réalisation du formalisme P-SIGMA.....</i>	<i>22</i>
<i>Tableau 3 : Rubriques de la partie Relations du formalisme P-SIGMA.....</i>	<i>23</i>
<i>Tableau 4 : Patron de documentation du composant métier Client.....</i>	<i>25</i>
<i>Tableau 5 : Patron de documentation d'un composant du framework WASSIT.....</i>	<i>27</i>
<i>Tableau 6 : Patron processus «Phase d'étude préalable » du processus Symphony</i>	<i>57</i>
<i>Tableau 7 : Un échec d'un protocole d'expérimentation pour la construction de la méthode ISEA.....</i>	<i>86</i>

1. INTRODUCTION

Le rôle des systèmes d'information dans le fonctionnement des organisations est de plus en plus crucial, la compétitivité d'une entreprise dépendant souvent directement de la qualité de son système d'information. Plusieurs définitions d'un système d'information ont été données dans la littérature. Dans (Reix, 1998), un système d'information est un ensemble organisé de ressources (matériel, personnel, données, procédures), permettant d'acquérir, de traiter, de stocker, et de communiquer des informations (textes, images, sons, etc.) dans les organisations. Cette définition apparente le système d'information à un système d'information « naturel ». Dans (Nurcan et Rolland, 2005), un système d'information est vu comme le dépositaire du savoir et du savoir-faire des organisations, c'est une base de connaissances de l'entreprise. Le système d'information s'apparente alors plus à un système d'information « artificiel », en particulier logiciel. Finalement, la définition donnée dans Wikipédia résume paradoxalement parfaitement la vision d'un système d'information aujourd'hui, en présentant un système d'information comme **un ensemble de moyens techniques, administratifs, et humains qui servent à la collecte, au classement et à la transmission d'informations entre les membres d'une organisation (institution, entreprise, association, ...)**. Ainsi, le système d'information peut aujourd'hui être vu comme le support des connaissances et des processus métier des organisations. Il se doit alors de supporter, voire de faciliter, les évolutions inéluctables et les changements de tous ordres (techniques, normatifs, institutionnels, etc.).

De manière générale, les travaux de recherche que j'ai effectués ou co-encadrés depuis ma nomination en tant que maître de conférence en 1998 visent à proposer des méthodes permettant de développer et de faire évoluer des systèmes d'information en fonction de l'évolution des besoins et des technologies. De nombreuses définitions d'une méthode d'ingénierie de systèmes d'information existent. Pour Harmsen (Harmsen, 1997), il s'agit d'une « collection de procédures, de techniques, de descriptions de produits et d'outils pour le support effectif, efficace et consistant du processus d'ingénierie d'un système d'information ». Pour Booch (Booch, 1991), une méthode est « un processus rigoureux permettant de générer un ensemble de modèles qui décrit divers aspects d'un logiciel en cours de construction en utilisant une notation bien définie ». Toutes ces définitions mettent en évidence la nécessité d'offrir d'une part des langages permettant de modéliser des produits, d'autre part un processus permettant de faire coopérer différents acteurs, de mobiliser des ressources, et de produire des résultats.

Dans le but de permettre un meilleur partage de la connaissance, j'ai toujours porté un intérêt particulier au développement d'un système d'information **par réutilisation**. Cette problématique de réutilisation était déjà au cœur de mes résultats de thèse de doctorat en 1997 où je proposais un système de patrons proposant un ensemble de solutions à des problèmes récurrents liés à la conception et au développement de systèmes de gestion de bases de données actives (Front, 1997). L'objectif premier du développement d'un système d'information par réutilisation est de permettre de ne plus développer un système d'information « à partir de rien » (« from scratch »), mais

par réutilisation de composants conceptuels et/ou logiciels éprouvés et mis à disposition dans une bibliothèque de composants.

En adéquation avec la définition citée plus haut d'une méthode de développement de SI, la réutilisation des composants conceptuels et logiciels (nous parlerons ici de réutilisation des produits issus du développement), peut tout naturellement être étendue à la réutilisation des processus de développement eux-mêmes. En effet, l'exécution d'un modèle de processus lors de l'ingénierie d'un système d'information correspond à l'utilisation de règles souvent appelées « bonnes pratiques » (Rolland, 2005) pour concevoir un nouveau système d'information. Ces bonnes pratiques sont la plupart du temps informellement schématisées, parfois formalisées sous la forme d'un guide méthodologique, et plus rarement instrumentées. On ne peut alors pas proprement parler, parler de réutilisation de modèles de processus, mais plutôt de l'exécution d'un modèle de processus. Souvent, l'exécution d'un modèle de processus est adaptée au domaine d'application visé, à l'entreprise concernée, ou même au projet lui-même... J'utiliserai ici de préférence le terme « processus adaptable » par rapport à celui de « processus réutilisable » car un processus d'ingénierie est rarement réutilisé tel quel, mais il est le plus souvent adapté au domaine, à l'entreprise, ou au projet (des étapes sont manquées, les acteurs sont agrégés, etc.).

La plupart des travaux de recherche que j'ai menés ou encadrés et qui sont présentés dans la première section de ce mémoire ont ainsi eu pour but d'augmenter la réutilisation de modèles de produits ou de modèles de processus lors du développement d'un système d'information, avec pour objectif les quatre avantages suivants : diminuer les coûts et le temps de conception et de développement, augmenter la qualité et la fiabilité des logiciels produits, réduire les coûts de maintenance et d'évolution des systèmes et enfin permettre une meilleure traçabilité des transformations des produits au sein du processus de développement. L'enjeu ici était lié au fait de rendre ces modèles suffisamment génériques pour qu'ils soient réutilisables ou adaptables à des domaines d'application différents et/ou des technologies variées. Néanmoins, les systèmes d'information visés dans ces travaux restaient traditionnels.

Depuis les années 1970, le rôle du système d'information au sein des organisations a régulièrement évolué. Cette évolution nécessaire pour suivre la complexité croissante des organisations, de leurs processus métier et de leurs acteurs, est la conséquence de l'évolution des objectifs des organisations d'une part, et de l'évolution des technologies d'autre part. Ainsi dans (KLC, 2004), les auteurs identifient trois périodes principales d'évolution : années 1960, années 1970-1980 et années 2000. Les années 1960 sont caractérisées par le début de l'automatisation des activités administratives lourdes, répétitives et régulières. Dans les années 1970, le système d'information est un outil basique au service de toute organisation, il permet la réalisation de processus administratifs simples tels que la gestion des commandes ou la gestion des clients. Un peu plus tard, dans les années 1980-90, le système d'information facilite la réalisation et l'évolution de processus métier complexes. Il peut être utilisé sur la totalité de la chaîne de conception, production, maintenance, vente... Il sert souvent de support pour l'aide à la décision au sein d'une organisation. Désormais, il est au cœur des échanges instantanés entre l'entreprise et l'ensemble de

ses équipes et partenaires, y compris le grand public et permet la maîtrise de processus souples.

Une classification un peu plus fine est celle proposée par S. Nurcan et C. Rolland (Nurcan et Rolland, 2005) qui définissent depuis les années 1960 cinq générations de systèmes d'information : la génération de l'*automation administrative* (années 1960), l'*observatoire du management* (années 1970), les systèmes d'information au service de la *performance opérationnelle* (années 1980), les *infrastructures de coopération et d'ouverture* (années 1990) et enfin, dans les années 2000, les systèmes d'information permettant la *mondialisation* et la *gestion des connaissances*. Cette évolution des systèmes d'information est évidemment liée à l'évolution des technologies (apparition d'Internet, mais aussi des technologies telles que services web, plateformes distribuées, systèmes à base de capteurs, ...), permettant l'interopérabilité entre des systèmes d'information jusque-là incompatibles de différentes organisations et engendrant de fait l'évolution des organisations (des organisations de plus en plus coopérantes, ouvertes, étendues, voire virtuelles).

Les méthodes d'ingénierie traditionnelles n'étant plus adaptées à ces nouveaux types de systèmes d'information, une deuxième partie de mes travaux de recherche ont contribué à développer des méthodes d'ingénierie pour ces nouveaux types de systèmes d'information : il s'agit ici de l'ingénierie de méthodes de développement. Mes contributions en la matière seront présentées en section 3. C'est également dans le domaine de l'ingénierie des méthodes que se situent mes principales perspectives de recherche que je présenterai en section 4.

C'est lors de ma thèse de doctorat, soutenue en décembre 1997, que je découvris l'intérêt de proposer des modèles réutilisables pour l'ingénierie de systèmes d'information. En effet, j'avais constaté que dans de très nombreuses applications de domaines différents existait un besoin récurrent de modéliser des situations déclenchées en réponse à des événements apparaissant dans ou en dehors de l'application. Par exemple, « Comment l'unité médico-technique d'un hôpital doit-elle réagir face à une demande d'acte de la part d'un patient ? », « Que doit faire le système de commande et contrôle d'un navire militaire lors de l'apparition d'une nouvelle menace pour le navire ? », « Comment réorganiser le planning des employés d'une usine lors de l'absence de l'un d'entre eux ? ». De tels besoins comportementaux étaient alors très bien implantés dans des systèmes de gestion de bases de données actives ou des systèmes basés sur la programmation événementielle, mais leur modélisation aux niveaux de l'analyse et de la conception faisait défaut. Je repris alors le concept émergent de patron proposé dans les catalogues de patrons tels que ceux du Gang of Four (Gamma et al., 1995), de P. Coad (Coad, 1995) et de M. Fowler (Fowler, 1997) et proposais le langage de patrons SCalP (Situations Comportementales à l'aide de Patrons) (Front, 1997) (Front, 1999) destiné à guider un concepteur d'applications dans la représentation et la réutilisation de ce que j'appelais à l'époque des « situations comportementales ».

Ces résultats de thèse précurseurs de mes travaux par la suite plus formalisés sur les patrons, m'avaient déjà convaincue de l'intérêt des approches à base de patrons. L'évaluation générale de la technique des patrons que je dressais à l'époque était la suivante :

- l'intégration d'une approche à base de patrons dans une méthode de conception orientée objet classique constitue une forme d'organisation de la modélisation à chaque étape de la conception ;
- les patrons sont des mini-systèmes complets utilisables comme support pédagogique intéressant : en effet, ils représentent des besoins d'applications et offrent des solutions dans un ou plusieurs systèmes cibles. Ils sont illustrés par de nombreux exemples et leur utilisation est constamment justifiée ;
- l'approche des patrons permet de réduire la complexité des systèmes grâce à la définition et à la réutilisation de composants éprouvés de granularité supérieure aux composants de base des méthodes de conception et des systèmes cibles.

Néanmoins, le manque de maturité des approches à base de patrons avait alors engendré un certain nombre de faiblesses dans cette approche. Par exemple, la rubrique *Voir Aussi* mettait en évidence le besoin d'exprimer des relations sémantiquement plus précises entre les différents patrons. D'autre part, la démarche d'utilisation du système de patrons SCalP restait trop informelle pour être réellement contrôlée. Ces difficultés constituaient alors un frein à une bonne réutilisation des patrons proposés dans SCalP ainsi que dans les autres catalogues de patrons et

engendraient plusieurs défis qu'il convenait de résoudre : comment sélectionner un patron ? ; comment utiliser un patron ? (ce que nous appellerons par la suite « imiter ») ; comment combiner l'utilisation de plusieurs patrons ? ... Les travaux de recherche que j'ai menés ont eu pour but d'apporter des éléments de réponse à ces enjeux liés à la réutilisation des patrons et plus généralement de tous les types de composants réutilisables, en particulier des composants métier (Cummins, 1999) (Casanave, 1996) (Guzélian et al., 2004) et des frameworks (Appleton, 1997) auxquels je me suis également intéressée. A ce propos, l'article (Front et al., 1999) définit une classification des différents types de composants réutilisables selon sept critères :

- la nature du problème pour lequel le composant réutilisable apporte une solution : produit ou processus,
- la couverture : les composants peuvent avoir une couverture générique (ils sont alors applicables à tous les domaines), être spécifiques à un domaine métier ou même à une entreprise,
- la portée, ou l'étape d'ingénierie à laquelle ils s'adressent (analyse, conception, implantation, etc.),
- la nature de la solution qu'ils offrent : conceptuelle et/ou logicielle,
- la technique de réutilisation mise en oeuvre : spécialisation, composition, imitation (« copier/coller »), etc.
- leur ouverture : les composants peuvent être de type boîte noire, boîte blanche, ou boîte en verre,
- leur granularité, exprimée généralement en terme de nombre de classes invoquées par la solution.

Ainsi, chaque composant réutilisable peut être classé selon chacun de ces sept critères. En particulier, les patrons offrent la plupart du temps une solution conceptuelle, de type boîte blanche, leur granularité est généralement faible et leur technique de réutilisation est basée sur des compositions d'imitations successives (Front-Conte et al., 2001a) (Arnaud et al., 2005). C'est pourquoi il est possible de restreindre au nombre de trois les critères de classification des patrons : leur nature, leur couverture et leur portée. Ces trois critères combinés permettent à leur tour d'obtenir une classification des patrons présentée dans « le cube » de la Figure 1.

Afin d'être facilement réutilisables, et aussi correctement réutilisés, ces différents types de composants doivent résoudre un certain nombre d'enjeux que je présente dans la suite de cette section : documentation, complétude, variabilité, traçabilité et instrumentation.

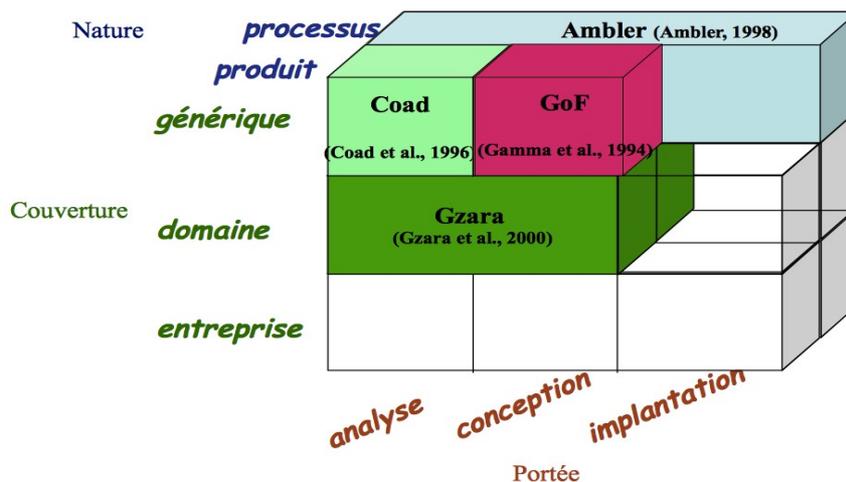


Figure 1 : Le cube, ou trois critères de classification des patrons

2.1. LES ENJEUX LIÉS À LA RÉUTILISATION

Même si leur vocation est de favoriser la réutilisation, patrons, composants métier et frameworks ne sont pas totalement réutilisables de manière satisfaisante et optimale. Ils doivent pour cela avant tout répondre à un certain nombre d'enjeux à mon sens incontournables.

2.1.1. Documentation complète et unifiée

Un composant, pour être réutilisable, doit tout d'abord être documenté de façon précise. Ainsi, quel que soit le type du composant (patron, composant métier, framework), il convient de préciser au minimum le problème auquel il répond, la solution qu'il propose ainsi que les relations qu'il entretient avec les autres composants.

Si le problème de documentation des patrons et des composants métier est relativement bien résolu grâce aux différents formalismes de représentation, il n'en reste pas moins qu'un certain nombre d'inconvénients persistent. D'une part, chaque formalisme possède ses propres rubriques, toutes différentes : 6 rubriques dans le formalisme de P. Coad, 13 dans celui de E. Gamma, 8 dans celui de S.W. Ambler... D'autre part, les formalismes de représentation mettent l'accent soit sur la représentation de solutions modèles (c'est le cas par exemple des formalismes de E. Gamma (Gamma et al., 1995) ou de P. Coad (Coad, 1996) pour les patrons produit ou du modèle conceptuel Symphony pour les composants métier), soit sur la représentation de solutions démarche (c'est le cas par exemple du formalisme de S.W. Ambler (Ambler, 1998) pour les patrons processus). Cela pose trois principaux types de problèmes :

- Dispersion des informations. Les informations utiles pour sélectionner, adapter, composer et organiser des patrons ou des composants métiers sont disséminées dans de nombreuses rubriques.

- Absence de formalisation des rubriques. Les formalismes sont peu adaptés à des techniques automatisées de recherche de patrons et de composants métier selon les besoins du concepteur. Les différentes rubriques sont souvent décrites en langue naturelle sous la forme de textes non structurés, à l'exception des rubriques exprimant les solutions qui font l'objet de formalisation sous forme de diagrammes le plus souvent exprimés en UML.
- Insuffisance des relations inter-patrons ou inter-composants métier. La plupart des formalismes de patrons propose en effet d'exprimer dans une unique rubrique, tous les types de relations entre les patrons d'un ou plusieurs catalogues (par exemple, la rubrique « Combinaisons » du formalisme de P. Coad (Coad, 1996), la rubrique « Patrons apparentés » du formalisme d'E. Gamma (Gamma, 1995) ou encore la rubrique « Patrons liés » de S.W. Ambler (Ambler, 1998)). Le même constat peut être fait pour les composants métier. Il est indispensable que dans un même catalogue, les patrons ou les composants métier ne soient pas simplement organisés dans une seule rubrique, mais soient organisés et hiérarchisés en identifiant différents types de relations inter-patrons ou inter-composants métier (relations de raffinement, d'utilisation, de versionnement...).

Pour pallier à ces inconvénients, j'ai, en collaboration avec d'autres membres de l'équipe SIGMA, proposé le formalisme P-SIGMA qui sera présenté en section 2.2. Nous proposons également de documenter les composants métier et les frameworks sous la forme de patrons exprimés dans un formalisme adapté basé sur P-SIGMA.

2.1.2. Spécification complète des solutions

Une bonne réutilisation passe avant tout par une bonne spécification. Par bonne spécification, nous entendons une spécification complète qui puisse retranscrire tous les apports possibles liés au patron, au composant métier ou au framework en question.

Dans le monde des patrons orientés objets, comme dans celui des composants métier ou des frameworks, l'immense majorité des solutions est constituée uniquement d'un modèle semi-formel statique (diagramme de classes en OMT ou UML, modèle conceptuel Symphony, ...). Le comportement des entités présentes dans cette solution semi-formelle est souvent décrit dans des notes apposées sur le diagramme de classes, parfois décrit de manière textuelle, plus rarement exprimé dans un diagramme de séquences, et quelquefois tout simplement omis par l'ingénieur de patrons. Pourtant, lorsqu'ils développent une application, les ingénieurs d'applications ont besoin d'en spécifier les aspects fonctionnels et comportementaux. C'est pourquoi nous proposons, à l'instar d'un système d'information classique, d'identifier plusieurs vues complémentaires dans la spécification d'un patron, d'un composant métier et d'un framework : une vue fonctionnelle, une vue dynamique et une vue statique.

2.1.3. Spécification variable des solutions

Dans de nombreux cas, par exemple pour les patrons de conception d'E. Gamma, la description est clairsemée d'informations exprimant des variantes possibles de cette

solution, et ceci selon tous les aspects : fonctionnels, dynamiques ou statiques. Qu'elles se situent au niveau de l'implémentation ou bien à un niveau plus fonctionnel, les variantes proposées devraient être systématiquement prises en compte lors de la réutilisation du patron ou du composant métier.

C'est pourquoi nous proposons d'intégrer la notion de variabilité dans les spécifications des patrons ou des composants métier. Les propositions que nous apportons pour rendre des spécifications de patrons, de composants métier ou de frameworks complètes et variables sont présentées en section 2.3. Le noyau dur de ces propositions a été défini dans le cadre de la thèse de Nicolas Arnaud (Arnaud, 2008) pour la spécification des patrons de conception, et complété dans le cadre des composants métier lors de la thèse de Rajaa Saidi (Saidi, 2009).

2.1.4. Garde-fous et tracabilité

Il est fondamental que des règles concernant l'utilisation d'un patron, d'un composant métier ou d'un framework soient précisées afin d'une part de permettre au concepteur d'applications de respecter ce que nous appelons « l'essence » d'un patron, d'un composant métier ou d'un framework (Arnaud et al., 2004), d'autre part de conserver la trace de toutes les utilisations de patrons ou de composants métier. Le concept d'« essence » d'un patron a été défini par (Pagel et Winter, 1996) et s'intitule le « leitmotiv » dans l'approche de (Eden, 2000). Selon le dictionnaire Petit Robert, la définition de l'essence est la suivante : « ce qui fait qu'une chose est ce qu'elle est et sans quoi elle ne serait pas ; ensemble des caractères constitutifs et invariables ».

La question est donc de savoir si les fondements d'un patron, d'un composant métier ou d'un framework sont respectés lors d'une utilisation de ces derniers, et toujours respectés lors de l'évolution ultérieure du système d'information. Ceci revient alors à conserver la trace des utilisations successives de ces composants réutilisables. Nos propositions permettant d'identifier, d'exprimer et de conserver les informations qui expriment l'essence d'un patron ou d'un composant métier et de tracer leur utilisation, sont présentées en section 2.4. Elles ont également été définies dans le cadre de la thèse de Nicolas Arnaud.

2.1.5. Processus et outils de réutilisation adaptés

Fournir des spécifications de patrons, de composants métiers ou de frameworks réutilisables n'a d'intérêt que si l'ingénieur d'applications est guidé dans l'utilisation de ces composants réutilisables. Ce guidage peut prendre plusieurs formes : au minimum, des règles de bonnes pratiques pour réutiliser de façon optimale ces composants réutilisables, et dans certains cas, un processus de réutilisation formalisé et parfois outillé guidant le concepteur dans les différentes étapes à suivre pour réutiliser ces composants. Nous verrons dans la section 2.5 que nous nous attachons à fournir des processus de réutilisation adaptés aux différents types de composants réutilisables proposés. Du point de vue des outils, plusieurs outils de développement à base de composants existent et tentent de mettre en œuvre une aide au processus de réutilisation des composants : les bibliothèques de composants qui ne font principalement que fournir une collection de composants réutilisables, l'approche système de réutilisation qui offre non seulement des fonctions de gestion de

composants mais aussi des mécanismes de manipulation de composants et enfin les environnements de développement par réutilisation qui permettent non seulement la manipulation de composants, mais supportent aussi le processus de réutilisation en guidant dans la sélection des composants et leur adaptation. Nous verrons dans les sections 2.2.4. et 2.6. trois environnements que nous avons proposé pour guider tantôt le concepteur de composants dans la gestion et l'organisation des composants qu'il propose, tantôt le concepteur d'applications lors de la réutilisation de ces composants.

Afin d'apporter des éléments de réponses aux différents enjeux présentés plus haut, je présente dans la suite un ensemble de techniques dont nous avons coordonné des recherches et qui contribuent à l'ingénierie de produits réutilisables, qu'ils soient des patrons, des composants métier ou encore des frameworks.

2.2. P-SIGMA : UN FORMALISME DE REPRÉSENTATION UNIFIÉ

Cette section présente le formalisme de représentation de patrons P-SIGMA proposé dans le cadre d'un travail collaboratif avec Dominique Rieu, Jean-Pierre Giraudin, Mounia Fredj et Ibtissem Hassine. Ce formalisme a fait l'objet de plusieurs publications dont (Front-Conte et al., 2001a) (Front-Conte et al., 2001b) (Front-Conte et al., 2002b) (Front-Conte et al., 2002c).

Initialement destiné à la représentation des patrons en apportant une représentation uniforme des patrons processus et des patrons produit, le formalisme P-SIGMA contribue à l'enjeu de documentation complète et unifiée. De plus, P-SIGMA permet une meilleure formalisation de l'interface de sélection des patrons selon cinq rubriques regroupées dans la partie Interface du formalisme (cf. section 2.2.1.1) et facilitant la sélection d'un patron ou d'un ensemble de patrons. Enfin, P-SIGMA permet une meilleure organisation des catalogues de patrons en proposant d'explicitier les différentes sémantiques de relations entre patrons dans une partie Relations (cf. section 2.2.1.3) afin d'organiser un système de patrons selon des relations clairement établies.

2.2.1. Présentation générale du formalisme P-SIGMA

Le formalisme P-SIGMA est constitué de trois parties : **Interface**, **Réalisation** et **Relations**. La partie Interface contient tous les éléments qui permettent la sélection d'un patron. La partie Réalisation exprime la solution d'un patron en termes de solution modèle et de solution démarche. Enfin, la partie Relations permet d'organiser les relations entre patrons, donc d'organiser les catalogues de patrons.

Chaque partie regroupe un certain nombre de rubriques composées d'un ou de plusieurs champs typés (de type texte, diagramme UML, expression logique de mots-clés, etc.). Toutes les rubriques sont optionnelles, exceptées les rubriques Identifiant, Classification, Contexte et Problème. Dans la suite, chacune de ces rubriques est détaillée et illustrée à l'aide de l'exemple du patron d'architecture MVC.

2.2.1.1. La partie Interface

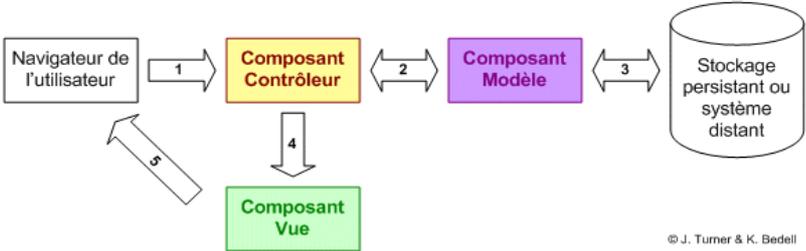
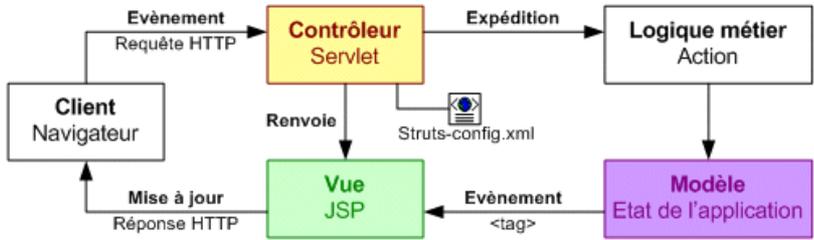
L'interface d'un patron est composée de cinq rubriques servant à la sélection d'un patron. Ces rubriques sont présentées dans le Tableau 1.

Identifiant	Définit le couple problème/solution à partir duquel le patron pourra être référencé. Constitue la clé principale de communication entre le concepteur de patrons et les utilisateurs. Ex : MVC (Modèle Vue Contrôleur)
Classification	Définit la fonction du patron par un ensemble de mots-clés du domaine (termes du domaine). Donne intuitivement la classification du domaine. Ex : Patron produit - Architecture applicative
Contexte	Décrit la pré-condition pour l'application du patron. Peut être obtenu en appliquant la solution modèle d'un ou de plusieurs patrons ; les noms des patrons correspondants constituent alors le champ formel. Ex : il existe une application à développer
Problème	Définit le problème résolu par le patron. Ex : permet de construire des programmes qui gèrent des interactions avec des utilisateurs, implémentent des règles de gestion en fonction des entrées des utilisateurs, et accèdent à des données résidant dans des bases de données ou sur des systèmes distants.
Force	Définit les apports induits par l'application du patron. Ex : Architecture éprouvée. Séparation du code en trois composants distincts : <ul style="list-style-type: none"> • le composant Modèle, qui maintient les données dans un stockage de <i>back-end</i> ou sur un système distant, • le composant Vue, qui constitue la couche présentation destinée à l'utilisateur final, • le composant Contrôleur, qui gère la logique conditionnelle. C'est lui qui décide des écrans qui sont présentés à l'utilisateur, de ce qui se passe lorsqu'une erreur se produit et de la façon et du moment exact où les systèmes distants sont mis à jour.

Tableau 1 : Rubriques de la partie Interface du formalisme P-SIGMA

2.2.1.2. *La partie Réalisation*

La partie Réalisation d'un patron comprend quatre rubriques présentées dans le Tableau 2. Soulignons par ailleurs l'intérêt de la rubrique *Solution démarche* qui permet d'indiquer les cas particuliers de construction de la *Solution modèle*.

<p>Solution Démarche</p>	<p>Indique, éventuellement sous la forme d'un diagramme d'activités UML, la solution du problème en termes de processus à suivre.</p>
<p>Solution Modèle</p>	<p>Décrit la solution en terme des produits attendus après l'application du patron.</p> <p>Ex : (Turner et Bedell, 2003)</p> <ol style="list-style-type: none"> 1. Le navigateur client envoie une requête à l'application. 2. Le composant Contrôleur reçoit la requête. Il commence alors à prendre des décisions sur la façon de continuer, en fonction des règles de gestion qui y sont encodées. 3. Le composant Modèle interagit avec le stockage de données persistant ou le système distant. 4. En fonction des résultats du traitement et des données retournées par le composant Modèle, le Contrôleur détermine le composant Vue qui servira à afficher les données à l'utilisateur. Les données sont préparées pour l'objet Vue, qui les affichera.  <p>5. Le composant Vue choisi génère la réponse à renvoyer à l'utilisateur.</p>
<p>Cas d'application</p>	<p>Décrit des exemples d'imitation de la Solution Modèle.</p> <p>Ex : le framework Struts est fondé sur une architecture MVC (Davis, 2001) :</p> 
<p>Conséquence</p>	<p>Présente les limites et les bénéfices de l'application de la solution.</p>

d'application	<p>Peut inclure un nouvel ensemble de problèmes faisant apparaître la nécessité d'appliquer de nouveaux patrons.</p> <p>Ex : Le MVC peut se révéler lourd à mettre en place, en particulier dans le cas des contrôleurs multiples.</p>
----------------------	--

Tableau 2 : Rubriques de la partie Réalisation du formalisme P-SIGMA

2.2.1.3. La partie Relations

La partie Relations est composée de quatre rubriques correspondant aux quatre types de relations possibles entre les patrons (Rieu et al., 1999). Dans le formalisme P-SIGMA, chaque relation est exprimée par une rubrique mono-champ donnant l'ensemble des patrons qui sont liés au patron en question par la relation concernée : à chaque rubrique est associé un seul champ formel exprimé sous la forme {Patrons}. La signification de chacune de ces relations est exprimée dans le Tableau 3 et s'appuie principalement sur les rubriques de la partie Interface.

Utilise	<p>Si un patron P1 utilise un patron P2, alors :</p> <ul style="list-style-type: none"> - la solution démarche de P1 doit être exprimée en utilisant le patron P2. - la classification de P2 peut être enrichie par rapport à celle de P1 : de nouveaux mots clés sont éventuellement ajoutés dans la classification de P2. - le contexte de P2 peut être enrichi par rapport à celui de P1.
Raffine	<p>Si un patron P1 raffine un patron P2, alors :</p> <ul style="list-style-type: none"> - le problème de P1 doit être une spécialisation de celui de P2. - la classification de P1 peut être enrichie par rapport à celle de P2. - la force de P1 peut être enrichie par rapport à celle de P2. - le contexte de P1 peut être enrichi par rapport à celui de P2.
Requiert	<p>Si un patron P1 requiert un patron P2, alors :</p> <ul style="list-style-type: none"> - l'application de P2 doit être un pré-requis à l'application de P1. - P2 doit apparaître dans le contexte de P1.
Alternative	<p>Un patron P1 est une alternative d'un patron P2 si les deux patrons se différencient par leur force qui justifie deux solutions différentes au même problème :</p> <ul style="list-style-type: none"> - P1 et P2 ont la même classification, le même contexte et le même problème. - seule la rubrique « Force » des deux patrons est différente. <p>Ex : MVC2 est une alternative à MVC : dans MVC2, un seul contrôleur pour simplifier la mise en oeuvre de MVC.</p>

Tableau 3 : Rubriques de la partie Relations du formalisme P-SIGMA

Depuis sa proposition originale en 2001, nous avons largement utilisé P-SIGMA pour décrire tous les systèmes de patrons proposés dans le cadre de thèses ou de projets de l'équipe (projets Initiative Centr'ACTOLL, TRAFIC...). De plus, P-SIGMA a été utilisé par d'autres équipes françaises pour représenter des patrons (Couturier, 2007), (Cocquebert et al., 2010).

2.2.2. Adaptations de P-SIGMA pour documenter des composants métier

Cette section présente l'un des résultats du projet TRAFIC (Transports : Réutilisation, Adaptation, Fiabilité, Intermodalité et Coopération inter-systèmes), projet de la Région Rhône-Alpes, thématiques prioritaires, thème 6 (Aide à la décision : transports), auquel j'ai participé en tant que responsable de 2 lots. L'objectif de ce projet était de définir un cadre méthodologique et une plate-forme expérimentale destinés à capitaliser et à diffuser des connaissances de spécification, de conception et de mise en œuvre de systèmes d'information dédiés en particulier à l'intermodalité des transports dans un contexte régional. Les résultats de ce projet sont présentés sur le site <http://projet-traffic.imag.fr/> et ont fait l'objet d'une publication collective (TRAFIC, 2005). Le projet TRAFIC constitue une suite au projet Initiative Centr'Actoll (2001-2003), projet du ministère de l'Industrie dans le domaine des logiciels de péage du futur, auquel j'avais participé.

Un usage de P-SIGMA inattendu au départ s'est rapidement révélé très pertinent : il s'agit de l'utilisation de P-SIGMA pour décrire des composants métier. En effet, pour être réutilisé, un composant métier doit être documenté le plus précisément possible, et c'est par l'intermédiaire de patrons de documentation que nous avons choisi de documenter les composants métier. Ainsi, dans le cadre du projet TRAFIC, une de nos contributions fut de proposer des composants métier dans le domaine de la billetterie et de documenter ces composants métier dans des patrons de documentation décrits en utilisant une adaptation du formalisme P-SIGMA. Cette solution avait été initialisée par Oualid Khayati dans le cadre de sa thèse (Khayati, 2005) et de ses propositions au sein du projet Initiative Centr'Actoll. L'adaptation du formalisme P-SIGMA adoptée dans le cadre du projet TRAFIC permet de définir pour un composant métier le problème auquel il s'adresse, sa classification (Entité, Processus ou Données de référence), sa solution modèle, les composants métier qu'il utilise, les composants métier par qui il est utilisé et sa solution modèle dans le modèle conceptuel Symphony (Hassine et al., 2005a) (cf. section 2.7.1.). Le Tableau 4 donne un exemple du patron de description du composant métier Client défini dans le cadre du projet TRAFIC.

2.2.3. Adaptations de P-SIGMA pour documenter des frameworks

Cette section présente l'un des résultats de la thèse d'Ahmed Zellou que j'ai co-encadrée avec D. Chiadmi dans le cadre du projet franco-marocain STIC-GL. Cette thèse a été soutenue à Rabat (Maroc) en avril 2008 (Zellou, 2008) et a fait l'objet des publications (Zellou et al., 2004) et (Zellou et al., 2006).

L'objectif principal de la thèse de A. Zellou était de proposer l'architecture du framework de composants WASSIT dans le domaine de la médiation d'information. Dans le but de documenter ce framework afin qu'il soit efficacement compris et réutilisé, chaque composant de ce framework a été décrit par un patron de documentation dans un formalisme adapté de P-SIGMA, le formalisme C-SIR. Outre les rubriques *Identifiant*, *Problème* et *Cas d'application*, le formalisme C-SIR contient des rubriques spécifiques à la description des modules d'un framework : *Informations internes*, *Informations d'entrée*, *Informations de sorties*, *Formalisation* (en réalité, la solution du composant) et *Contraintes*. Le Tableau 5 décrit le patron de documentation du module de réécriture du framework WASSIT (Zellou, 2008).

Identifiant	Patron de documentation du composant métier Client
Problème	Personne ayant ou ayant eu des liens avec la société de transport
Solution Modèle	<p>Created with Poseidon for UML Community Edition. Not for Commercial Use.</p> <ul style="list-style-type: none"> - Client : la classe maître du composant Client, contient les informations nécessaires à l'identification d'un client : ID, Nom, Prénom, Date_De_Naissance, Lieu_De_Naissance. - Statut : une classe partie du composant Client qui gère le statut du client en cours (jeune, handicapé, etc.) pour des réductions ou des offres personnalisées.
Utilise	<ul style="list-style-type: none"> - Titre : il s'agit des différents titres de transport que le client a achetés. - Historisation_Client : c'est un rôle du composant Historisation qui sert à mettre en mémoire les événements de création ou de changement d'état d'un client.
Utilisé par	Le composant Client est utilisé par le composant Usager via le rôle Payeur. Cela désigne les clients payant pour un usager.

Tableau 4 : Patron de documentation du composant métier Client

Identifiant	Composant de réécriture du framework WASSIT
Problème	Réécrire une requête d'entrée formulée en termes d'un schéma global sous forme d'une ou de plusieurs sous-requêtes formulées en termes d'un ensemble de schémas locaux.
Informations internes	Un schéma global, un ensemble de schémas locaux et la liaison entre le schéma global et les schémas locaux.
Informations d'entrée	Une requête formulée en termes du schéma global.
Informations de sorties	Un ensemble de requêtes formulées en termes des schémas locaux.
Cas d'application	<p>Supposons que nous disposions de trois sources différentes : Edition, Auteur et Livre qui publient des informations sur les livres, leurs auteurs et leurs sociétés d'édition.</p> <p>$S_1 : Edition(ISBN, éditeur, année)$ $S_2 : Auteur(auteur, pays, @électronique)$ $S_3 : Livre(titre, auteur, ISBN, nb_pages, dim)$</p> <p>Supposons que le schéma global contienne les relations suivantes :</p> <p>$R_1(titre, auteur, pays, ISBN)$ $R_2(ISBN, éditeur, dim)$</p> <p>Supposons maintenant que la requête d'entrée (exprimée en termes de R_1 et R_2) demande la liste des auteurs français des livres édités par la société Hermès. La requête d'entrée à la forme :</p> <p>$Q_{entrée}(auteur) :- R_1(titre, auteur, pays, ISBN), R_2(ISBN, éditeur, dim), pays = 'France', éditeur = 'Hermès'$</p> <p>Le module de réécriture réécrit la requête en utilisant les trois relations Edition, Auteur et Livre. Il obtient les trois sous-requêtes Q_1, Q_2 et Q_3 destinées respectivement aux sources S_1, S_2 et S_3.</p> <p>$Q_1(ISBN) :- Edition(ISBN, éditeur, année), éditeur = 'Hermès'$ $Q_2(auteur) :- Auteur(auteur, pays, @électronique), pays = 'France'$ $Q_3(auteur, ISBN) :- Livre(titre, auteur, ISBN, nb_pages, dim).$</p> <p>$Q_{retour}(auteur) :- Q_2(auteur), Q_1.ISBN = Q_3.ISBN, Q_2.auteur = Q_3.auteur$</p> <p>Cette réécriture permet d'obtenir la requête de sortie suivante :</p> <p>$Q_{sortie}(auteur) :- Edition(ISBN, éditeur, année), Auteur(auteur, pays, @électronique), Livre(titre, auteur, ISBN, nb_pages, dim), pays = 'France', éditeur = 'Hermès'$</p>
Formalisation	<p>Schéma : Un schéma est un ensemble de relations R_{ij} pour $i=1, \dots, n$. $j=1, \dots, X_i$.</p> <p>Schéma des sources : Un système de médiation est composé d'un ensemble de sources (n sources). Chaque source S_i est décrite avec un schéma local $SchL_i$ décrivant son contenu pour $i=1, \dots, n$. Chaque schéma local $SchL_i$ contient un ensemble de relations liées par une fonction F_i pour $i=1, \dots, n$.</p>

	<p>$SchL_i = F_i (R_{i,j})_{j=1,\dots,x_i}$; où $x_i \in \mathbb{N} \forall i=1,\dots,n$.</p> <p>où F_i est une fonction qui implique les sous-relations $R_{i,j}$.</p> <p>Une relation a la forme $R_{i,j} (X_{i,j,1}; X_{i,j,2}; \dots, X_{i,j,t(i,j)})$ où $t(i,j) \in \mathbb{N}$; $X_{i,j,k}$ est une variable.</p> <p>S est le schéma des sources : $S = F_s (S_i)_{i=1,\dots,n}$.</p> <p>$S = F_s (R_{i,j})_{i=1,\dots,n ; j=1,\dots,x_i}$.</p> <p>où F_s est une fonction qui implique l'ensemble des sous-relations $R_{i,j}$ des sources.</p> <p>Schéma global : Un schéma global R_g est un ensemble de relations $R_{gav,i}$ pour $i=1,\dots,m$.</p> <p>$R_{gav,i} (Y_{i,1}; Y_{i,2}; \dots, Y_{i,u(i)})$ où $Y_{i,u(i)}$ est une variable et $u(i) \in \mathbb{N}$.</p> <p>$R_g = F_g (Y_{i,j})_{i=1,\dots,m ; j=1,\dots,u(i)}$,</p> <p>où F_g est une fonction qui implique les relations du schéma global.</p> <p><i>Inclusion de relations</i> : une relation $R_{i,j}$ est contenue dans une autre relation $R_{k,l}$ notée ($R_{i,j} \subseteq R_{k,l}$) ssi pour chaque variable X tel que $X \in R_{i,j}$, alors $X \in R_{k,l}$. Les deux relations sont équivalentes si ($R_{i,j} \subseteq R_{k,l}$) et ($R_{k,l} \subseteq R_{i,j}$).</p> <p><i>Vue totale et vue partielle</i> : le système de médiation possède une vue totale sur les sources si $S = R_g$; le système possède une vue partielle sur les sources si $R_g \subseteq S$.</p>
Contraintes	<p>La jointure des requêtes de sortie doit être, soit une réécriture équivalente, soit une réécriture au contenu maximal de la requête d'entrée.</p>

Tableau 5 : Patron de documentation d'un composant du framework WASSIT

2.2.4. AGAP : Un outil de gestion de patrons

La spécification de l'atelier AGAP présenté ici est le fruit d'une collaboration avec plusieurs membres de l'équipe SIGMA (en particulier Dominique Rieu, Jean-Pierre Giraudin, Mounia Fredj, Ibtissem Hassine). Son développement a été effectué par plusieurs stagiaires et ingénieurs CNAM que j'ai encadrés ou co-encadrés (Stéphanie Pognat, Cathy Descombes, Laurent Tastet, Emmanuel Jausseran, Thibaut Montanier et Wouter Debie). Cet outil est accessible à l'adresse <http://sigmacomp.imag.fr:8080/AGAPDIST> et a fait l'objet de plusieurs publications dont (Front-Conte et al., 2001c) et (Front-Conte et al., 2002a).

L'atelier AGAP a été développé conjointement à la spécification du formalisme P-SIGMA. A l'origine, l'un de ses principaux buts était de pouvoir fournir à un ingénieur de patrons la possibilité de saisir et d'organiser des patrons en les documentant à l'aide du formalisme P-SIGMA ou d'une de ses adaptations. Conçu de manière générique, il a très vite proposé deux modules : un **module par la réutilisation (« by reuse »)** destiné à l'**ingénieur d'applications** et centré sur la sélection, l'imitation, l'adaptation et l'intégration des patrons dans des systèmes d'information, mais surtout un **module pour la réutilisation (« for reuse »)** destiné à l'**ingénieur de patrons** et centré sur l'identification, la spécification, et l'organisation des patrons. En ce sens, il diffère de

la plupart des outils de génie logiciel existants (citons en particulier les ateliers Objecteering, Poséidon, VP-UML, Rational Rose...), relativement « figés », le but principal étant de permettre la réutilisation et l'intégration dans le code de patrons existants tels que ceux du catalogue du GoF ou d'autres patrons plus spécifiques (patrons pour J2EE, patrons pour EJB, etc.).

AGAP dispose ainsi des fonctionnalités illustrées par le diagramme de cas d'utilisation UML simplifié de la Figure 2 où les cas d'utilisation commençant par « Gérer ... » englobent les sous-cas d'utilisation « Créer ... », « Modifier ... », « Supprimer ... », « Visualiser ... », « Valider ... » et « Enrichir ... ».

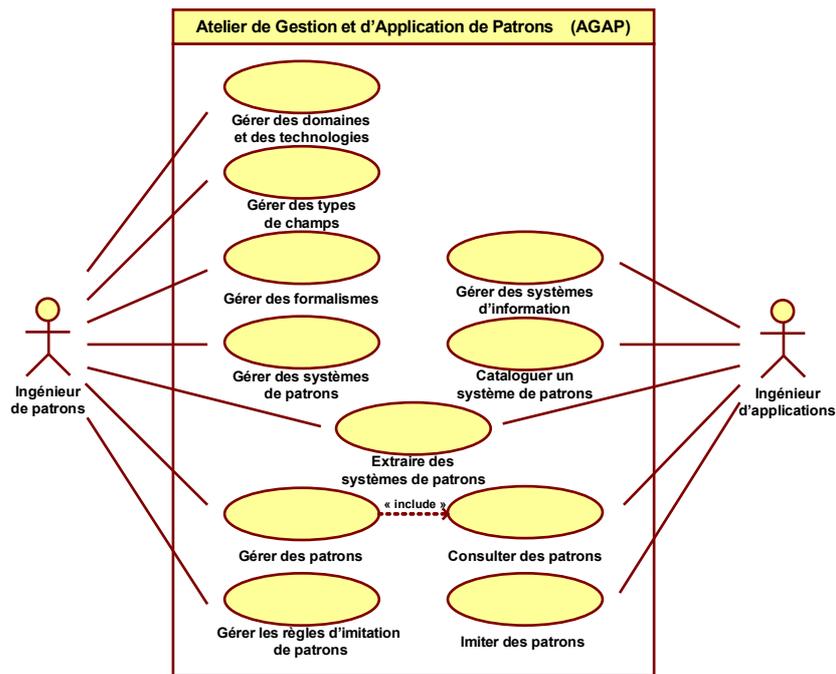


Figure 2 : AGAP, diagramme simplifié des cas d'utilisation

L'atelier AGAP permet ainsi de gérer simultanément plusieurs formalismes et systèmes de patrons. Il a été développé suivant une architecture technique trois tiers MVC2 (Modèle Vue Contrôleur) par l'intermédiaire du framework Struts. La Figure 3 présente un exemple d'utilisation d'AGAP, en ingénierie des patrons, avec la création d'un nouveau formalisme et notamment l'utilisation d'une rubrique déjà existante (rubrique Contexte du formalisme P-SIGMA en l'occurrence).

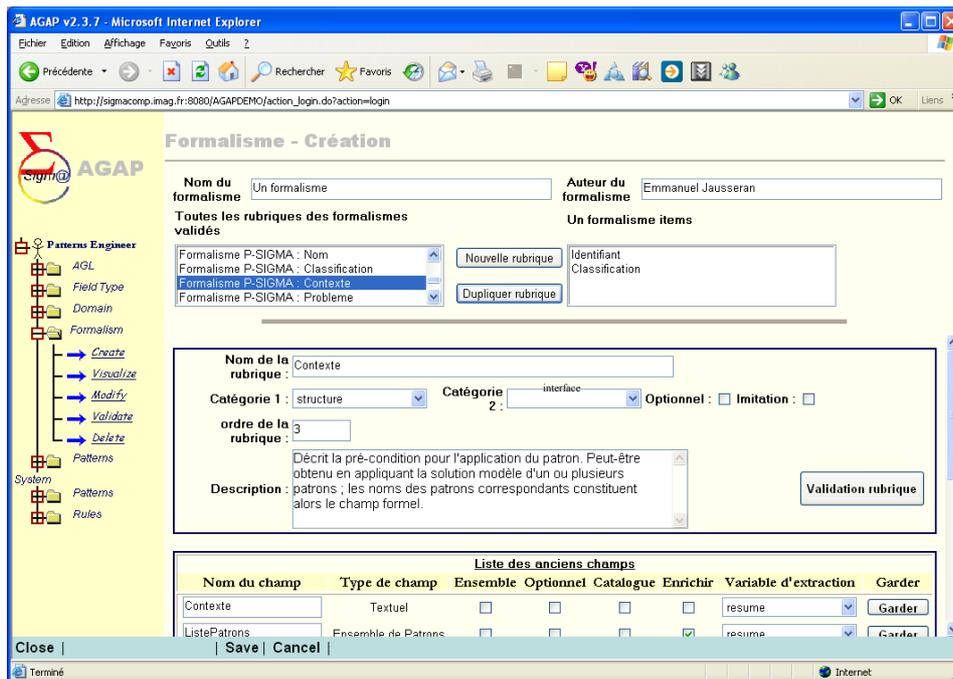


Figure 3 : AGAP, création d'un nouveau formalisme

La Figure 4 présente un exemple d'utilisation d'AGAP, en ingénierie des applications, avec la consultation de la fiche du patron Composite de Gamma (Gamma et al., 1995).

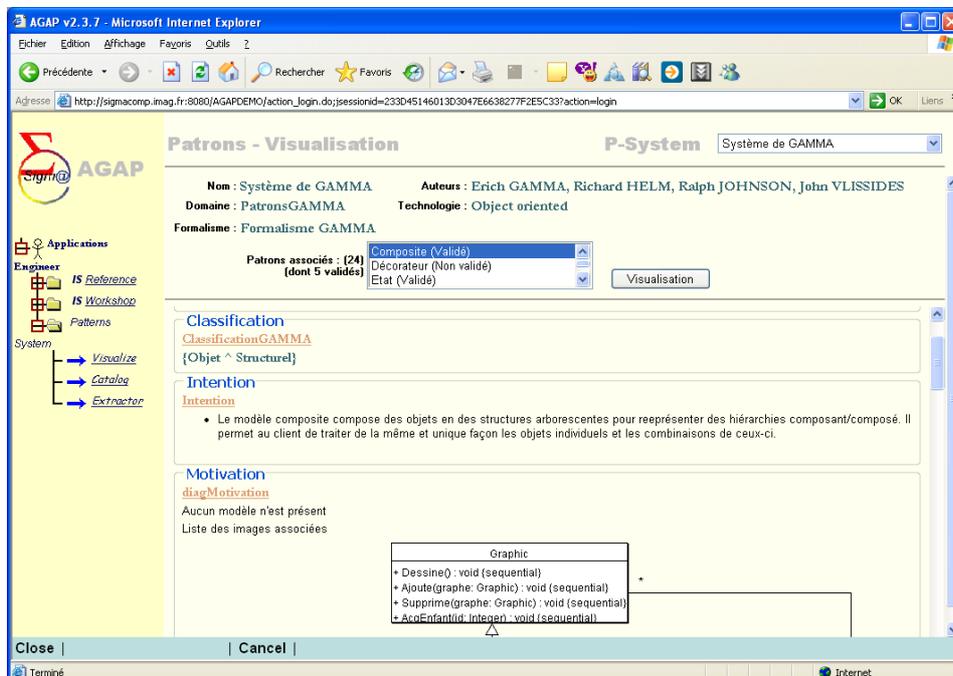


Figure 4 : AGAP, consultation d'un patron

2.3. COMPLÉTUDE ET VARIABILITÉ DES SPÉCIFICATIONS

Outre une bonne documentation, une réutilisation efficace des patrons et des composants métier passe avant tout par une spécification complète. Nous proposons de spécifier les patrons et les composants métier dans des mini-systèmes complets intégrant à l'instar des systèmes d'information traditionnels, plusieurs vues de développement (une vue fonctionnelle, une vue dynamique et une vue statique pour les patrons de conception, complétées par une vue métier pour les composants métier). De plus, chacune de ces vues prend en compte les aspects variables des différentes solutions possibles. Pour ce faire, nous proposons des extensions du métamodèle UML intégrant des concepts de variabilité dans les quatre diagrammes suivants : le diagramme d'activités pour la vue métier (composants métier uniquement), le diagramme des cas d'utilisation pour la vue fonctionnelle, le diagramme de séquence pour la vue dynamique et le diagramme de classes pour la vue statique.

2.3.1. Concept de variabilité

La variabilité est définie comme la capacité d'un système à être changé, personnalisé et configuré en fonction d'un contexte spécifique (Van Grup, 2000). Un point de variation est un endroit du système où il y a une variation (Czarnecki et Eisenecker, 2000), c'est-à-dire où des choix devront être faits afin d'identifier les variantes à utiliser. Plusieurs types de variabilité existent pour un point de variation (Bachmann et al., 2001) : les options (choix de zéro ou plusieurs variantes parmi plusieurs), les alternatives (choix d'une variante parmi plusieurs,) les alternatives optionnelles (choix de zéro ou une variante parmi plusieurs) et les ensembles d'alternatives (choix d'au moins une variante parmi plusieurs).

De nombreuses approches ont été proposées pour représenter la variabilité dans les spécifications : les diagrammes de « features » de FODA (Kang et al., 1990), les diagrammes de cas d'utilisation de (VanDerMaßen et Lichter, 2002), les diagrammes de classes de (Clauss, 2001), les diagrammes de séquences de (Ziadi et Jézéquel, 2005), etc. Comme les deux dernières approches, nous proposons d'intégrer la variabilité dans une extension d'UML à l'aide, entre autres, de stéréotypes.

2.3.2. Complétude et variabilité pour les patrons de conception

Ce travail a été réalisé dans le cadre de la thèse de Nicolas Arnaud, que j'ai co-encadrée avec Dominique Rieu. Cette thèse a été soutenue en octobre 2008 (Arnaud, 2008) et a fait l'objet de plusieurs publications dont (Arnaud et al., 2009) (Arnaud et al., 2007a) et (Arnaud et al., 2006).

Dans le métamodèle UML-PI (cf. Figure 5), les principes de la variabilité ont été utilisés dans le cadre de l'expression de la solution des patrons de conception. Ainsi, le métamodèle UML-PI est une extension du métamodèle UML dédiée aux spécifications avec variantes des patrons. La Figure 5 montre un extrait de ce nouveau métamodèle, où la variabilité est illustrée ici au sein de la vue fonctionnelle (les éléments en gris sont les ajouts). Une *Solution* est composée de plusieurs fonctionnalités appelées *CasUtilisationImitable*. Un *CasUtilisationImitable* est un

UseCase classique qui comporte un *FragmentStatique* et un *FragmentDynamique* qui est une *Interaction*. Les points de variation et variantes sont respectivement représentés par les métaclasse *VariationPoint* et *Variant*. Les cardinalités d'un point de variation, qui définissent le type de variabilité, sont représentées comme des attributs de *VariationPoint*. La *Racine* est un point de variation de type « option » de cardinalité 0..*.

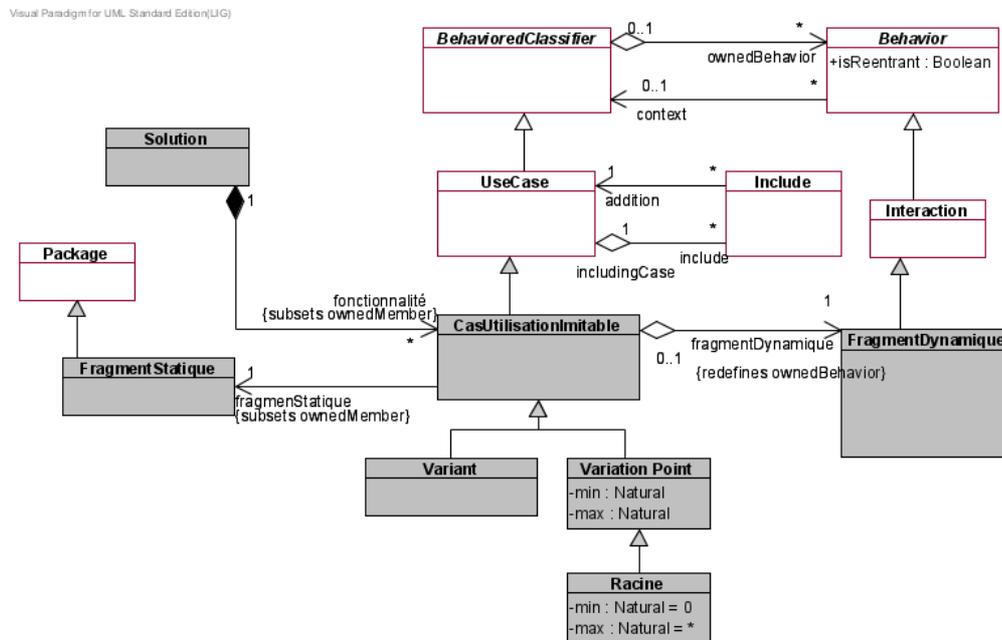


Figure 5 : Le métamodèle UML-PI pour l'expression de la variabilité dans les patrons de conception (Arnaud, 2008)

Comme illustration, nous nous appuyons sur l'exemple du patron *Observateur* défini dans (Gamma et al., 1995) et que nous rappelons dans la Figure 6.

D'une part, les auteurs précisent dans la rubrique Implémentation que le protocole de mise à jour pourrait être différent de celui nominale spécifié, en donnant directement, lors de l'appel à la méthode *MiseAJour()*, les informations nécessaires aux observateurs. Ce protocole implique une spécification différente de la solution, et en particulier au niveau de la vue dynamique, puisque l'appel à la méthode *AcqEtat()* n'aurait plus sens. Cette variante est en fait une variante d'implémentation du patron, puisque du point de vue du *Client*, rien n'est différent.

D'autre part, du point de vue du déclenchement de la mise à jour, la notification des observateurs est systématiquement exécutée dans la solution nominale du patron après le changement d'état du *Sujet*. Ce déclenchement « implicite » peut être remplacé par un déclenchement commandé par le *Client*.

Enfin, contrairement à l'intention précisée dans (Gamma et al., 1995), la solution du patron *Observateur* ne se limite pas à mettre en œuvre la notification des observateurs mais permet également d'attacher ou de détacher ces derniers à un sujet.

La solution devrait donc introduire le fait que la gestion des observateurs ne constitue qu'une fonctionnalité secondaire qui peut ne pas être retenue lors de l'imitation.

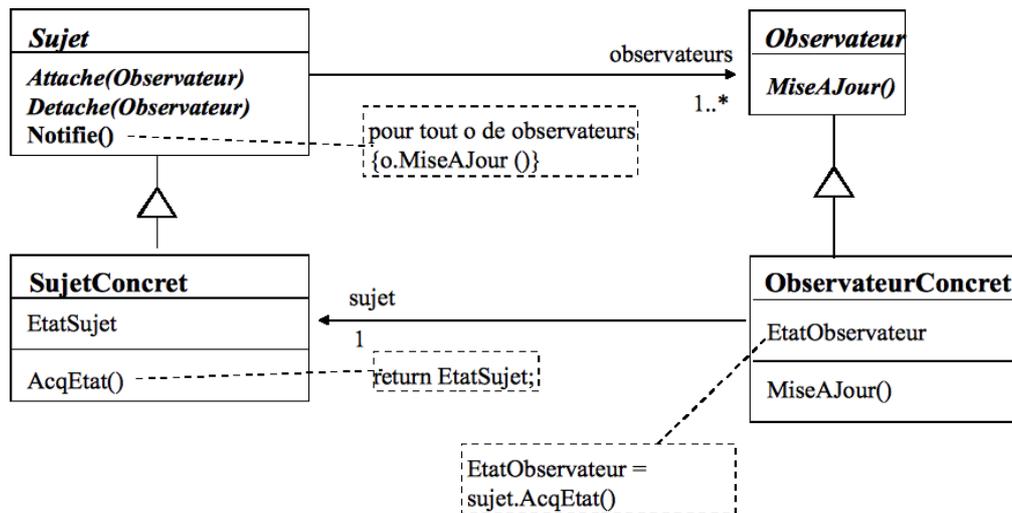


Figure 6 : La solution originale du patron Observateur (Gamma, 1995)

Vue fonctionnelle avec variantes

Conformément aux principes de la variabilité, les trois aspects évoqués précédemment décrivent en fait trois points de variation qu'il s'agit de définir au sein de la solution du patron. Nous proposons de les représenter dans la vue fonctionnelle avec variantes du patron *Observateur* (cf. Figure 7) où le cas d'utilisation abstrait racine *Observateur* permet de sélectionner 0 ou plusieurs cas d'utilisation parmi les deux cas d'utilisation *Modifier le sujet* et *Gérer les observateurs*. Le cas d'utilisation *Modifier le sujet* est obligatoire (non stéréotypé). Par contre, le cas d'utilisation *Gérer les observateurs* est optionnel (stéréotypé « variant »), il pourra donc ne pas être choisi par le concepteur. Cette distinction entre cas d'utilisation obligatoire et cas d'utilisation optionnel permet entre autres d'exprimer l'essence du patron *Observateur*.

D'autre part, le cas d'utilisation *Modifier le sujet* inclut le cas d'utilisation *Notifier les observateurs*, qui est un point de variation (stéréotypé *Variation*) incluant 2 variantes : *Notifier implicitement* et *Notifier explicitement*. Les cardinalités 1..1 permettent de préciser que le concepteur d'applications devra choisir un et un seul parmi ces deux cas d'utilisation variants.

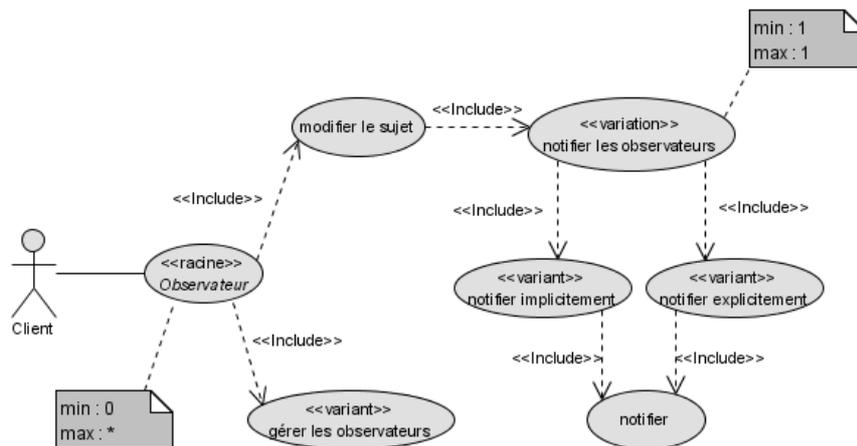


Figure 7 : Vue fonctionnelle avec variantes du patron Observateur

Vue dynamique avec variantes

Dans la spécification d'un patron comme dans tout système d'information, la vue dynamique a pour objectif de préciser la réalisation des éléments de la vue fonctionnelle au sein du déroulement des scénarii. Dans un contexte variable, la vue dynamique se doit de préciser comment les comportements associés aux variantes sont organisés chronologiquement et comment ils s'enchaînent. La vue dynamique est donc décomposée selon les cas d'utilisation et doit permettre de définir le comportement des variantes, d'exprimer une contrainte chronologique entre les variantes d'un même point de variation ou de définir l'impact du point de variation sur le scénario dans lequel il intervient.

Pour l'expression de la vue dynamique, notre approche s'appuie sur l'utilisation de fragments combinés (*CombinedFragment*). Un fragment combiné est un type de fragment d'interaction UML2 qui comporte un opérateur ainsi qu'un ou plusieurs opérandes. Plusieurs opérateurs sont disponibles, parmi lesquels l'opérateur *loop* qui permet d'exprimer une boucle, l'opérateur *opt* qui permet de spécifier qu'un fragment d'interaction est optionnel, l'opérateur *alt* destiné à la définition de structures de choix conditionnels entre plusieurs fragments, l'opérateur *seq* qui indique que différents opérandes doivent être exécutés en séquence, mais dans n'importe quel ordre et enfin l'opérateur *strict* qui exprime lui aussi une séquence, mais selon l'ordonnancement donné.

Plusieurs règles permettent, à partir d'une vue fonctionnelle avec variantes, de construire la vue dynamique avec variantes. Ainsi par exemple, la Figure 8 montre une partie de la vue dynamique avec variantes pour le patron *Observateur*. On y retrouve un fragment d'interaction pour le point de variation *Modifier le sujet* (partie invariable) ainsi qu'un fragment d'interaction par variante. La séquence de la méthode *notifier()* est la même pour les deux variantes.

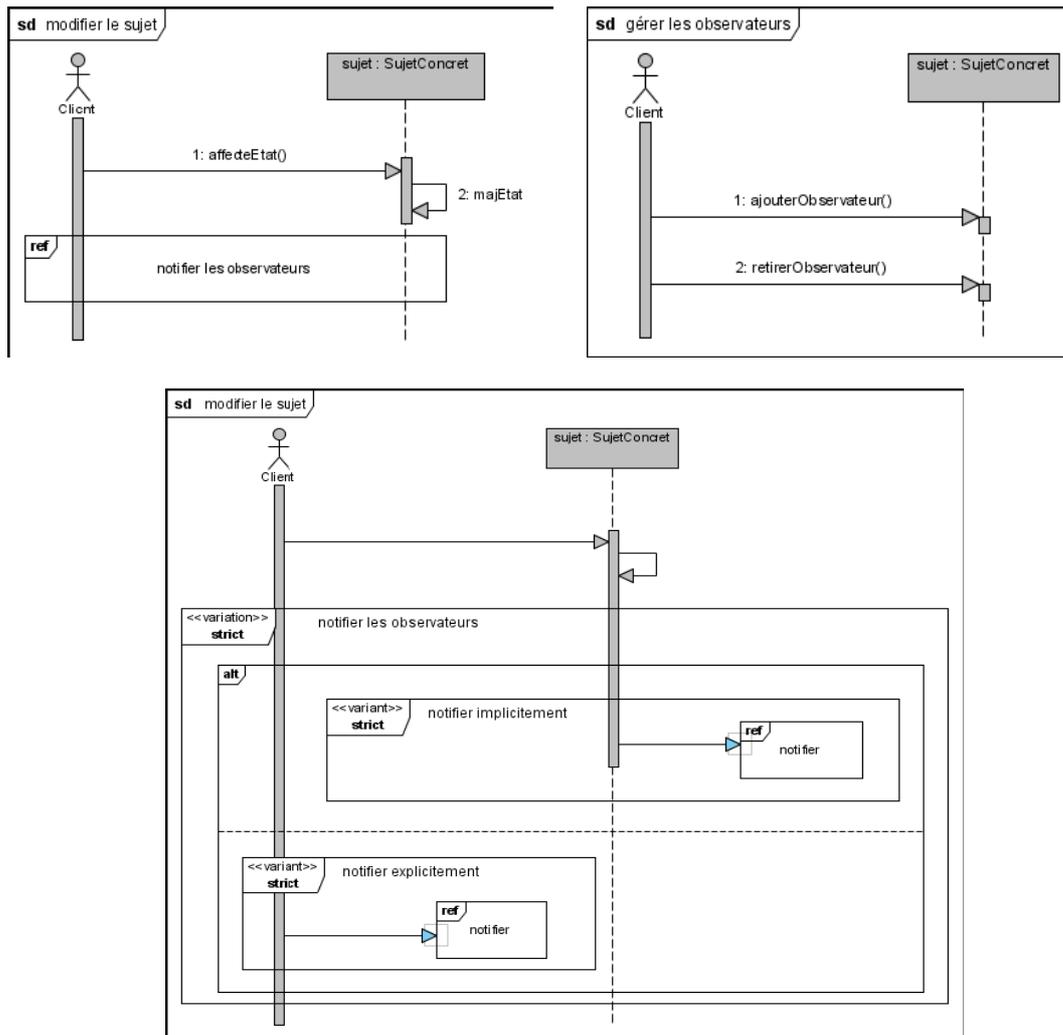


Figure 8 : Un extrait de la vue dynamique avec variantes du patron *Observateur*

Vue statique avec variantes

Enfin, la vue statique est en grande partie déductible de la vue dynamique : chaque cas d'utilisation donne lieu à un fragment statique cohérent avec l'interaction définie dans la vue dynamique. Un fragment statique est un paquetage UML2 stéréotypé « *fragment statique* ». La vue statique variable est composée de plusieurs fragments statiques qui s'assemblent lors de l'imitation grâce à l'opérateur *merge* d'UML2 pour former une vue statique classique.

La Figure 9 représente la vue statique avec variantes du patron *Observateur*. Dans l'interaction *Modifier le sujet*, seule la classe *SujetConcret* est utilisée par l'intermédiaire d'une ligne de vie et des méthodes *affecteEtat()* et *majEtat()* qu'elle comporte. En suivant les recommandations ci-dessus, le fragment statique de *Notifier les observateurs* devrait être vide, puisque l'interaction ne comporte aucun message propre. Toutefois, nous préconisons la généralisation des propriétés qui sont communes, y compris par inclusion, à toutes les variantes d'un même point de variation. D'un point de vue statique, la seule différence entre les deux variantes de

notification est la visibilité de la méthode *notifier()*. Tous les autres éléments (associations, classe *Observateur*, classe *ObservateurConcret*, ...) sont communs, c'est pourquoi ils sont présents dans le fragment statique du point de variation.

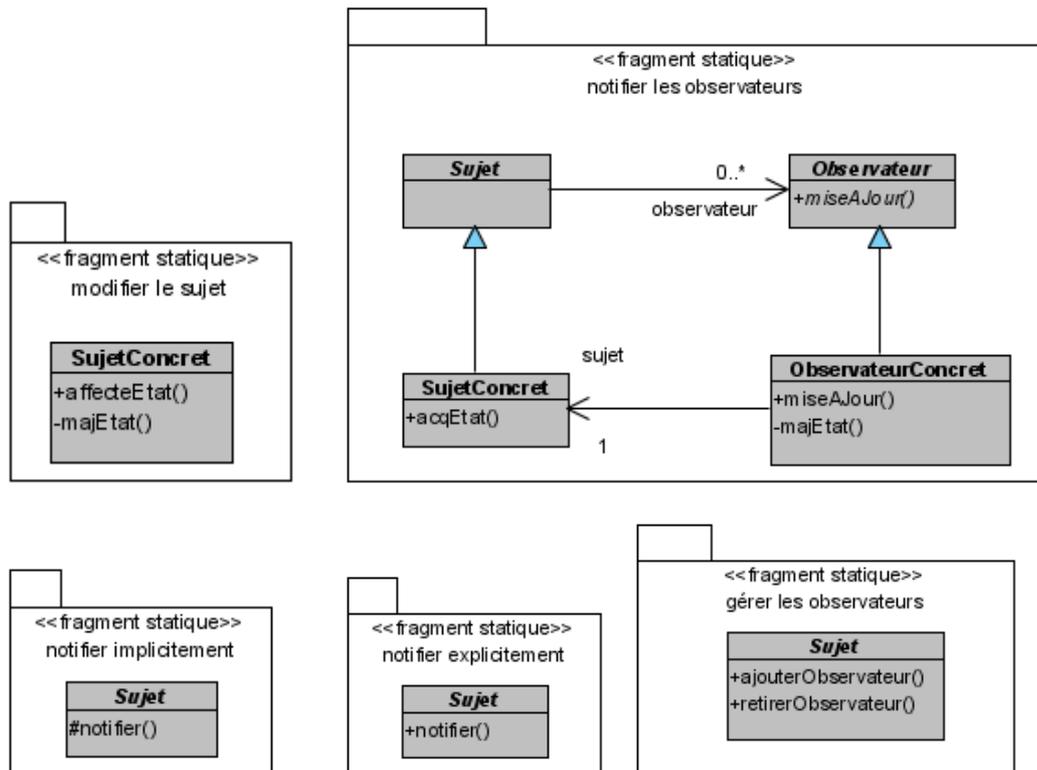


Figure 9 : Fragments statiques pour le patron Observateur

2.3.3. Complétude et variabilité pour les composants métier

Le travail présenté ici a été réalisé dans le cadre de la thèse de Rajaa Saidi, thèse en co-tutelle avec l'université Mohammed V-Agdal à Rabat (Maroc) dans le cadre du projet d'action intégrée franco-marocain COMPUS. J'ai co-encadré cette thèse avec Dominique Rieu côté français, et Mounia Fredj et Salma Mouline côté marocain. Cette thèse a été soutenue en septembre 2009 à Rabat (Maroc) (Saidi, 2009) et a fait l'objet de plusieurs publications dont (Saidi et al., 2009a), (Saidi et al., 2009b) et (Saidi et al., 2008a).

Les propositions présentées en section 2.3.2. pour les patrons de conception ont été complétées pour la spécification de composants métier réutilisables (par opposition aux objets métier) avec l'objectif de rendre réutilisables des processus métier identiques présents dans plusieurs systèmes d'information. En particulier, une nouvelle vue a été introduite : la vue métier (représentée sous forme d'un diagramme d'activités) permettant de spécifier, à un niveau métier, des composants métier réutilisables de type processus.

Vue métier avec variantes

La vue métier propose une spécification organisationnelle d'un composant métier processus mettant l'accent sur l'identification et la spécification d'une vue d'ensemble des activités internes d'un métier. Elle est modélisée par un diagramme d'activités UML étendu par les concepts de variabilité. En particulier, la variabilité dans la vue métier peut porter sur :

- le type de variabilité de l'activité elle-même, conforme aux 4 types de variation : Alternative, Alternative Optionnelle, Option, Ensemble d'alternatives,
- le degré d'informatisation d'une activité : manuel ou informatisé,
- le rôle d'un acteur : selon le cas, un rôle peut être optionnel (présent dans un processus métier d'un système d'information et absent dans le même processus métier d'un autre système d'information) ou varier d'un processus métier à un autre.

Pour exemple, la Figure 10 illustre une partie de la vue métier du composant « Demande Allocation Abstraite » modélisée par un diagramme d'activités supportant la variabilité. Ce composant est en réalité un sous-composant du composant métier processus « Gestion Allocation Ressource » qui traite du problème général de la gestion de ressources, par exemple la gestion des chambres d'hôtel (ressource non consommable), la gestion des billets de train (ressource consommable), etc. Dans le sous-composant « Demande Allocation Abstraite », deux points de variation sont identifiés :

- le point de variation « Demande Allocation Abstraite » est un point de variation de type Ensemble d'Alternatives, il représente le fait que le système devra pouvoir proposer au client une demande d'allocation manuelle et/ou informatisée (il s'agit d'une variation du degré d'informatisation),
- le point de variation « Traitement Demande Allocation » est un point de variation de type Alternative, il représente le fait que le traitement de la demande devra se faire avec ou sans réservation.

La vue métier constitue la vue organisationnelle du composant processus composée d'activités fixes (par exemple, l'activité DemandeAllocation dans la Figure 10) et d'activités variables. A partir des activités stéréotypées « **Informatisée** » de cette vue métier, la vue fonctionnelle avec variantes est construite par génération et mise en relation des cas d'utilisation sous la forme d'un diagramme de cas d'utilisation. En effet, le passage entre ces deux vues conduit à la génération d'une variabilité fonctionnelle identique à celle spécifiée par Nicolas Arnaud dans le cadre des patrons de conception et matérialisée par des fonctionnalités variables du composant processus. A titre d'exemple, la Figure 11 montre la vue fonctionnelle avec variantes du sous-composant « Demande Allocation Abstraite ».

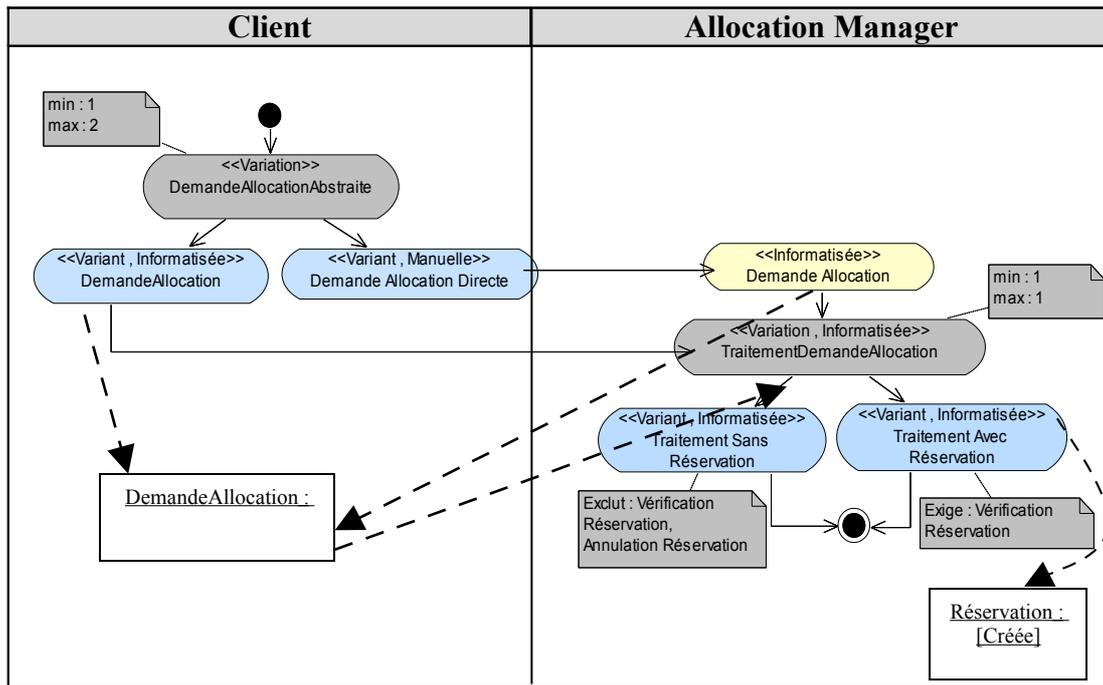


Figure 10 : Vue métier du sous-composant « Demande Allocation Abstraite »

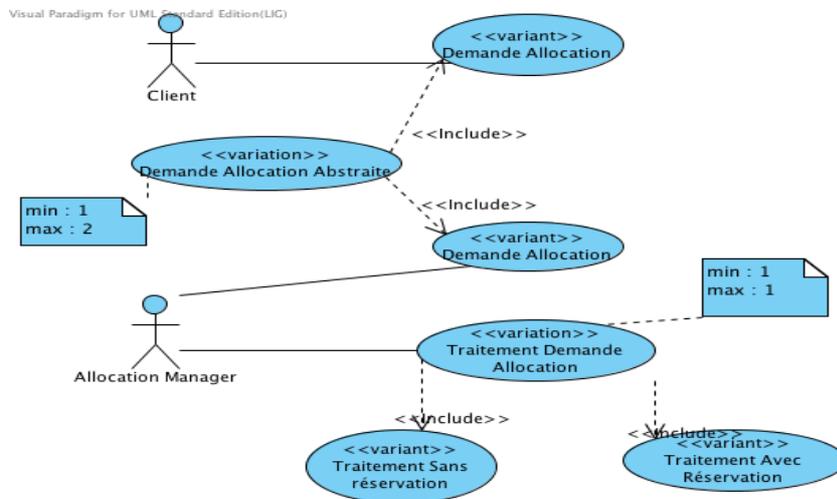


Figure 11 : Vue fonctionnelle avec variantes du composant métier « Gestion Allocation Ressource »

Ensuite, les vues dynamique et statique peuvent être automatiquement déduites de la même manière que pour les patrons de conception à la différence près que les fragments statiques sont représentés à l'aide du modèle conceptuel Symphony. Comme les activités et les cas d'utilisation, les fragments structurels peuvent être fixes ou variants. La Figure 12 montre un exemple de fragment structurel variant qu'il

conviendra de fusionner avec les fragments structurels fixes après réduction de la variabilité.

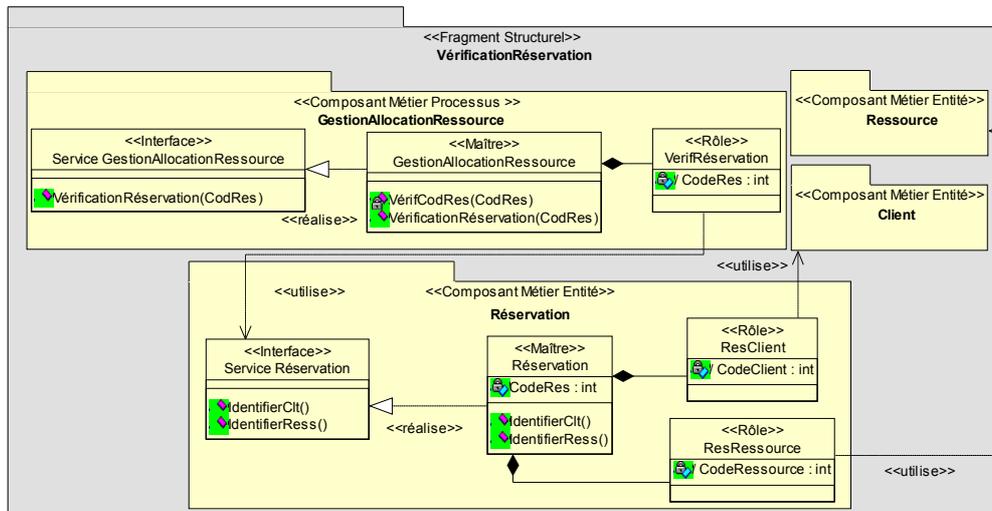


Figure 12: Exemple d'un fragment structurel variant

2.4. GÉNÉRICITÉ DES SPÉCIFICATIONS

Ce travail a également été l'une des contributions de la thèse de Nicolas Arnaud, il a été en partie implémenté dans le cadre du mémoire d'ingénieur CNAM de Thibaut Montanier que j'ai co-encadré avec Nicolas Arnaud et Dominique Rieu.

Comme nous l'avons souligné dans les enjeux, il est fondamental que des règles concernant l'utilisation d'un patron, d'un composant métier ou d'un framework soient précisées afin d'une part de permettre au concepteur d'applications de respecter l'essence d'un patron, d'un composant métier ou d'un framework, d'autre part de conserver la trace de toutes les utilisations de patrons ou de composants métier. De manière consensuelle, il est établi que la validité d'une imitation d'un patron par exemple s'appuie sur le respect de propriétés inhérentes à sa solution exprimant « l'essence » du patron (Pagel et Winter, 1996) (Arnaud et al., 2004). Une partie de ces propriétés existe déjà au sein de tout patron à travers les spécifications objet de la solution, elles sont destinées à être adaptées au contexte d'imitation.

D'autres propriétés dites génériques donnent lieu à des contraintes à respecter lors de l'imitation du patron. Un exemple d'une telle propriété générique concerne le fait qu'une classe ou qu'une opération doit rester abstraite lors de l'imitation d'un patron. C'est le cas par exemple des imitations des classes *Sujet* et *Observateur* dans une imitation du patron *Observateur* (cf. Figure 14). Un autre exemple concerne le nombre d'imitations possible d'un élément imitable (classe, attribut, opération...). Ainsi, dans une imitation du patron *Observateur*, les classes *Sujet* et *Observateur* ne doivent pas être multiples. Par contre, les autres classes, comme par exemple *ObservateurConcret*, ont vocation, en fonction du contexte, à être imitées plusieurs fois. Le but est donc de permettre à l'ingénieur de patrons de préciser, pour chaque classe et dans le cadre de l'essence de la solution du patron, si elle peut être imitée

plus d'une fois, l'objectif étant de fixer des « garde-fous » à respecter lors de l'imitation d'un patron.

De plus, il doit être possible, pour tout élément imité (appartenant à l'imitation), de retrouver l'élément imitable (appartenant à la solution) dont il est « issu ». Cette traçabilité est essentielle pour la vérification globale de l'imitation. Il s'agit ici d'une réponse au problème de traçabilité lors de l'imitation.

Notre proposition consiste à intégrer cette généricité dans les fragments statiques des patrons, c'est-à-dire dans la vue statique avec variantes des patrons. La Figure 13 montre comment le métamodèle UML-PI permet de définir ces propriétés génériques dans les constituants du fragment statique, ici sur le concept de classe imitable. Les deux métaclASSES *Solution* et *Imitation* regroupent les différents participants d'une solution (ses éléments imitables) et d'une imitation (ses éléments imités). Le lien solution-imitation est réalisé par délégation en mettant en relation un *ElementImitable* avec un *Element* (racine des types définis par UML). Dans la Figure 14, la métaclasse *ClasseImitable* est instanciée sur deux classes du patron *Observateur* pour garantir que la classe *Sujet* devra rester abstraite lors de l'imitation, mais que la classe *ObservateurConcret* pourra, elle, être rendue concrète par l'ingénieur d'applications lors de l'imitation.

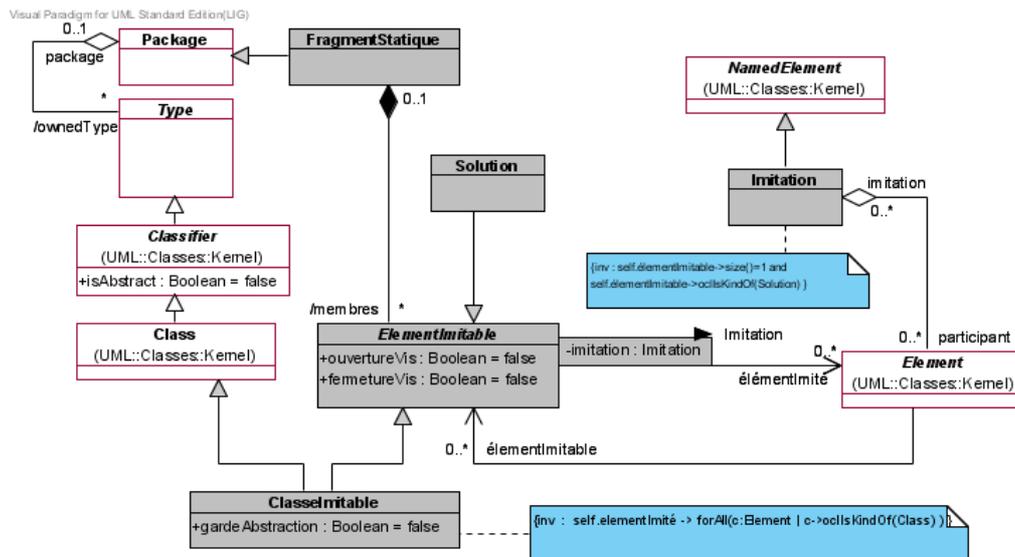


Figure 13 : Propriétés génériques dans le métamodèle UML-PI



Figure 14 : Deux instanciations de la métaclasse ClasseImitable

2.5. DES PROCESSUS DE RÉUTILISATION ADAPTÉS

Ce travail a également été mené dans le cadre des thèses de Nicolas Arnaud et Rajaa Saidi. Nous présentons ici le processus de réutilisation des composants métier proposé par Rajaa Saidi de par l'intérêt de son intégration dans un processus d'ingénierie de systèmes d'information.

Les techniques basées sur la complétude, la variabilité et la généricité que nous avons présentées précédemment nous permettent de proposer des solutions de patrons et de composants métier bien documentées, plus complètes, prônant une meilleure réutilisation et intégrant des garde-fous et des mécanismes de traçabilité censés minimiser les mauvaises utilisations. Néanmoins, de telles spécifications ne pourront être réellement adoptées et concrètement réutilisées par des ingénieurs d'applications que si elles sont accompagnées d'un processus et d'outils facilitant leur utilisation.

Ainsi, il est nécessaire de guider au maximum l'ingénieur d'applications lors de la réutilisation des solutions de patrons ou de composants métier que nous proposons. En l'occurrence, réutiliser des patrons ou des composants métier revient tout d'abord dans nos propositions à résoudre la variabilité, puis à adapter la solution obtenue après réduction au système d'information en cours de développement. Nous proposons donc des processus de réutilisation adaptés permettant de guider un ingénieur d'applications lors de la réutilisation d'un patron ou d'un composant métier en lui proposant de réduire la variabilité dans la vue « point d'entrée » (la vue métier pour les composants métier, la vue fonctionnelle pour les patrons). Par exemple, le processus de réutilisation d'un composant métier processus est composé de quatre étapes (cf. Figure 15) :

- *Recherche* : la recherche d'un composant est directement liée aux informations employées pour sa documentation.
- *Réduction* : il s'agit de choisir les variantes utiles spécifiées dans un CMP, selon trois activités principales :
 - *Sélection des variantes* : il s'agit de sélectionner dans la vue métier les variantes de chaque point de variation à l'aide de la documentation fournie pour chaque variante.
 - *Vérification de contraintes de dépendance* : une vérification des contraintes de dépendance est établie dans le but d'assurer une réduction cohérente. En cas de conflit, une nouvelle sélection des variantes est exigée.
 - *Réduction du CMP* : la réduction des vues fonctionnelle, dynamique et statique est automatiquement réalisée en utilisant des mécanismes de traçabilité de la variabilité implémentés dans des outils de transformation de modèles (cf. section 2.6.2.).

- *Adaptation* : l'adaptation du CMP réduit au contexte du système d'information en cours de développement se fait généralement par renommage et ajout de propriétés du CMP selon le contexte de réutilisation.
- *Intégration* : il s'agit d'assembler les solutions obtenues après la réduction de tous les CMP sélectionnés afin de concevoir un système complet.

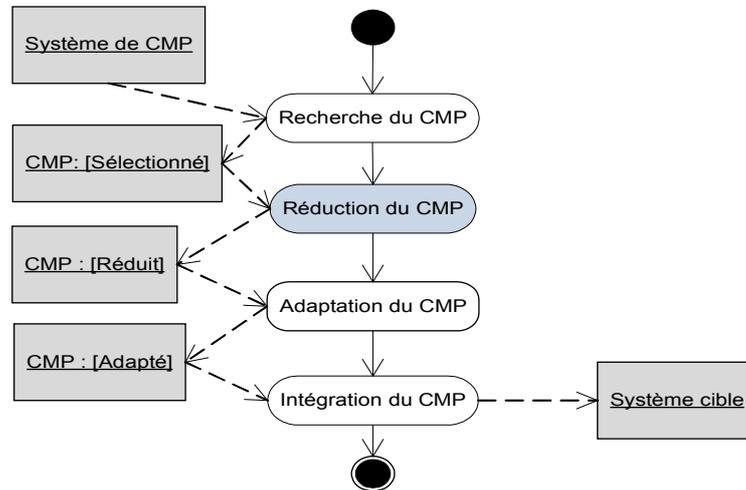


Figure 15: Processus de réutilisation de composants métier

De tels processus peuvent être complètement intégrés dans des processus de développement traditionnel de systèmes d'information. Ainsi, le processus de réutilisation des composants métier a été intégré au processus de développement Symphony (Hassine et al., 2003) (Hassine et al., 2005b), principalement dans la branche gauche de ce processus comme illustré en Figure 16. L'intégration de ce processus de réutilisation dans le processus Symphony permet, outre le « simple » développement d'objets métier modulaires lors de la conception d'un système d'information, de pouvoir réutiliser, dès la phase de spécification conceptuelle des besoins, des composants métier éprouvés afin de garantir une meilleure qualité du système d'information ainsi qu'un coût et des délais de développement moindres.

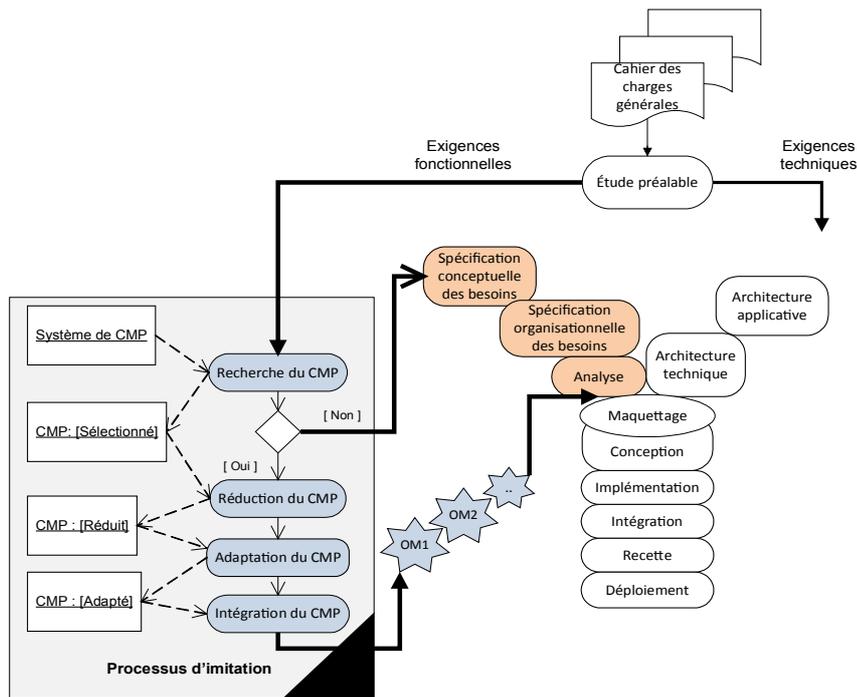


Figure 16: Processus d'imitation de composants métier intégré au processus Symphony

2.6. DES OUTILS POUR SUPPORTER LES PROCESSUS DE RÉUTILISATION

Les processus de réutilisation tels que celui présenté dans la section précédente peuvent être en grande partie automatisés dans des outils spécifiques. Je présente dans la suite de ce chapitre deux outils que nous avons développés, le premier pour la phase de recherche de composants, le deuxième dans le but spécifique d'automatiser la phase de réduction du processus de réduction des composants métier.

2.6.1. Un système de recherche de composants

Ce travail constitue l'un des résultats de la thèse de Oualid Khayati que j'ai co-encadrée avec Jean-Pierre Giraudin. Cette thèse a été soutenue en décembre 2005 (Khayati, 2005) et a fait l'objet de plusieurs publications dont (Khayati et al., 2005) et (Khayati et al., 2004). Bien que réalisé antérieurement aux travaux de thèse sur les patrons de conception et sur les composants métier, le but de cet outil est générique et il peut donc tout à fait être utilisé pour supporter la phase de recherche de composants dans les processus de réutilisation présentés précédemment.

Nous l'avons vu, la documentation d'un composant est capitale et illustre le contexte dans lequel il peut être utilisé en spécifiant les contraintes et les autres composants dont il a besoin pour offrir sa solution. Des bases documentées de composants sont donc des éléments clés dans les environnements de développement à base de composants. Malheureusement, mettre simplement à disposition des

développeurs une base de composants testés et validés de taille importante n'augmente pas forcément la productivité faute d'une organisation et de techniques de recherche appropriées. Dans le cadre de la thèse de Oualid Khayati dont la problématique est basée sur des travaux antérieurs effectués en collaboration avec un collègue spécialiste en recherche d'information (Berrut et Front, 2000), nous apportons des éléments de réponse à ces problèmes de recherche de composants. Plus précisément, les objectifs s'articulent autour de la construction d'un environnement d'aide à la réalisation et à l'utilisation de composants permettant de satisfaire aux conditions suivantes : hétérogénéité des besoins utilisateurs, hétérogénéité des niveaux d'abstraction de composants, hétérogénéité des modèles et des sources de composants, hétérogénéité des techniques de recherche de composants, adaptabilité et évolutivité des composants.

L'environnement proposé est basé sur un système de recherche de composants qui permet l'écriture et l'exécution de requêtes utilisateurs sur le contenu d'une « base descriptive de composants », référentiel fédérateur pour décrire tous les composants avec un formalisme très proche de P-SIGMA. Dans sa version prototype, le système de recherche de composants traite uniquement les requêtes simples applicables sur un seul type de données : les diagrammes de classes UML. L'interrogation de la base se fait donc simplement puisqu'une requête n'est autre qu'un diagramme de classes UML. L'idée est de pouvoir interroger une base descriptive de composants en fonction d'un diagramme de classes cible, de manière à poser des requêtes telles que « rechercher tous les composants qui dans leur solution, offrent un diagramme de classes proche du diagramme de classes ci-joint ». Ensuite, une technique structurelle de recherche de composants permet de représenter la signature et la structure interne des composants en utilisant des spécifications formelles en logique du premier ordre, et d'indexer les composants à travers leur description sous forme de diagrammes de classes UML. La phase d'indexation est entièrement automatisée, ce qui diminue le coût de mise en œuvre.

Ce système de recherche de composants est donc destiné aux *ingénieurs d'applications* qui recherchent dans une base descriptive de composants, les composants qui répondent au mieux à leurs besoins pour la construction de nouveaux systèmes d'information. Il adopte une architecture 2 tiers composée d'un SGBD relationnel et de cinq modules : Éditeur de requêtes, Générateur de spécifications, Moteur d'évaluation de requêtes, Moteur d'indexation, et Module d'import.

Grâce à l'éditeur de requêtes, la première étape consiste à exprimer la requête sous la forme d'un diagramme de classes UML. L'atelier de génie logiciel *ArgoUML*¹ est utilisé pour la saisie des requêtes, ce qui facilite son adoption par les ingénieurs d'applications. Par exemple, dans la Figure 17, la requête consiste à rechercher tous les composants dont la solution correspond au diagramme de classes spécifié.

¹ <http://argouml.tigris.org/>

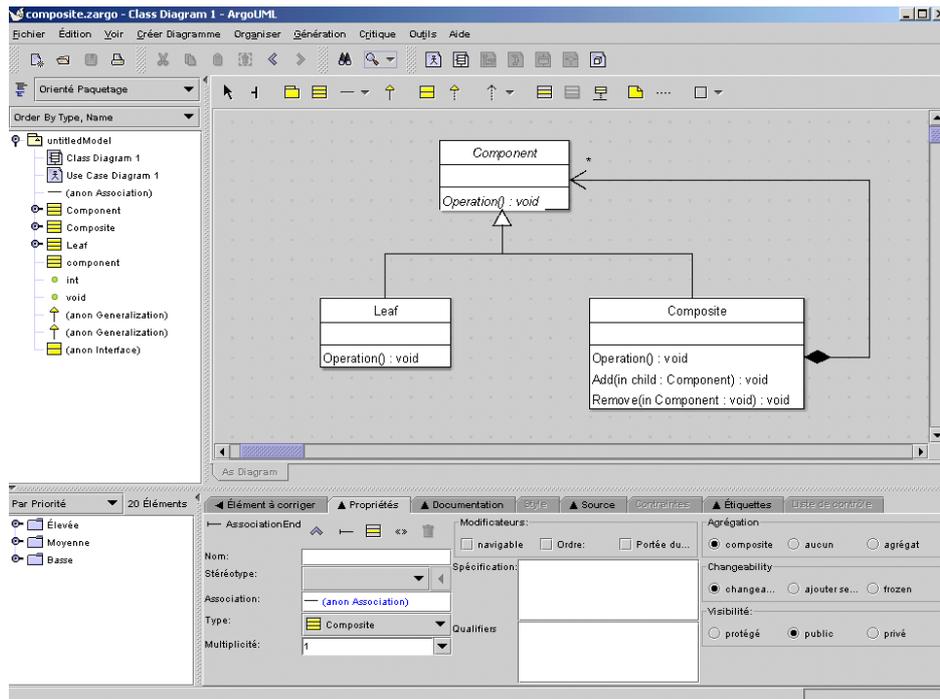


Figure 17 : Capture d'écran de l'AGL ArgoUML comme éditeur de requêtes

Le diagramme requête est enregistré en utilisant le format d'échange XMI. Le générateur de spécifications a alors pour rôle de récupérer ce diagramme requête et de le transformer dans un langage basé sur la logique du premier ordre (Prolog). Cette transformation d'un diagramme de classes au format XMI en une spécification formelle source en Prolog s'effectue grâce à une transformation XSLT dans l'outil Altova XMLSPY².

Le moteur d'évaluation des requêtes (cf. Figure 18) repose sur le moteur de Prolog pour réaliser l'appariement entre la requête et les spécifications des composants sources stockés dans le SGBD. Il compare alors la requête aux spécifications des composants sources, puis retourne les résultats.

Cet environnement constitue un premier pas vers la recherche de composants et diffère des moteurs de recherche traditionnels au sens où il ne recherche pas les composants selon les mots-clés du problème qu'ils résolvent, mais selon leur solution.

² <http://www.altova.com>

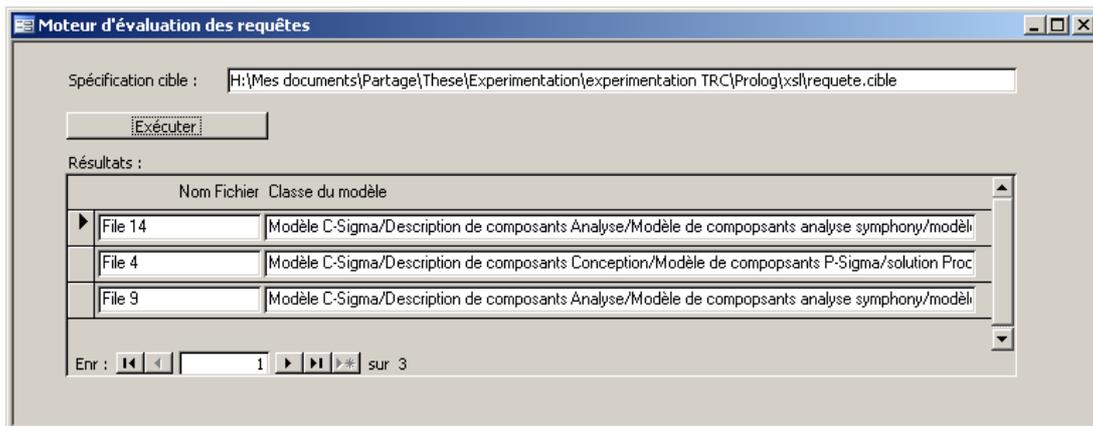


Figure 18 : Capture d'écran du moteur d'évaluation de requêtes

2.6.2. Les transformations dans les outils IDM

L'utilisation des outils IDM présentée ici a été expérimentée principalement dans le cadre des thèses de Nicolas Arnaud et Rajaa Saidi, ainsi que dans le projet de recherche de Master 2 Recherche de Safa Hachani soutenu en juin 2009 et du projet de fin d'études de Master 2 Professionnel de Stéphanie Marsal-Layat soutenu en juin 2007, que j'ai tous deux co-encadrés avec Sophie Dupuy-Chessa.

La phase de réduction des processus d'imitation de patrons et de composants métier peut être rapidement automatisée de manière pertinente grâce à l'essor des principes et outils de l'approche IDM (Ingénierie Dirigée par les Modèles). En effet, cette approche permettant de créer et transformer des modèles en se basant sur leur métamodèle, est particulièrement adaptée pour instrumenter les processus d'imitation de patrons ou de réutilisation de composants métier. La réduction d'un patron ou d'un composant métier est alors vue comme une suite de transformations. Dans d'autres travaux de l'équipe, elle est également utilisée pour vérifier des contraintes de cohérence en particulier entre des modèles de métamodèles différents. Par exemple, dans une optique d'ingénierie de méthodes collaboratives, des vérifications de cohérence sont effectuées entre des modèles UML utilisés par des spécialistes de génie logiciel et des arbres de tâches utilisés par des spécialistes de l'interaction homme-machine (Perez-Medina et al., 2007).

2.6.2.1. Transformation de modèles pour la réduction des composants métier

Détaillons ici une utilisation du langage ATL (Atlas Transformation Language) pour l'automatisation du processus de réutilisation des composants métier dans un prototype appelé RCMP (Réduction de Composants Métier Processus). L'objectif de RCMP est d'assister l'ingénieur d'applications lors de la réduction d'un composant métier processus. Ainsi, il permet à un ingénieur d'applications de sélectionner des variantes en visualisant l'enchaînement des activités de la vue métier. Une fois les variantes sélectionnées, l'ingénieur d'applications déclenche la réduction du composant métier processus et peut par la suite extraire la solution réduite.

Le prototype RCMP (cf. Figure 19) est implémenté à l'aide du langage ATL (*ATLAS Transformation Language*) disponible en tant que plugin ADT (*ATL Development Tools*) d'Eclipse dans lequel les métamodèles doivent être eux-mêmes conformes au méta-métamodèle ECORE. Ainsi, le métamodèle d'entrée (CMP.ecore) représente le métamodèle proposé pour la spécification de la vue métier avec variantes des composants métier processus), et le métamodèle de sortie (UML.ecore) représente le métamodèle UML (en effet, la vue métier réduite est un diagramme d'activités sans variation, donc un diagramme d'activités UML « classique »). De plus, des règles de transformation, représentant les règles de réduction de la variabilité de la vue métier d'un composant métier, sont définies en ATL (CMP_Reduction.atl). Elles permettent, si le modèle source (ex. AllocRess.xmi) est conforme au métamodèle CMP.ecore, d'effectuer la réduction de ce dernier en un CMP réduit (ex. AllocRess_Réduit.xmi) conforme au métamodèle UML.ecore.

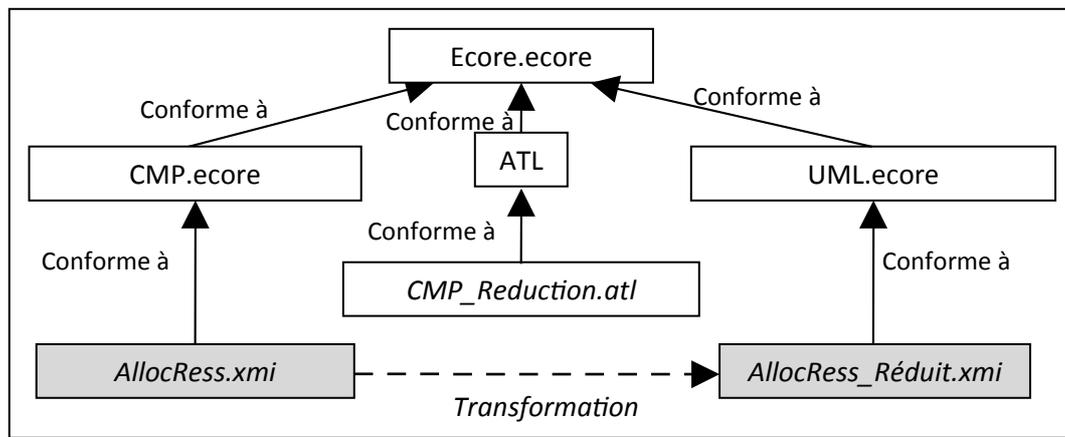


Figure 19 : Architecture technique du prototype RCMP (Saidi, 2009)

Pour utiliser le prototype RCMP afin de réduire un composant métier processus, un ingénieur d'applications doit tout d'abord importer le modèle (.xmi) du CMP sélectionné qui constitue le modèle source de la transformation. La Figure 20, partie gauche, illustre le modèle source « AllocRess.xmi » utilisé comme entrée de l'activité de réduction et qui correspond à la vue métier du CMP « Gestion Allocation Ressource ». On y retrouve en particulier les deux variantes « Traitement sans Réservation » et « Traitement avec Réservation » du point de variation « Traitement Demande Abstraite » de la Figure 10. La Figure 20, partie droite, illustre le modèle cible « AllocRess_Réduit » obtenu après exécution des règles de transformation (*Transformation.atl*). La réduction est réalisée très simplement en supprimant (dans le modèle source) les variantes non choisies, dans ce cas *CréationDemandeAlloc*, *TraitementAvecRéservation*, *VérificationRéservation* et *Suppression Ressource*. La Figure 25 illustre une partie de la vue métier réduite issue de ce processus de réduction.

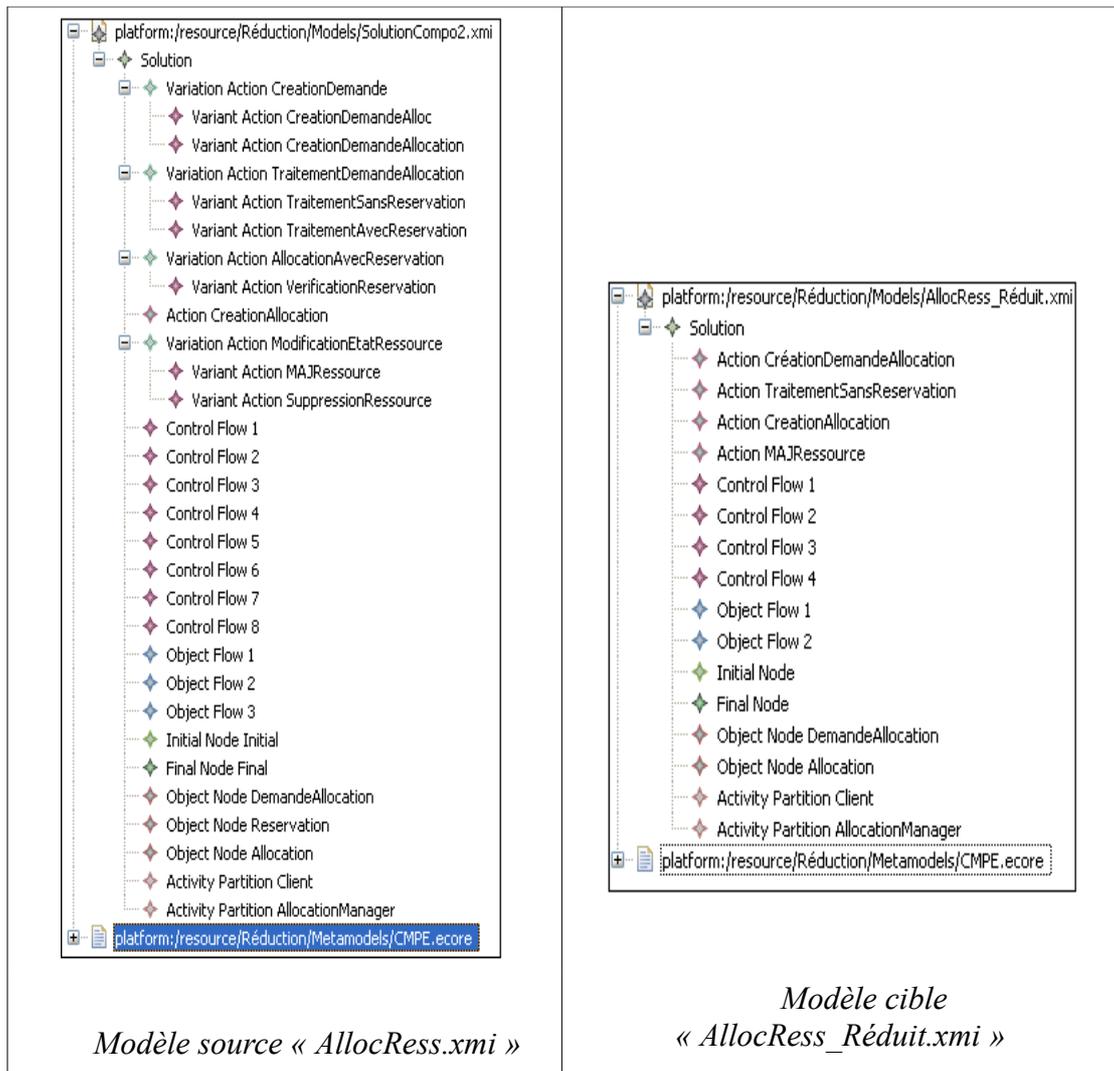


Figure 20 : Réduction d'un composant métier - Approche IDM

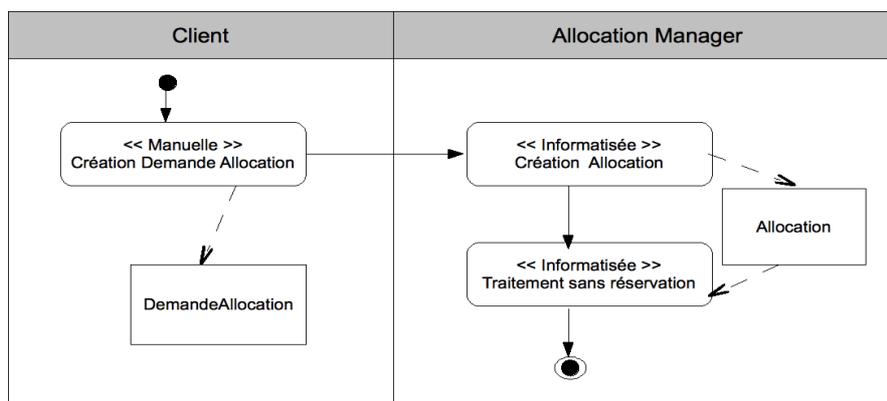
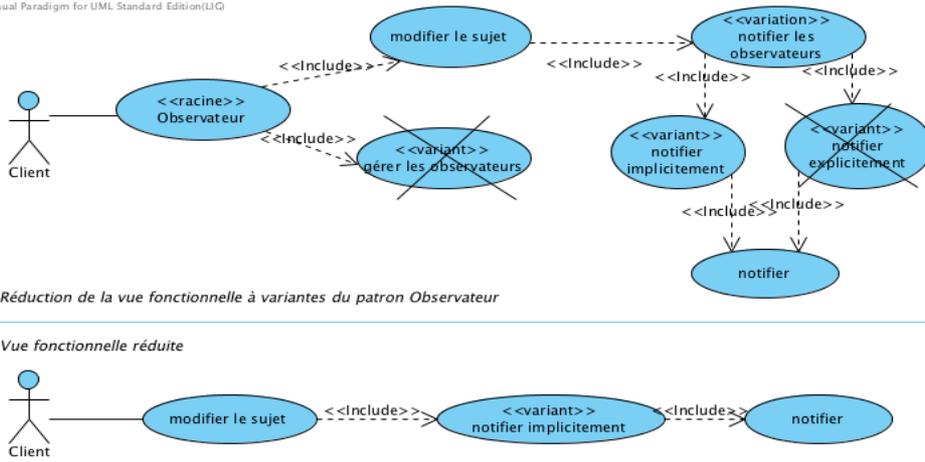


Figure 21 : Extrait de la vue métier après réduction

2.6.2.2. *Transformation de modèles pour la réduction des patrons*

Dans la thèse de Nicolas Arnaud, la même approche permet d'obtenir, à partir de la réduction de la vue fonctionnelle avec variantes du patron Observateur (cf. Figure 6), les vues dynamique et statique de façon simple et cohérente (Arnaud et al., 2006) (Arnaud et al. 2007a). Ainsi, construire la vue fonctionnelle est aisé : les cas d'utilisation sélectionnés sont gardés, les autres éliminés ; dans le cas d'une variante, cette dernière vient tout simplement remplacer son point de variation ; dans le cas d'une sélection multiple de variantes, les relations auxquelles participait le point de variation concerné sont dupliquées. Toute variante non sélectionnée est éliminée. Enfin, les stéréotypes de variabilité sont supprimés. L'acteur « client » est conservé. La Figure 22 illustre la réduction fonctionnelle du patron « Observateur » via la sélection de l'alternative *notifier implicitement*. La partie basse représente donc la vue fonctionnelle du mini-système imité.



Réduction de la vue fonctionnelle à variantes du patron Observateur

Vue fonctionnelle réduite

Figure 22 : Une réduction de la vue fonctionnelle du patron Observateur

Les vues dynamique et statique sont alors automatiquement déduites. Dans la vue dynamique, tous les fragments ne correspondant pas à un cas d'utilisation sélectionné n'ont plus de sens et sont à ignorer ; les stéréotypes sont également supprimés (cf. Figure 23). En ce qui concerne la vue statique il y a fusion des fragments correspondants aux cas d'utilisation sélectionnés (cf. Figure 24). La solution obtenue est proche de la solution originale proposée par (Gamma et al., 1994) (cf. Figure 6), mais est directement issue des choix de variantes réalisés dans la vue fonctionnelle par l'ingénieur d'applications : méthode *notifier()* protégée, et absence des méthodes de gestion des observateurs (méthodes *Attache()* et *Détache()*).

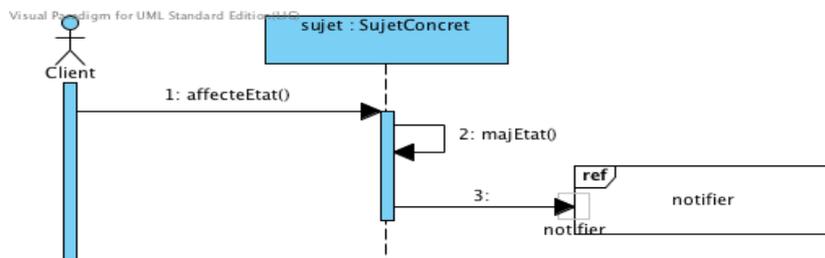
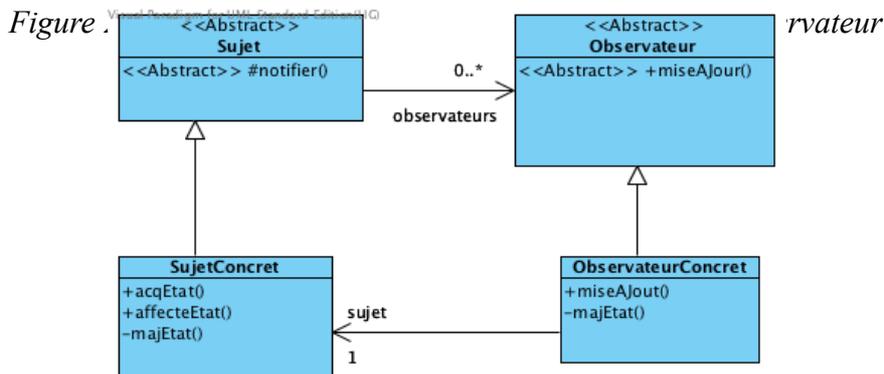


Figure 23 : Une réduction de la vue dynamique du patron Observateur



2.7. ADAPTATION DE MODÈLES DE PROCESSUS

En fournissant un ensemble de bonnes pratiques à respecter lors de la conception d'un système d'information, il apparaît évident que les processus d'ingénierie de systèmes d'information représentent des savoir-faire éprouvés qu'il est important de mettre à disposition des concepteurs. Il est donc indispensable de fournir des techniques permettant aux processus d'être non seulement réutilisés, mais surtout adaptés d'une organisation à une autre ou d'un type d'organisation à un autre. Ceci est d'autant plus vrai que les systèmes d'information sont aujourd'hui de plus en plus complexes.

Comme pour l'ingénierie de produits réutilisables, je présente ici quelques enjeux à mon sens incontournables dès lors qu'on parle d'ingénierie de processus adaptables. Ces enjeux sont globalement les mêmes que ceux liés à l'ingénierie de produits réutilisables, un modèle de processus pouvant être vu comme un produit particulier.

- Formalisation, compréhension et exécution des modèles de processus

Tout modèle de processus d'ingénierie de systèmes d'information doit être formalisé et documenté afin de pouvoir être correctement appréhendé par les ingénieurs d'applications. D'autre part, il est fortement souhaitable que les ingénieurs d'applications soient aidés et guidés lors de l'exécution d'un modèle de processus d'ingénierie de système d'information. L'exécution automatique d'un modèle de processus d'ingénierie de système d'information n'étant pas envisageable, il convient donc de favoriser le guidage des ingénieurs d'applications lors du développement d'un système d'information selon un processus donné.

- Complétude des processus

Différents types de modèles et métamodèles de processus d'ingénierie de systèmes d'information existent (orientés activité, orientés produit, orientés décision, orientés contexte et orientés stratégie) (Rolland et al., 1999) (Hug et al., 2009). Ces différents types représentent en l'occurrence différents points de vue des processus d'ingénierie de systèmes d'information et ne sont pas contradictoires, mais au contraire complémentaires. Il convient donc de combiner et de mettre en accord ces différents types de modèles et métamodèles de processus afin de fournir des descriptions complètes des processus d'ingénierie de systèmes d'information et adaptées au contexte d'utilisation et à l'expertise des ingénieurs d'applications.

Dans la suite de cette section, je présente mes principales contributions permettant de résoudre ces enjeux du point de vue de l'adaptation de processus d'ingénierie de systèmes d'information.

2.7.1. Préambule : la méthode Symphony

La méthode Symphony présentée ici a été proposée et spécifiée dans le cadre de la thèse CIFRE d'Ibtissem Hassine, que j'ai co-encadrée avec Dominique Rieu et Fethi Bounaas dans le cadre d'une collaboration avec la société Umanis. Cette thèse a été soutenue en septembre 2005 (Hassine, 2005) et a fait l'objet de plusieurs publications dont (Hassine et al., 2005a), (Hassine et al., 2005b) et (Hassine et al., 2003). La

formalisation et l'instrumentation de ce processus ont été réalisées dans le cadre du mémoire d'ingénieur CNAM d'Emmanuel Jausseran, que j'ai également co-encadré avec Dominique Rieu.

La méthode Symphony a vu le jour dans le cadre du contexte industriel de la société Umanis³ basée sur Paris. En effet, Umanis avait identifié début 2001, que de nombreux projets de développement de systèmes d'information qu'elle menait étaient en échec ou en dérapage (23 % d'échec et 49 % de dérapage constatés). Les causes identifiées étaient multiples : problèmes d'organisation et de qualité projet, problèmes de délais, mauvaise identification des besoins au démarrage du projet, absence de fil conducteur pour l'équipe de développement, etc. Outre des solutions en terme de pilotage (gestion du planning, suivi des charges de l'équipe, etc.) et de qualité projet (revue de projets, revues de documents, audits qualité, etc.), la société Umanis a émis le besoin d'une méthode de développement formalisée dès les phases amont du processus de développement (expression des besoins et analyse) et adaptée à la technologie à base de composants qu'elle utilisait pour mener ses projets de développement. Les objectifs de la thèse CIFRE d'Ibtissem Hassine furent donc les suivants :

- la prise en compte de l'ensemble du cycle d'ingénierie, dès les phases amont du développement,
- la prise en compte de la technologie composants tout au long du cycle de développement,
- la réutilisation systématique de l'expertise et des solutions,
- le guidage flexible du processus d'ingénierie.

Pour répondre à ces objectifs, les résultats de la thèse d'Ibtissem Hassine ont porté sur la formalisation de la démarche de développement de systèmes d'information Symphony. La démarche Symphony est constituée d'un modèle de processus (appelé dans la suite processus Symphony) et de différents métamodèles de produits, dont le modèle d'objets métier Symphony.

2.7.1.1. Le modèle conceptuel d'objets métier Symphony

Dans les travaux de thèse d'Ibtissem Hassine, nous avons adopté le terme d'objet métier en opposition à celui de composant métier car un objet métier Symphony n'a pas vocation à être réutilisable. Un objet métier Symphony (OM) est représenté sous la forme d'un paquetage UML composé de trois parties (Hassine et al., 2005a) (cf. Figure 25) :

- **une partie contrat avec l'extérieur** (ce que je sais faire) caractérisée par une classe Interface qui définit les opérations réalisées par l'OM et permet d'accéder à la structure interne de l'OM et d'exécuter les services qu'il offre.

³ <http://www.groupeumanis.com/>

- **une partie structurelle** (ce que je suis) caractérisée par une classe Maître et éventuellement des classes Partie. La classe Maître est la classe principale de l'OM. Dans l'exemple de la Figure 25, la classe Maître de l'OM *Sinistre* est la classe *Sinistre* dont une seule instance sera en cause lors de la création. Une classe Partie est une classe complémentaire de la classe Maître à laquelle elle est reliée par une relation de composition, c'est une structuration des attributs de la classe Maître pour permettre soit une mise en valeur d'un aspect de l'OM, soit une multi-occurrence d'un concept ou un aspect optionnel d'une caractéristique. Dans notre exemple, un sinistre peut faire l'objet de plusieurs expertises (classe Partie *Expertise*).
- **une partie collaboratrice** (ce que j'utilise) caractérisée par des classes Rôle. Une classe Rôle représente un fournisseur de services auprès de l'objet client, c'est-à-dire un autre OM qui rend un ou plusieurs services. Ces derniers sont appelés chez un client au travers d'une classe Rôle et d'un lien de dépendance stéréotypé « use ». La Figure 25 montre l'exemple de deux objets métiers *Sinistre* et *Facture*. Chaque sinistre peut être lié potentiellement à plusieurs factures de réparation. Une facture de réparation constitue une facture particulière (fournisseur) identifiée dans la compagnie d'assurance par un numéro de référence. Le concept *FactureReparation* est placé dans l'OM *Sinistre* et porte la formalisation du contrat entre le client *Sinistre* et le fournisseur *Facture* au travers d'une classe Rôle. L'attribut *agrée* correspond à une notion complémentaire aux factures classiques liée à une facture de réparation suite à un dommage.

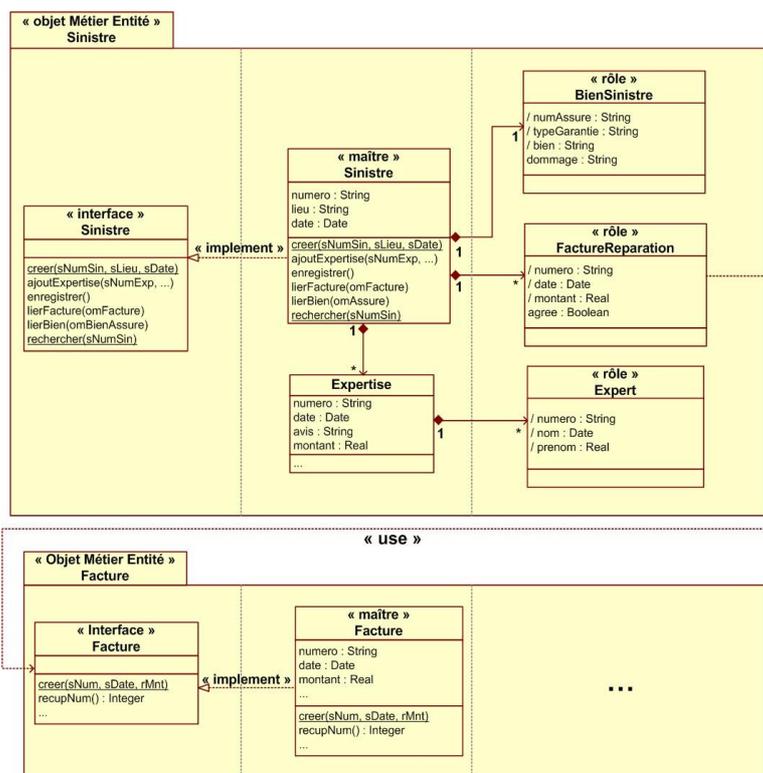


Figure 25 : Exemple de deux objets métier liés par un lien d'utilisation.

2.7.1.2. *Le processus Symphony*

Le processus Symphony a pour but de spécifier les différentes phases et activités d'un projet, de définir les tâches et les responsabilités de chacun des intervenants et de contrôler les coûts, les délais et la qualité de l'application logicielle produite. En ce sens, il est un processus orienté activité, instance du métamodèle de processus orienté activité SPEM.

De plus, Symphony est itératif et organisé au travers d'un certain nombre de cycles de développement en Y. Chaque cycle de développement, appelé aussi itération, est centré sur un processus métier ou le raffinement d'un processus métier identifié, décrit et pondéré lors de la phase d'étude préalable.

Le modèle de cycle de vie en Y (cf. Figure 26) s'articule autour de trois branches. La branche fonctionnelle (gauche) correspond à la tâche traditionnelle de modélisation du domaine et des besoins des utilisateurs. La phase de spécification conceptuelle des besoins et la phase de spécification organisationnelle des besoins contribuent à une première identification d'objets métier de haut niveau, et à un découpage ordonné (affectation de priorités) du processus de développement complet en itérations. Les résultats de l'analyse ne dépendent d'aucune technologie particulière et se traduisent dans un modèle d'objets métier. La branche technique (droite) a pour objectif le recensement des contraintes et des choix techniques nécessaires pour la conception du système tels que la sécurité, la montée en charge, l'intégration à l'existant, etc. Ces éléments permettent, par la suite, l'élaboration des architectures applicatives et techniques de l'application. L'architecture applicative identifie les composants techniques (frameworks, composants d'authentification, composants de notification, etc.) à mettre en place pour supporter l'exécution des objets métier. L'architecture technique décrit l'architecture matérielle et réseau du système de production. Enfin, la branche centrale est une intégration des branches fonctionnelle et technique. La phase de maquettage sert à définir et valider le scénario de navigation et l'organisation des écrans à travers une maquette. La phase de conception transforme les objets métier de niveau analyse en des objets métier de niveau conception. Le modèle de conception correspondant est obtenu par application de patrons de conception et de transformation, et est par la suite traduit dans un langage de programmation en utilisant les outils de développement choisis dans la branche technique. Les tests unitaires des composants sont réalisés au fur et à mesure que leurs unités de code sont développées. La phase de recette permet de tester fonctionnellement l'application en suivant le cahier des charges préalablement rédigé. La phase de déploiement consiste enfin à mettre en production l'application testée techniquement et fonctionnellement.

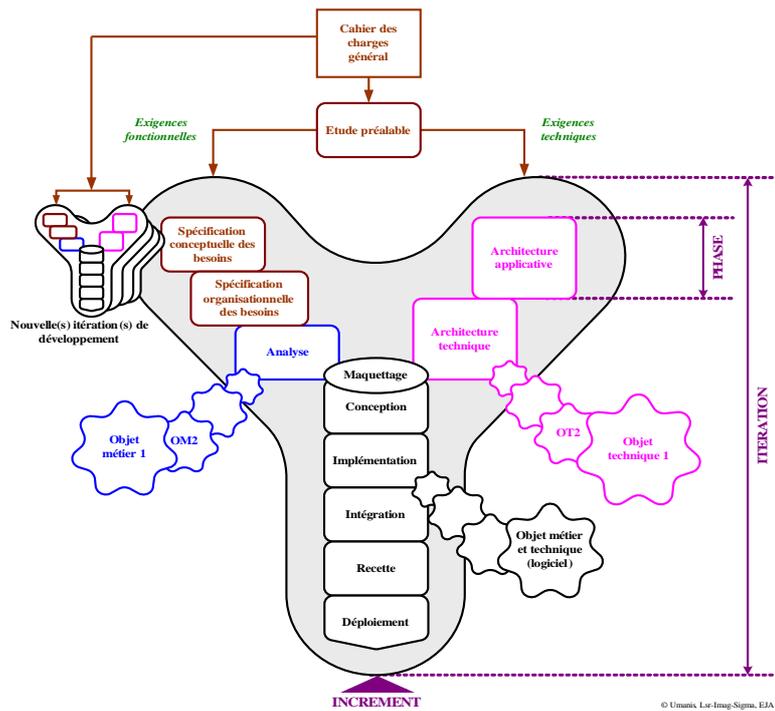


Figure 26 : Cycle de vie en Y du processus Symphony

Un cycle de développement est une instantiation de la succession des différentes phases du processus. L'enchaînement de plusieurs cycles de développement aboutit progressivement au produit logiciel final. Le processus global itératif de la démarche Symphony relève d'un nouveau type de cycle de vie généralisant le cycle de développement en Y : le modèle en flocons (cf. Figure 27).

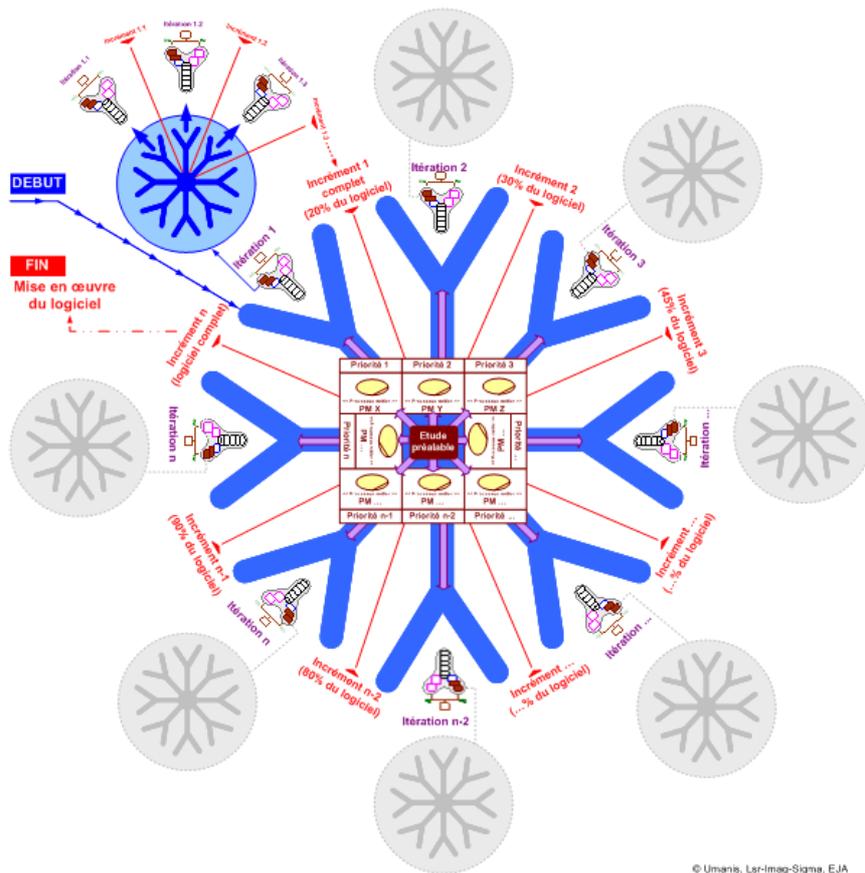


Figure 27 : Processus Symphony, modèle en flocons

Les travaux autour de Symphony ont apporté plusieurs résultats (Hassine et al., 2005) en termes :

- d'intégration des technologies à base de composants dès les phases amont (expression des besoins et analyse) du développement d'une application : une des contributions principales de la thèse d'Ibtissem Hassine a été la proposition du modèle conceptuel d'objets métier Symphony, qui permet d'identifier, dès le niveau de l'expression des besoins, des structures modulaires d'objets métier fortement inspirées de la technologie composants.
- de traçabilité des produits depuis l'expression des besoins jusqu'à l'implémentation : les cas d'utilisation identifiés au niveau de la phase de spécification conceptuelle des besoins représentent les processus métier à prendre en compte et, de la même manière que dans l'approche de Jacobson (Jacobson, 2000), sont déclinés dans les phases suivantes de la méthode en différents modèles grâce à des liens de dépendance de trois types : « est spécifié par » entre les modèles de cas d'utilisation et les modèles des phases de spécification organisationnelle des besoins et d'analyse, « est réalisé par » entre les modèles de cas d'utilisation et les modèles de la phase de conception, et « est implémenté par » entre les modèles de cas d'utilisation et les modèles

de codage ou de maquettage des phases d'implémentation. Ces liens de dépendance permettent de naviguer entre les modèles des différents niveaux d'abstraction, de remonter d'un modèle à un autre et de faciliter la propagation des modifications, garantissant ainsi une certaine forme de traçabilité.

- de réutilisation de solutions éprouvées : dans la version originale de Symphony, la réutilisation de solutions éprouvées est surtout mise en oeuvre dans la branche droite et dans la branche centrale. Dans la branche droite, la phase d'architecture applicative identifie un ensemble de composants techniques (frameworks, composants d'authentification, composants de notification, etc.) réutilisables et éprouvés. Dans la branche centrale, des patrons de conception sont proposés pour traduire les objets métiers identifiés dans la branche gauche dans des technologies cibles (Java, J2EE, EJB, etc.) capitalisant l'expérience des développeurs de la société. Dans la branche gauche, les objets métier identifiés n'avaient pas encore vocation à être réutilisés, mais les travaux de thèse de Rajaa Saidi autour du modèle de composants métier présentés en section 2.3.3. ont permis, dans une version plus évoluée de Symphony de renforcer la réutilisation des composants conceptuels fonctionnels.

2.7.2. Formalisation d'un processus sous forme de patrons

Afin de répondre aux objectifs de la société (réutilisation de l'expertise) d'une part, et à l'enjeu de formalisation et de compréhension des processus d'autre part, nous avons proposé de documenter le processus Symphony sous la forme de patrons processus exprimés en P-SIGMA et couplés à l'atelier AGAP pour guider le concepteur de systèmes d'information dans l'exécution du processus.

Exprimés en P-SIGMA, les patrons du Système de Patrons Symphony offrent une documentation complète et précise de chacune des étapes du processus. Le Tableau 6 montre l'exemple du patron « Phase d'étude préalable » du processus orienté activité Symphony. Cette phase est composée de différentes étapes et fournit un ensemble de livrables détaillés dans la rubrique « Solution Démarche ».

Patron : Phase d'étude préalable	
Domaine	Domaine de développement
Technologie	Technologie de développement
P-System	Système de Patrons Symphony (SPS)
Formalisme	P-SIGMA
Identifiant	
Phase d'étude préalable	
Participant(s)	
<ul style="list-style-type: none"> • Maître d'ouvrage • Responsable métier 	

<ul style="list-style-type: none"> Analyste des processus métier 	
Classification	
Terme(s) de domaine	{ Modélisation métier ^ Processus métier ^ Processus ^ Objet métier }
Contexte	
Contexte textuel	<ul style="list-style-type: none"> Système d'Information important se prêtant à la modélisation métier Cahier Des Charges (CDC)
Problème	
L'objectif de cette phase est un découpage fonctionnel du métier en vue d'identifier les processus métier (processus liés au métier de l'entreprise) et les différents acteurs (externes et internes) de chacun d'eux.	
Force(s)	
Force(s)	Modèle des processus métier : <ul style="list-style-type: none"> processus métier, diagramme UML des cas d'utilisation des processus métier, description des processus métier, classification des acteurs. Cartographie des objets métier principaux. Itérations de développement par processus métier ordonnancées par priorité décroissante.
Solution Démarche	
Démarche formelle	
Démarche textuelle	La solution proposée par ce patron repose sur les activités détaillées suivantes : <ul style="list-style-type: none"> Recueil des besoins. Le maître d'ouvrage rédige le recueil des besoins. Découpage fonctionnel global. L'objectif est d'identifier et de spécifier les différents processus métier. Une vue d'ensemble du domaine est nécessaire pour une première modélisation des besoins du système final. Un processus métier est défini comme un sous-ensemble des activités internes du domaine d'étude dont l'objectif est de fournir un résultat observable et mesurable pour un utilisateur individuel du métier. Un processus métier est modélisé par un cas d'utilisation. Plusieurs acteurs peuvent être concernés dont l'un plus particulièrement : l'acteur principal. Description des processus métier. Les processus métier sont décrits en langage naturel mettant en évidence les interactions (flux

	<p>d'informations) entre les acteurs externes et les processus métier.</p> <ul style="list-style-type: none"> • Identification des acteurs externes et internes pour chaque processus métier. La recherche des acteurs externes relève de l'étude du périmètre du domaine d'étude. La recherche des acteurs internes se base plutôt sur l'étude des rôles des différents membres de l'organisation. Plus formellement, l'identification des acteurs se déduit des réponses aux différentes questions suivantes : à qui est destiné le processus métier (acteur externe principal) ? Qui bénéficiera de l'utilisation du système (acteur externe principal) ? Qui interagit avec le processus sans qu'il lui soit destiné (acteur secondaire) ? Qui utilise le processus métier (acteur interne) ? Qui est le déclencheur (acteur externe) ? • Validation et affectation des priorités. Une fois la liste des processus métier validés, il est recommandé d'appliquer une stratégie d'affectation de priorités des processus métier précédemment identifiés et validés dans le but d'organiser les différentes itérations par valeur ajoutée. ... <p>Chaque processus métier concerne intuitivement un objet métier principal. Cela permet d'établir une cartographie intuitive des objets métier.</p>
Solution textuelle	<p>Artefacts produits :</p> <ul style="list-style-type: none"> • Un livrable global : il donne, d'une part, une vue récapitulative des résultats (sous forme de diagrammes UML, formulaires, ...) obtenus lors de l'exécution des différentes activités, un modèle des processus métier, une cartographie intuitive des objets métier, et d'autre par une idée de l'avancement du projet. • Un livrable par processus métier. Ce type de livrable concerne un processus métier particulier en donnant la description du processus métier en langage naturel et une classification des acteurs externes et internes identifiés.

Tableau 6 : Patron processus «Phase d'étude préalable » du processus Symphony

La rubrique « Utilise » (non montrée dans le Tableau 6 car non pertinente dans le cadre du patron « Phase d'étude préalable ») identifie tous les patrons utilisés par le patron en cours, facilitant ainsi la navigation à travers l'ensemble des patrons, donc la « circulation » dans les différentes phases du processus.

La formalisation sous forme de patrons est un premier pas vers le fait de rendre les modèles de processus adaptables : chaque patron décrit en effet un fragment de processus, qui peut être sélectionné ou pas (aux contraintes inter-patrons près) par l'ingénieur de méthodes qui souhaite constituer un modèle de processus adapté à son organisation. La formalisation de Symphony sous la forme d'un système de patrons a ainsi facilité ses adaptations et ses évolutions ultérieures.

2.7.3. Guidage méthodologique pour l'exécution et le suivi du processus

A partir de l'ensemble des patrons permettant de formaliser un processus de développement de systèmes d'information particulier, un extracteur de démarches couplé à l'atelier AGAP permet d'extraire sous forme de site web indépendant, un didacticiel autonome permettant de former et de guider les ingénieurs d'applications

dans un processus de développement. La Figure 28 montre ainsi un exemple de site web généré à partir de la formalisation du processus Symphony. Ainsi, dans la fiche du patron « Phase de conception » du Système de Patrons Symphony, le champ « Démarche formelle » (rubrique « Solution Démarche ») présente le diagramme d'activité de cette phase et précise que cette image a été cartographiée. Cela signifie que des zones de ce diagramme permettent d'accéder à d'autres patrons (cf. Figure 29).

L'extracteur est couplé à un moteur de recherche qui permet d'effectuer des recherches par mots-clés sur les patrons à utiliser. Ce moteur de recherches adopte une approche plus classique que celle de l'outil de recherche de composants présenté en section 2.6.1. puisqu'il se base sur une connaissance a priori des mots-clés indexant un patron.

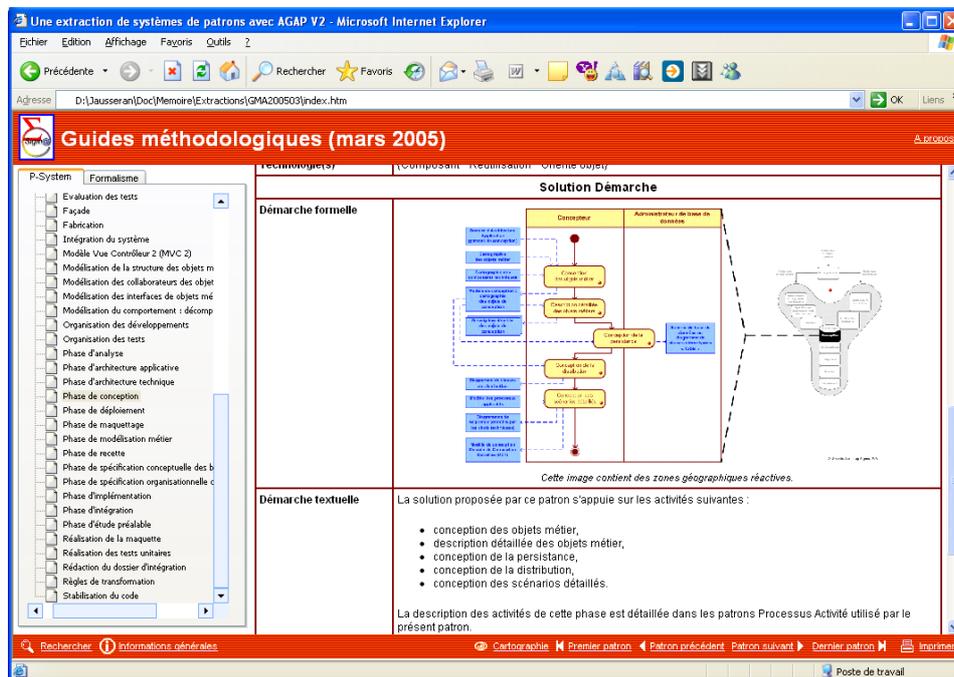


Figure 28 : Site Web autonome généré à partir d'AGAP

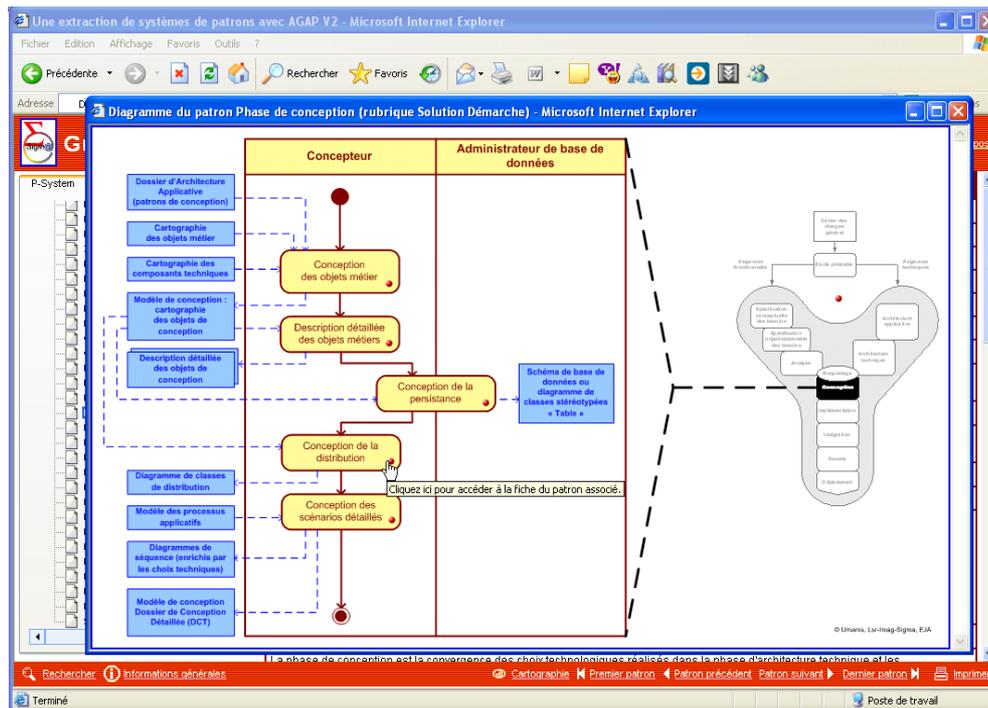


Figure 29 : Guidage inter-patrons processus

2.8. SYNTHÈSE

Une des problématiques de la recherche en systèmes d'information concerne le besoin de fournir des modèles (de produits ou de processus) efficacement et facilement réutilisables pour le développement d'un nouveau système d'information. Dans ce cadre, j'ai participé à la proposition de plusieurs contributions que je résume ci-dessous.

Tout d'abord, le formalisme de représentation de patrons P-SIGMA ainsi que l'atelier AGAP, proposés en collaboration avec Dominique Rieu, Jean-Pierre Giraudin, Ibtissem Hassine et Mounia Fredj, sont des supports efficaces pour une documentation complète et précise des patrons, facilitant ainsi leur réutilisation. Le formalisme P-SIGMA est lui-même réutilisable et adaptable et permet ainsi de documenter tout type de composant réutilisable, qu'il soit framework, composant métier ou modèle de processus. Ainsi, le framework WASSIT développé par Ahmed Zellou dans le cadre de sa thèse co-encadrée avec Dalila Chiadmi dans le cadre du projet franco-marocain STIC-GL a été documenté dans un formalisme adapté de P-SIGMA. Il en est de même pour les composants métier développés dans le cadre du projet de la Région Rhône-Alpes TRAFIC. Enfin, les modèles de processus peuvent être formalisés dans des patrons de type processus en vue de leur réutilisation ultérieure. Intégrés à l'outil AGAP, ces patrons processus peuvent ensuite être extraits sous la forme d'un site web autonome et guider l'ingénieur d'applications lors de l'exécution du processus. Cette contribution a été initialisée principalement dans le cadre de la thèse d'Ibtissem Hassine et du mémoire d'ingénieur CNAM d'Emmanuel Jausseran.

L'atelier AGAP, développé tout naturellement en support au formalisme P-SIGMA dans le cadre de plusieurs mémoires d'ingénieurs CNAM (Cathy Descombes, Laurent Tastet, Emmanuel Jausseran et Thibaut Montanier), est générique et permet de saisir n'importe quel formalisme de représentation de produits réutilisables et ainsi de gérer n'importe quel type de système de composants réutilisables. Couplé à l'extracteur, il permet en outre le guidage méthodologique lors de l'exécution d'un modèle de processus par un ingénieur d'applications. L'outil AGAP fut pendant plusieurs années l'outil support principal des thèses de l'équipe SIGMA. Aujourd'hui, son évolution est en cours, d'une part du point de vue de son aspect graphique afin de le rendre plus intuitif à utiliser, d'autre part du point de vue de ses fonctionnalités afin de le rendre collaboratif. Ainsi, le logiciel Copen, en cours de développement, fournira à terme une plate-forme collaborative de gestion des connaissances basée sur des technologies CMS.

Ensuite, l'introduction des concepts de variabilité dans une spécification complète et générique permet de proposer des solutions de composants plus précises. Associées à des processus de réutilisation adaptés, à des garde-fous et à des mécanismes de traçabilité plus ou moins automatisés, elles permettent à l'ingénieur d'applications, une fois le composant choisi, de réduire la variabilité sur l'une des vues spécifiées avant de simplement renommer les variantes choisies pour les adapter à son application. Ces propositions ont constitué les résultats des thèses co-encadrées avec Dominique Rieu, de Nicolas Arnaud sur les patrons, et de Rajaa Saidi sur les composants métier.

Les travaux présentés dans le chapitre précédent avaient pour objectif de formaliser et d'instrumenter les modèles de produit et de processus de manière suffisamment riche pour permettre leur réutilisation le plus aisément possible tout en respectant au maximum leur intention de départ.

Dans cette section, je présente quelques travaux dont l'objectif est légèrement différent : l'idée est ici soit de créer des nouvelles méthodes d'ingénierie spécifiquement adaptées à des organisations particulières, à des projets particuliers ou plus largement à de nouveaux types de systèmes d'information basés sur des technologies particulières, soit de proposer de nouvelles techniques permettant de construire ou d'adapter des méthodes existantes. Ces deux problématiques font actuellement l'objet de nombreux travaux de recherche dans des groupes de travail et conférences du domaine : groupe de travail *Ingénierie des Méthodes* du thème *Ingénierie par et pour la réutilisation* du GDR I3, IFIP Working Group 8.1 et son cycle de conférences ME dont la dernière édition a eu lieu à Paris en avril 2011, etc.

Concernant la proposition de nouvelles techniques pour la construction de méthodes, la plupart des approches propose des méthodes dites situationnelles basées sur des composants de méthodes ou des fragments de méthodes (Henderson-Sellers et Gonzales-Perez, 2011) (Mirbel and Ralyté, 2006) (Ralyté et al., 2007). Nous proposons dans la section 3.1. une approche différente basée sur la métamodélisation pour la construction de métamodèles de processus adaptés aux spécificités des organisations et des projets.

D'un autre côté, et poussés par l'avancée des technologies, en particulier ces dernières années par l'apparition de l'approche à base de services, les différents types de systèmes d'information sont de plus en plus complexes, omniprésents, multiformes et évolutifs. Les technologies embarquées, les dispositifs mobiles, les nouveaux types de données utilisés (comme les données issues de capteurs ou d'événements continus), permettent aujourd'hui de concevoir des systèmes d'information capables de s'adapter au dispositif utilisé, à l'utilisateur final et même à l'environnement. Ainsi, on voit apparaître des méthodes de conception à base de services (Fahkri et Cauvet, 2011) (Maamar et al., 2011), des méthodes pour les systèmes d'information dits durables (Pernici, 2010) (Pernici, 2011), des méthodes de conception pour les entrepôts de données (Annoni et al., 2006), etc. Dans le même ordre d'idées, les sections 3.2 et 3.3 présenteront des travaux dont l'objectif est de construire de nouvelles méthodes pour les nouveaux types de systèmes d'information.

Enfin, la section 3.4 présente une manière innovante de construire des nouvelles méthodes : la construction de méthodes par expérimentations.

3.1. UNE APPROCHE PAR MÉTAMODÉLISATION POUR LA GÉNÉRATION DE MÉTAMODÈLES DE PROCESSUS SPÉCIFIQUES

Ce travail a été réalisé dans le cadre de la thèse de Charlotte Hug que j'ai co-encadrée avec Dominique Rieu. Cette thèse a été soutenue en octobre 2009 (Hug, 2009) et a fait l'objet de plusieurs publications dont (Hug et al., 2009), (Hug et al., 2010a) et (Hug et al., 2010b).

L'objectif des travaux de la thèse de Charlotte Hug concerne le fait de permettre à des ingénieurs de méthodes de construire des métamodèles de processus pour l'ingénierie des systèmes d'information tenant compte des spécificités et des contraintes de leurs organisations.

Pour exemple, considérons le cas du service des systèmes d'information de l'hôpital de Grenoble⁴ où les processus sont analysés à deux niveaux : auprès des chefs de services soins et supports et auprès du personnel du service SI. Les chefs de services soins et supports s'intéressent aux objectifs d'un processus d'ingénierie de systèmes d'information. Par exemple, dans le cas de la mise en place de la gestion centralisée du dossier patient, les chefs de service veulent connaître l'impact de la nouvelle fonctionnalité du SI sur leur service. Cet objectif peut être détaillé en plusieurs points dont, par exemple connaître l'impact du changement sur les processus métier mis en place au sein de leur service, connaître les personnes de leur service impactées par le changement, connaître l'impact sur l'informatisation des bureaux et salles, etc. Par contre, dans le service SI, les chefs de projets et développeurs s'intéressent plus à l'aspect opérationnel des processus d'ingénierie de SI, c'est-à-dire aux différentes étapes à suivre pour réaliser les fonctionnalités, aux produits réalisés et aux acteurs intervenant durant les étapes. Par exemple, l'activité de pré-étude fonctionnelle qui consiste à définir de manière globale les besoins des utilisateurs, est composée de plusieurs tâches dont la rédaction du cahier des charges simplifié par l'analyste, la constitution du glossaire des termes métiers, la modélisation des acteurs qui produit un diagramme d'acteurs en UML, etc. L'ensemble des documents produits par l'activité de pré-étude fonctionnelle constitue un document appelé dossier fonctionnel.

Cette organisation a donc besoin de disposer de modèles de processus à deux niveaux : au niveau intentionnel pour les chefs des services, et au niveau opérationnel pour les chefs de projets et développeurs. Ces deux niveaux correspondent à deux visions complémentaires des modèles de processus :

- ceux des modèles de processus dits intentionnels tels que les modèles de processus orientés décision (Potts et Brun, 1988) qui présentent les transformations successives des produits dues à des décisions, orientés contexte (Rolland et al., 1995) prenant en compte l'intention et la situation d'un acteur à un instant donné du projet ou encore orientés stratégie (Rolland, 2005) permettant de représenter les processus multi-démarches,
- ceux des modèles de processus « traditionnels » orientés activité pour représenter les activités et leur ordonnancement pour la réalisation d'un

⁴ <http://www.chu-grenoble.fr/>

système d'information (modèles en cascade (Royce, 1970), modèles en V (Mc Dermid et Ripken, 1984), modèles itératifs tels que RUP (Kruchten, 2000), modèles en Y (2TUP (Roques et Vallée, 2002), Symphony (Hassine, 2005), ...), ou encore, plus récemment les processus des méthodes agiles telles que XP (Beck, 1999) ou Scrum (Schwaber et Beedle, 2001).

Tous ces métamodèles étant instances de métamodèles (en général du même nom), nous préconisons une approche par métamodélisation pour répondre à l'enjeu d'adaptabilité des processus dans un cadre structurant rigoureux. L'idée est de permettre à des ingénieurs de méthodes de construire des métamodèles de processus pour l'ingénierie des systèmes d'information tenant compte des spécificités et des contraintes de leurs organisations. Par opposition avec les techniques proposées dans l'approche des méthodes situationnelles (Kumar et Welke, 1992) (Harmsen, 1997) (Ralyté et Rolland, 2001) (Mirbel and Ralyté, 2006) (Ralyté et al., 2007) (Henderson-Sellers et Gonzales-Perez, 2011), ce ne sont pas des modèles de processus qui sont construits, mais bien des métamodèles de processus. En effet, comme pour les produits, l'utilisation d'un métamodèle permet de guider et de contraindre la définition des concepts des modèles de processus. Un métamodèle a ainsi un rôle structurant et permet de définir une trame générale des processus d'ingénierie de SI pouvant être réutilisée dans une organisation en définissant des modèles de processus propres à chaque projet à l'intérieur de cette même organisation. Par rapport à une approche de type « patrons de processus », fournir des métamodèles de processus permet de définir un cadre plus rigoureux permettant d'instancier des modèles de processus strictement conformes au métamodèle proposé, avec une sémantique et une notation graphique bien définies.

Cependant, nous sommes conscients que proposer un métamodèle de processus « from scratch » est difficile, en particulier pour les ingénieurs méthodes des organisations qui n'ont pas suffisamment de temps pour se former aux techniques de métamodélisation. Nous proposons donc d'aider ces concepteurs grâce à une méthode permettant de créer des métamodèles adaptés aux besoins des organisations, puis d'instancier ces métamodèles pour obtenir des modèles de processus si nécessaire propres à chaque projet, tout en vérifiant leur cohérence afin que les modèles instanciés soient eux aussi cohérents.

3.1.1. Une méthode pour guider la construction de métamodèles de processus

La méthode que nous proposons pour construire des métamodèles de processus d'ingénierie de systèmes d'information est destinée à des concepteurs de processus, elle est supportée par deux ressources principales :

- un graphe conceptuel point de départ de la méthode (cf. Figure 30) dont chacun des nœuds correspond à un concept de la métamodélisation des processus et peut être intégré sous forme de classe ou d'association dans les métamodèles de processus construits par les ingénieurs méthodes. Ce graphe est lui-même basé sur un métamodèle de domaine regroupant les principales classes des différents types de métamodèles de processus existants.

3.1.2. Un outil support

Afin de mieux guider les concepteurs de processus, la méthode est supportée par l'outil ProMISE dont un exemple d'utilisation est donné dans la Figure 32. Cet outil permet à des concepteurs de processus de naviguer dans le graphe conceptuel (cf. partie basse de la fenêtre), de sélectionner des concepts de ce graphe et de voir (onglet Process Meta-model Under Construction) le métamodèle de processus en cours de construction.

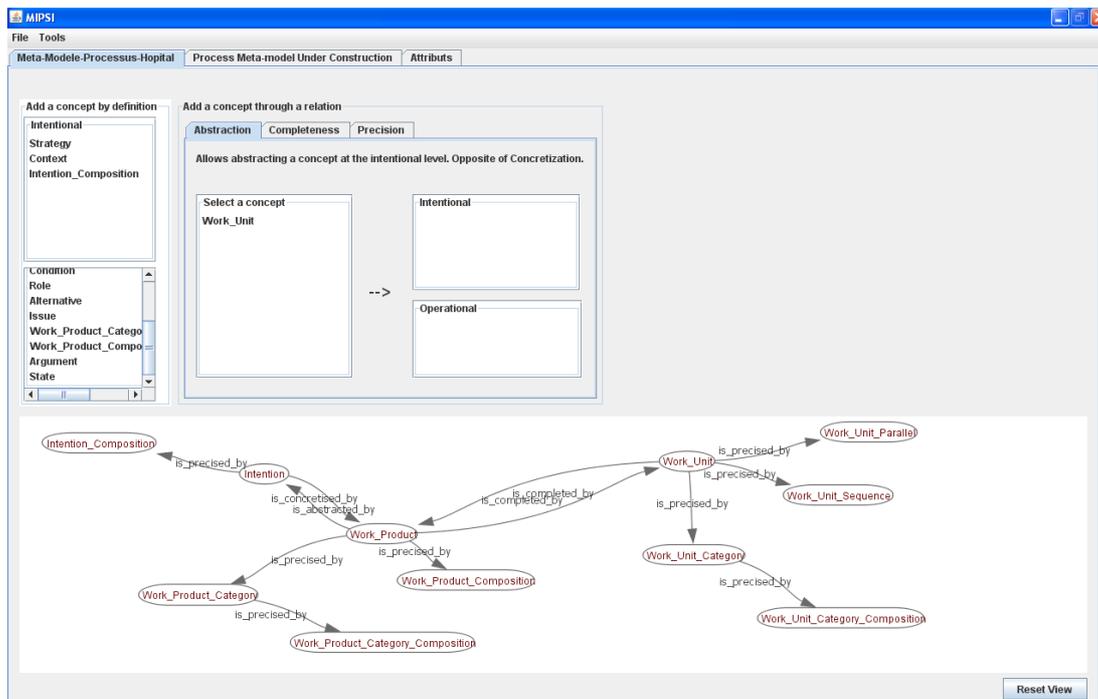


Figure 32: L'outil ProMISE, support de la méthode

La Figure 33 montre un métamodèle de processus obtenu par application de la méthode proposée sur le cas d'étude. La métaclasse *Intention* permet de modéliser les objectifs et sous-objectifs du processus mis en place à destination des chefs des services soins et supports. Ces objectifs se concrétisent sous la forme de *produits*, qui peuvent eux-mêmes être composés, et qui sont utilisés en entrée ou en sortie d'*unités de travail*. Les unités de travail correspondent aux différentes étapes qu'il est nécessaire de suivre, elles peuvent elles-mêmes être composées et sont réalisées par un ou plusieurs *rôles*. Enfin chaque produit et chaque unité de travail appartient à une *catégorie*. Produits, unités de travail et rôles sont du niveau opérationnel, à destination des chefs de projets et développeurs du service SI.

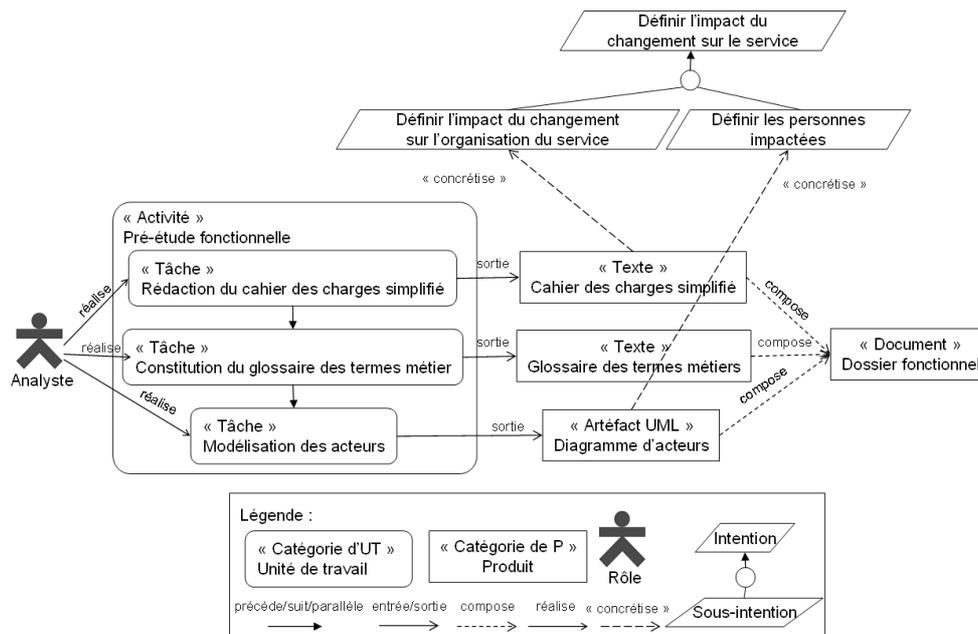


Figure 35 : Modèle de processus avec le formalisme KAOS

3.2. UNE MÉTHODE DE DÉVELOPPEMENT POUR LES SYSTÈMES D'INFORMATION DES ORGANISATIONS OUVERTES

Ce travail a été réalisé dans le cadre de la thèse de Luz-Maria Priego-Roche, soutenue en avril 2011 (Priego, 2011) et que j'ai co-encadrée avec Dominique Rieu. Cette thèse a fait l'objet des publications (Priego et al., 2009), (Priego et al., 2010a) et (Priego et al., 2010b).

Avec le développement des nouvelles technologies de l'information et de la communication basées en particulier sur Internet et les réseaux de communication, de plus en plus d'organisations se regroupent sous la forme d'organisation dites « étendues » ou « ouvertes » dans le but d'être plus compétitif ou de répondre à des demandes du marché auxquelles une organisation seule serait incapable de répondre. Ainsi, une organisation ouverte peut être définie comme « une alliance qui a pour but d'intégrer les compétences et les ressources de plusieurs entreprises réelles indépendantes, qui se trouvent dispersées géographiquement » (Davidow et al., 1992) (Travica, 1997) (Mazzechi, 2001) (Cheng et al., 2005) (Yang et al., 2006).

Pour fonctionner, une organisation ouverte a besoin d'un système d'information collaboratif, s'appuyant sur les services offerts par les différentes organisations participantes et devant permettre aux différentes organisations participantes de mieux communiquer, coopérer et collaborer dans le cadre de l'organisation virtuelle formée (Saad et al., 2003). Or, les différentes organisations participantes ne se connaissent pas toujours précisément et ne savent pas nécessairement ce qu'elles peuvent attendre des autres ou inversement leur apporter. Dans ce cadre, l'élicitation des besoins, déjà difficile dans le cadre de la conception d'un système d'information classique, devient problématique dans le cas de la conception du système d'information d'une

organisation ouverte, étant donné le nombre d'organisations et de personnes de domaines et de cultures différentes qui y sont impliquées.

De nombreux travaux ont mis en évidence l'importance d'une meilleure prise en compte du niveau intentionnel dans une méthode de développement de systèmes d'information. En particulier, les approches KAOS (Lamsweerde et al., 1998), i* (Yu, 1997), CREWS (Maiden, 1998) et MAP (Rolland et al., 2000) abordent le problème de l'élicitation des besoins à travers des buts (ou intentions) et des scénarios et proposent différents modèles pour représenter les buts à atteindre par les différents acteurs d'une organisation et les scénarios permettant d'atteindre ces buts. Néanmoins, ces différents modèles ne permettent généralement de définir les besoins que dans le cadre d'une seule organisation et restent donc insuffisamment précis dans certains cas, en particulier dans le cadre des organisations ouvertes ou étendues qui mettent en jeu plusieurs organisations.

Pour répondre à ce besoin, la thèse de Luz-Maria Priego-Roche propose un cadre méthodologique visant à compléter le niveau intentionnel d'un processus d'ingénierie de systèmes d'information par l'identification des caractéristiques et des besoins des différentes organisations participantes à l'organisation virtuelle dans le contexte de la mise en place de l'alliance liée à cette organisation virtuelle. Ce cadre méthodologique a pour objectifs :

- la mise en accord des organisations participantes par l'identification des caractéristiques liées à leur alliance (identification des organisations participantes, des compromis, des objectifs, etc.),
- la formalisation des besoins de l'organisation virtuelle, étape incontournable pour définir les processus organisationnels et par la suite, un système d'information collaboratif adapté.

Pour ce faire, ce travail se base sur des travaux issus du domaine de la gestion, travaux souvent très précis en ce qui concerne la définition des caractéristiques des organisations participantes et des stratégies souhaitées par l'organisation virtuelle, mais généralement peu formalisés et peu outillés. Ainsi, les sections suivantes montrent comment nous avons adapté et formalisé sous la forme de modèles UML certains travaux du domaine de la gestion, puis proposé des descriptions graphiques et textuelles simples basées sur ces modèles et permettant la définition des caractéristiques et des buts des organisations participantes à l'organisation virtuelle.

3.2.1. Cinq aspects pour caractériser le niveau intentionnel

De nombreux travaux de recherche dans le domaine de la gestion ont mis en évidence des propriétés permettant de caractériser une organisation virtuelle. Certaines de ces approches s'intéressent aux aspects collaboratifs (Kanter, 1994), (Robinson et al., 2005), d'autres mettent l'accent sur les processus métier (Barnett et al., 1994) (Cheng et al., 2005), sur l'identification des bénéfices communs (Marshall et al., 1999) ou la dispersion physique (Tripathy et al., 2007). Malgré l'existence de certaines plateformes technologiques dédiées à la collaboration entre petites et

moyennes entreprises telles que SME Collaborate⁵ ou ECOLEAD⁶, la plupart de ces travaux restent théoriques et peu ou pas formalisés. Néanmoins, ils semblent intéressants, voire nécessaires, pour compléter la phase d'élicitation des besoins au niveau intentionnel d'un processus d'ingénierie de systèmes d'information, en particulier dans le cadre d'une organisation virtuelle.

Les travaux de thèse de Luz-Maria Priego-Roche ont donc pour but d'intégrer et d'adapter les propositions issues de ces travaux tout en les formalisant par le biais de modèles UML, et en les instrumentant dans des plateformes logicielles. L'idée est en particulier de définir trois axes d'étude pour identifier et caractériser les organisations virtuelles au niveau intentionnel :

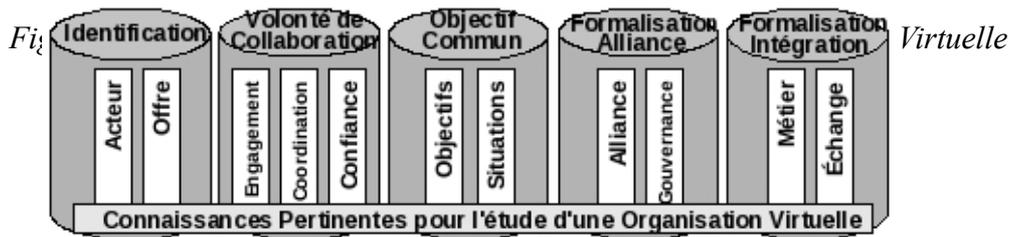
- l'élicitation des besoins intra-organisationnels : il s'agit ici de se focaliser sur l'orientation interne de l'organisation virtuelle, en particulier les relations entre les acteurs fonctionnels (individus ou groupes) des organisations participantes. Cet axe analyse en particulier les forces que mettent en jeu chacune des organisations participantes : leurs compétences, leurs ressources, leurs capacités, leurs processus métier internes, etc., donc de manière générale tout ce que chaque organisation possède et peut mettre à disposition de l'organisation virtuelle.
- l'élicitation des besoins inter-organisationnels : il s'agit ici de considérer les liens entre les organisations qui forment l'organisation virtuelle. Cet axe étudie la stratégie liée à la formation de l'organisation virtuelle en tant que plusieurs organisations collaborant pour atteindre un but commun. En particulier, les relations entre les diverses organisations participantes sont étudiées via la caractérisation de leur alliance et de leur collaboration.
- l'élicitation des besoins extra-organisationnels : il s'agit ici d'étudier l'environnement externe de l'organisation virtuelle, plus précisément les influences des acteurs hors de l'organisation quels que soient leur nature et l'impact de leurs influences, ceci afin que l'organisation virtuelle puisse mieux faire face à la concurrence ou rechercher de nouvelles alliances.

Ces trois axes sont définis au niveau intentionnel grâce à cinq aspects orthogonaux, chacun de ces aspects étant lui-même décomposé en deux à trois sous-aspects, comme illustré dans la Figure 36.

- Aspect *Identification* : il caractérise l'organisation virtuelle et sa composition en termes d'organisations et de services offerts,
- Aspect *Volonté de collaboration* : il caractérise les compromis entre les organisations participantes pour établir l'alliance et pour travailler ensemble,

⁵<http://www.smecollaborate.com>

⁶<http://ecolead.vtt.fi>



- Aspect *Objectif commun* : il représente le but partagé et les directions à suivre pour atteindre ce but,
- Aspect *Formalisation de l'alliance* : il précise la façon dont est construite l'alliance formant l'organisation virtuelle en termes de durée, de forme juridique (contrat, accord, fusion légale, etc.) et de « pouvoir » que chaque organisation aura dans l'alliance (une organisation leader, toutes les organisations sur un pied d'égalité, etc.),
- Aspect *Formalisation de l'intégration* : il caractérise le fait que selon l'accord, les organisations consentent à partager une partie de leurs compétences et de leurs ressources. En particulier, il permet de définir les processus métier que chaque organisation participante « offre » à l'organisation virtuelle, les ressources qu'elle est prête à donner et celles qu'elle attend de l'alliance.

Chacun de ces aspects (ou l'un de ses sous-aspects) est décrit par un ou plusieurs critères ou sous-critères, puis formalisé par un diagramme de classes UML. A partir de l'instanciation de ces modèles, réalisée dans un prototype support, des modèles basés sur des langages graphiques adaptés peuvent être générés comme nous le présentons dans les sections suivantes. Nous ne présentons pas ici l'ensemble de ces aspects, mais nous nous concentrons sur l'aspect Identification (Priego-Roche et al., 2010) en prenant appui sur l'étude de cas d'une organisation virtuelle particulière : l'Union des Eleveurs de Tabasco.

3.2.2. L'aspect Identification

L'aspect Identification est composé de 2 sous-aspects : l'acteur et l'offre.

L'acteur a pour but d'identifier et de caractériser les organisations (indépendamment de leur participation aux OV) ainsi que l'OV. Il se compose de 2 critères.

- Les Organisations. Chaque organisation est caractérisée par un nom, une description, une taille, une localisation (ville, département, pays), une constitution (organisation individuelle ou groupe) et des secteurs d'activité. La taille est déterminée selon une échelle allant de Petite à Grande, en se basant sur le nombre d'employés, le chiffre d'affaires, la part de marché et les particularités de l'organisation. Une organisation individuelle n'est pas composée d'autres organisations. Une organisation de type groupe est filiale d'une organisation mère qui contrôle ou dirige les activités du groupe. Les secteurs d'activité (spécifique et général)

caractérisent l'organisation dans son contexte d'affaires. Les secteurs sont caractérisés par un code, un nom et une description, ils sont liés à des activités (exemple d'activité spécifique : Production Animale ; exemples d'activité générale : Agriculture, Exploitation de forêts, Pêche).

- L'Organisation Virtuelle. Une OV a un nom, une description, une date de création et des secteurs d'activité spécifiques. L'OV est composée d'organisations (au moins deux), chacune d'elles jouant un rôle dans l'OV : soit en tant qu'Organisation Participante (OP) à l'alliance, soit en tant qu'Organisation Externe (OE) qui bien que ne faisant pas partie de l'alliance interagit directement avec l'OV en sollicitant ou en rendant des services. Un rôle a une durée fixée par une date de début et une date de terminaison.

L'offre a pour but d'identifier et de caractériser les services offerts par les organisations et les rôles que jouent les organisations pour produire ces services. Ce sous-aspect se compose de 2 critères.

- Les Services. Une OV génère un ou plusieurs services correspondant (Levitt, 1972) soit à l'offre d'un produit physique (comme la viande par exemple), soit à celle d'un service proprement dit (comme la distribution de la viande). Un Service a un nom et une description.
- Le Rôle. Chaque OP ou OE a un rôle dans l'accomplissement ou la consommation de ce service. Il peut s'agir d'un rôle d'approvisionnement ou d'utilisateur du service (on parle de relation de l'OP ou de l'OE vis-à-vis du service). Ce rôle peut être essentiel ou secondaire pour l'accomplissement du service, on parle d'influence de l'activité de l'OP de l'OE. Les rôles identifiés sont issus des travaux de (Basole et al., 2008) : un fournisseur de service auxiliaire qui approvisionne les autres acteurs du réseau avec des services essentiels mais qui ne sont pas directement liés au domaine industriel, un fournisseur indirect qui approvisionne les fournisseurs directs, un fournisseur direct qui approvisionne les fournisseurs de service, un pourvoyeur de service qui fournit le service aux consommateurs et a un contact direct avec eux et un consommateur qui utilise le service.

La Figure 37 propose un modèle pour l'aspect « Identification ». La Figure 38 illustre une instanciation de cet aspect. L'écran en haut à gauche permet d'affecter des organisations à une OV (ici l'Union des Eleveurs de Tabasco), soit en tant que OP, soit en tant que OE. Chaque organisation (écran en haut à droite), par exemple l'organisation Eleveur, est caractérisée (taille, localisation, etc.) et associée à ses secteurs d'activité (écran en bas à droite). Finalement le service de l'OV est montré dans l'écran en bas à gauche ainsi que le rôle des éleveurs dans le service dans la fenêtre superposée en bas à droite.

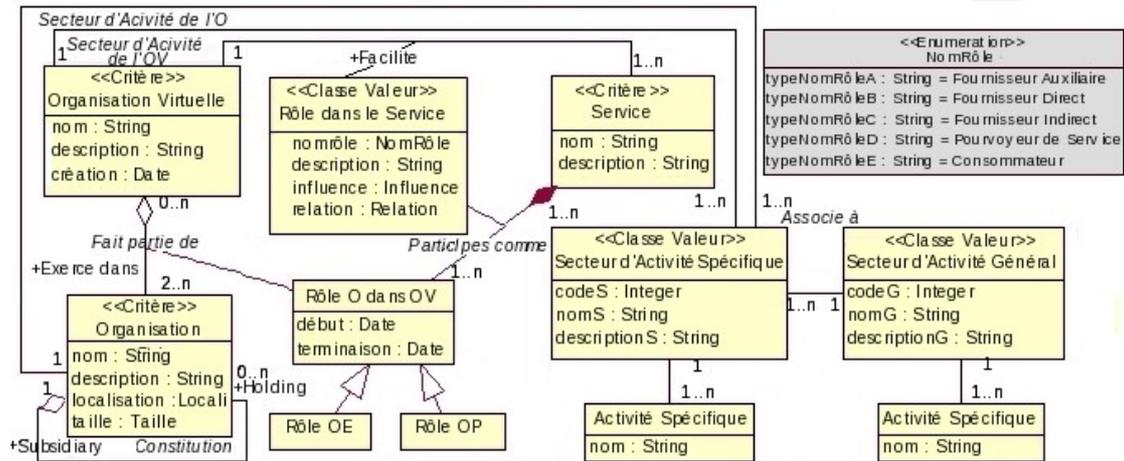


Figure 37 : Le modèle pour l'aspect Identification

Figure 38 : Une instanciation de l'aspect Identification

A partir de ces propriétés, il est possible de proposer des visualisations graphiques. La Figure 39 montre par exemple l'ensemble des organisations qui forment l'OV de notre étude de cas en se basant sur une adaptation du langage de modélisation de (Basole et al., 2008) : les OP (Eleveur, Associations, Abattoir, Transport, Commercialisation) et les OE (Chaîne de Supermarchés, Régulateurs Gouvernementaux, Consommateurs de Viande) qui composent le service « Conditionnement et vente de la viande bovine de qualité ». Les rôles de chaque organisation ou groupe d'organisations ainsi que leurs relations sont également illustrés. Dans notre outil support, cette représentation graphique est générée automatiquement à partir des données saisies.

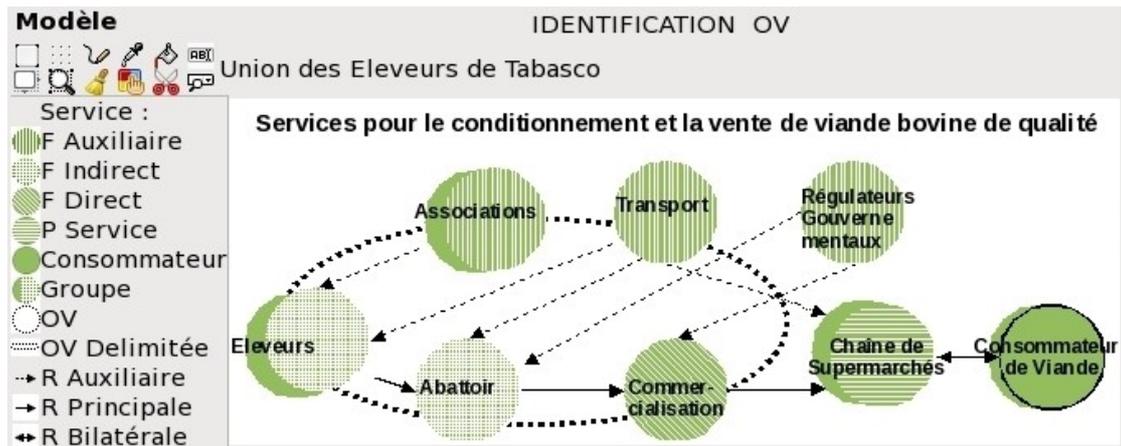


Figure 39 : Représentation graphique de l'OV Union des Eleveurs de Tabasco

3.3. UNE MÉTHODE DE CONCEPTION POUR DES SYSTÈMES D'INFORMATION UBIQUITAIRES PERSONNALISABLES

Ce travail est réalisé dans le cadre du projet DésIT, projet du cluster Territoire, Transport et Société de la Région Rhône-Alpes, démarré en septembre 2008 et que je coordonne, et de la thèse d'Ansem Ben-Cheikh que je co-encadre avec Jean-Pierre Giraudin du laboratoire LIG et Stéphane Coulondre du laboratoire LIRIS de Lyon. Cette thèse a fait l'objet des publications (Ben Cheikh et al., 2010a), (Ben Cheikh et al., 2010b) et (Ben-Cheikh et al., 2009b).

Aujourd'hui, les dispositifs mobiles tels que les téléphones portables, les smartphones ou les PDA permettent à des utilisateurs distants d'accéder à des systèmes d'information variés, distants et hétérogènes. On parle d'ubiquité ou de mobilité, c'est-à-dire le fait de pouvoir accéder à des informations n'importe où, n'importe quand et quasi n'importe comment. Or, chacun le sait, « trop d'information tue l'information ». Dès lors, la problématique de la personnalisation de l'information se pose : il s'agit de ne donner à l'utilisateur que des informations qui soient pertinentes pour lui, c'est-à-dire adaptées à son profil (ses préférences, ses activités...), à son contexte d'utilisation (sa localisation, les caractéristiques du dispositif mobile qu'il utilise...) et à son rôle par rapport à ces informations (simple consommateur d'informations, exécuter d'une partie du processus métier impliqué...), tout en continuant de respecter les droits d'accès aux informations. Ces deux problématiques (ubiquité et personnalisation) étant réunies, on parlera alors de **systèmes d'information ubiquitaires personnalisables**.

3.3.1. Un domaine d'application privilégié

Le domaine du transport de voyageurs est un excellent candidat pour l'expérimentation de ce type de système d'information. L'enjeu des systèmes d'information dans ce domaine n'est plus simplement de guider l'utilisateur dans la préparation de son parcours, mais bien plus d'accompagner ce dernier en temps réel durant tout son trajet. Pour ce faire, les systèmes d'information voyageur doivent être capables de donner à l'utilisateur les bonnes informations, c'est-à-dire celles qui lui seront réellement utiles, au bon moment et selon une présentation adéquate. Cependant,

même quand ces informations sont disponibles au sein du système d'information, les porter à la connaissance des usagers est difficile et implique d'adapter plusieurs aspects :

- **la nature des informations distribuées** : il faut sélectionner les informations pertinentes à chaque type d'utilisateur (les informations destinées aux contrôleurs ou à la police des transports ne sont pas les mêmes que pour l'utilisateur) selon sa position géographique. De plus, ces informations doivent être valides à l'instant de leur demande : par exemple, un voyageur doit pouvoir recevoir des horaires de transport prévus pour préparer un voyage, mais aussi des informations temps réel pour modifier son itinéraire en cas de perturbation. Un autre exemple concerne les personnes à mobilité réduite qui ont d'énormes difficultés à s'assurer que la prise en compte de leur handicap puisse se faire à chaque changement (bus, taxi, train, tramway) et jusqu'à leur point d'arrivée.
- **la présentation des informations à l'utilisateur** : l'ubiquité implique aujourd'hui de pouvoir distribuer les informations sur les systèmes mobiles personnels des utilisateurs. Il convient donc de pouvoir adapter la présentation des informations à tous les types de dispositifs. Par ailleurs, les personnes dites « empêchées » doivent bénéficier d'informations présentées d'une manière adaptée à leur handicap (les mal-voyants disposeront par exemple d'informations auditives ou sensibles (type vibreur SMS), alors que les mal-entendants disposeront d'informations visuelles ou également sensibles).
- **les activités ou les processus métier à réaliser** : tout comme le fait que les informations doivent être adaptées à chaque type d'utilisateur, il convient d'adapter également les activités que chaque type d'utilisateur peut faire sur le système d'information ou les processus métier auxquels il peut participer (un contrôleur pourra par exemple accéder au même système d'information qu'un usager, mais avoir plus de droits et de tâches à réaliser sur ce système).

Néanmoins, concevoir des systèmes d'information personnalisables conduit généralement à collecter et traiter des informations nominatives relatives à la vie privée des personnes. Ces données sensibles doivent malgré tout être disséminées, ce qui implique que leur sécurité est de plus en plus indispensable. Au delà de la confidentialité et de la propriété des données d'entreprise ou institutionnelles, le respect de la vie privée et de la confidentialité des données personnelles est devenu une source d'inquiétude qui a conduit la législation à encadrer leur gestion. La CNIL accorde ainsi différents droits au propriétaire de la donnée : accès, rectification, suppression, définition de l'usage, droit à l'oubli, consentement d'utilisation, etc. Cependant, si ces droits sont garantis légalement, il en est tout à fait autrement techniquement : comme le montre très souvent l'actualité, l'accès aux données personnelles est encore relativement facile techniquement. Cette contrainte au premier abord purement éthique nécessite de repenser l'accès à l'information transport, de créer de nouveaux modèles et composants adaptés, fiables, extensibles et confidentiels, et de disposer d'une véritable méthode de conception pour assurer la qualité des systèmes déployés. C'est là tout l'objectif du projet DÉFIT.

3.3.2. Architecture technique visée

D'un point de vue technique, nous préconisons une architecture client/serveur basée sur la diffusion/réception de flux et la technologie bluetooth. Dans cette architecture, le système d'information du fournisseur de transport (SNCF, TAG...) émet un flux d'information décrivant en temps réel d'une part la topologie du réseau de transport, d'autre part les événements impactant le réseau (panne, grève...). Ces flux sont reçus et traités par le serveur DÉSIT, puis diffusés à tous les usagers par l'intermédiaire de la technologie bluetooth dont un certain nombre de bus et de tramways sont actuellement déjà équipés. Le système d'information de l'utilisateur, accessible depuis son téléphone portable par exemple, équipé lui aussi de la technologie bluetooth, et stockant son profil, est à l'écoute de ce flux, informe l'utilisateur des informations pertinentes pour lui et l'accompagne en lui proposant des services adaptés à son contexte et à son profil. Dans cette approche par diffusion/réception de flux, les informations sont traitées par le système d'information de l'utilisateur stocké sur son portable. Ainsi, les données personnelles des usagers sont sécurisées puisqu'elles sont et restent stockées sur le dispositif mobile de l'utilisateur.

3.3.3. La méthode E-CARE

D'un point de vue conceptuel, l'enjeu majeur est de fournir une démarche d'ingénierie permettant, dans des domaines d'application variés, de concevoir de tels systèmes d'information ubiquitaires, personnalisables et sécurisés. En effet, outre le transport, les domaines d'application de cette approche sont nombreux (tourisme, domaine médical, ...), d'autant qu'aujourd'hui, les systèmes à base de capteurs semblent promis à un bel avenir. Les flux générés par les systèmes à base de capteurs doivent pouvoir être traités en fonction du contexte et du profil des utilisateurs (ou des systèmes) qui les reçoivent.

Plusieurs démarches d'ingénierie de systèmes d'information ubiquitaires ont été proposées ces dernières années, en particulier dans (Henricksen et Indulska, 2006) (Ayed et al., 2007) (Kapitsaki et al., 2009) (Achilleos et al., 2010) (Serral et al., 2010) et (Vieira et al., 2010). Ces démarches accordent un intérêt particulier à la sensibilité au contexte considérée comme caractère dominant et unique des systèmes ubiquitaires. Cependant, plusieurs autres critères liés à ce type de systèmes tels que la distribution, la mobilité et la disponibilité ne sont pas traités dans ces approches. De plus, la plupart des approches se concentrent uniquement sur l'analyse et la conception du contexte en négligeant les aspects fonctionnels et techniques de développement des systèmes, ce qui rend ces approches incomplètes, à l'exception de [Henricksen et Indulska, 2006] et [Serral et al., 2010] qui évoquent ces aspects sans les traiter profondément. De plus, la plupart des approches proposent des métamodèles de contexte, mais ne fournissent pas de guides ni de consignes pour l'usage de ces métamodèles en vue de faciliter la tâche aux concepteurs. Enfin, l'usage des méthodes d'ingénierie des besoins pour définir les besoins ubiquitaires est absent dans toutes les démarches étudiées.

Dans la thèse d'Ansem Ben-Cheik, nous étendons la démarche de développement de systèmes d'information Symphony en ajoutant, en plus des branches fonctionnelle, technique et centrale, une branche contextuelle intégrant un ensemble de concepts et

de métamodèles (en particulier des modèles couplant diffusion d'événements et contexte) ainsi que de nouvelles étapes du processus de développement (cf. Figure 40). Cette branche conceptuelle est en grande partie menée par un concepteur de contexte qui collabore avec les autres acteurs du développement (expert GL, concepteur IHM, etc.) afin de déterminer les besoins du système d'information en matière de sensibilité au contexte. Dans la Figure 40, les phases originales et non modifiées de Symphony sont en clair, les phases colorées sont les phases ajoutées à Symphony ou modifiées dans la démarche E-CARE.

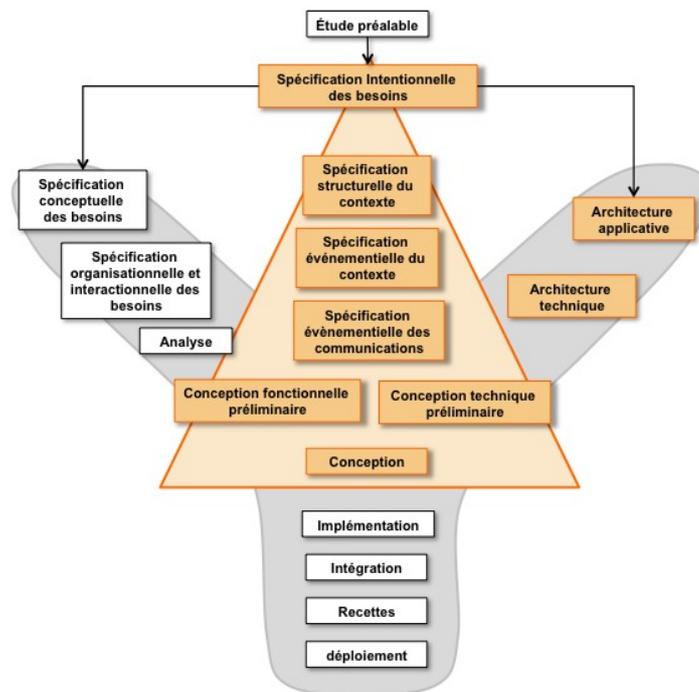


Figure 40 : La démarche E-CARE pour la conception de SI ubiquitaires

3.3.4. Des métamodèles spécifiques

La démarche E-CARE est basée sur des métamodèles spécifiques à la prise en compte des systèmes ubiquitaires : un métamodèle générique de contexte (cf. Figure 41) utilisé dans la phase « Spécification structurelle du contexte » et permettant l'acquisition du contexte, et un métamodèle d'événements (cf. Figure 42) utilisé dans la phase « Spécification événementielle du contexte » permettant d'exprimer dans quelle mesure l'application ubiquitaire doit réagir aux événements ambiants issus de son contexte.

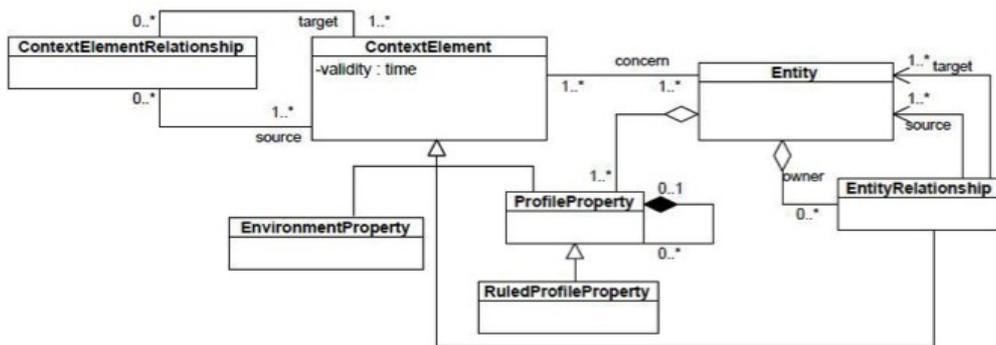


Figure 41 : Le métamodèle générique de contexte utilisé dans la méthode E-CARE

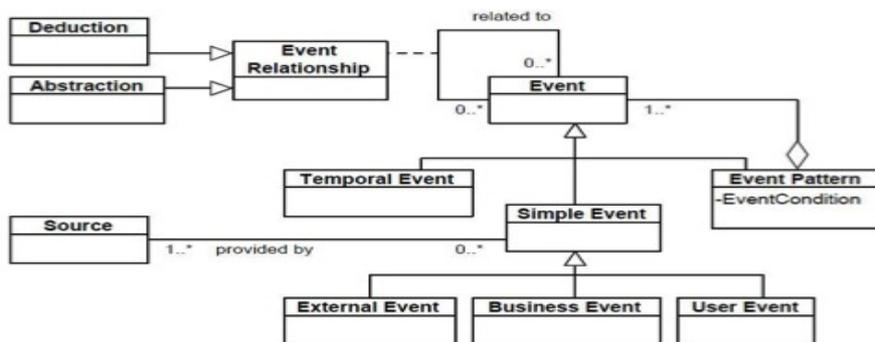


Figure 42 : Le métamodèle d'événements utilisé dans la méthode E-CARE

3.3.5. Les différents prototypes techniques réalisés

Dans le cadre du projet DÉSIT, plusieurs prototypes techniques ont été réalisés pour montrer la faisabilité des propositions réalisées dans le cadre du projet.

3.3.5.1. Un système d'assistance au voyageur avec Cayuga

Dans un premier temps, nous avons exploré dans le cadre du mémoire d'ingénieur CNAM de Yannick Tourasse, l'utilisation de l'approche à base de diffusion/réception de flux pour la conception d'un système d'assistance au voyageur. Cette implémentation a été réalisée à l'aide du Système de Gestion de Flux de Données Cayuga⁷, elle prend en compte :

- la modélisation du réseau de transports sous forme d'un graphe orienté dont les sommets représentent les arrêts (bus, tramway, etc.) et les arcs représentent les lignes de transport reliant les différents arrêts, ainsi que la géolocalisation des moyens de transport (cf. Figure 43),
- le calcul d'un itinéraire selon les préférences du voyageur (plus court chemin en distance, en temps, etc.), la détection des différentes perturbations pouvant

⁷ développé par Cornell University, <http://www.cs.cornell.edu/bigreddata/cayuga/>

survenir sur le réseau de transport (grève, manifestation...) et le recalcul automatique de l'itinéraire en fonction de ces perturbations (cf. Figure 44).

La réalisation de ce prototype a permis de valider l'architecture logique envisagée en début du projet DÉSIT et de montrer qu'il était approprié d'utiliser une approche orientée événements et diffusion/réception de flux afin d'accompagner les usagers des transports en commun lors de leurs déplacements.

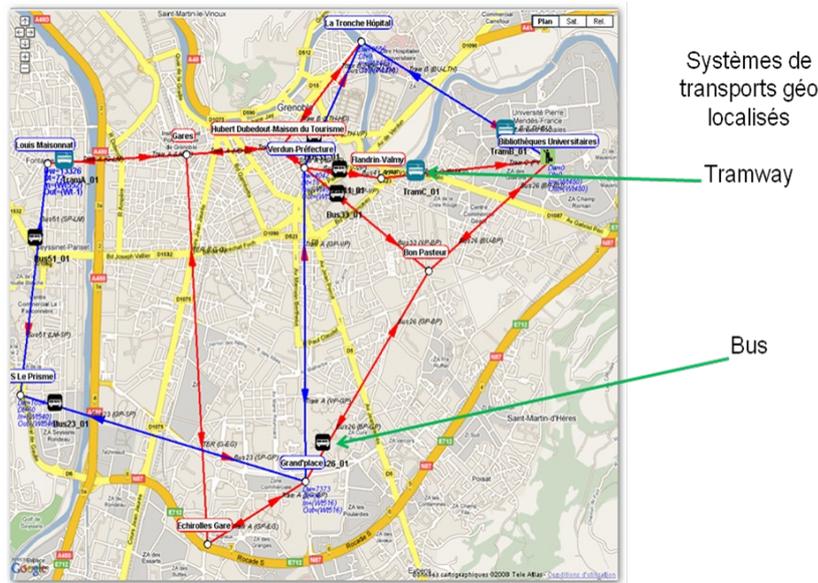


Figure 43 : Modélisation d'un réseau de transport et géolocalisation des systèmes de transport

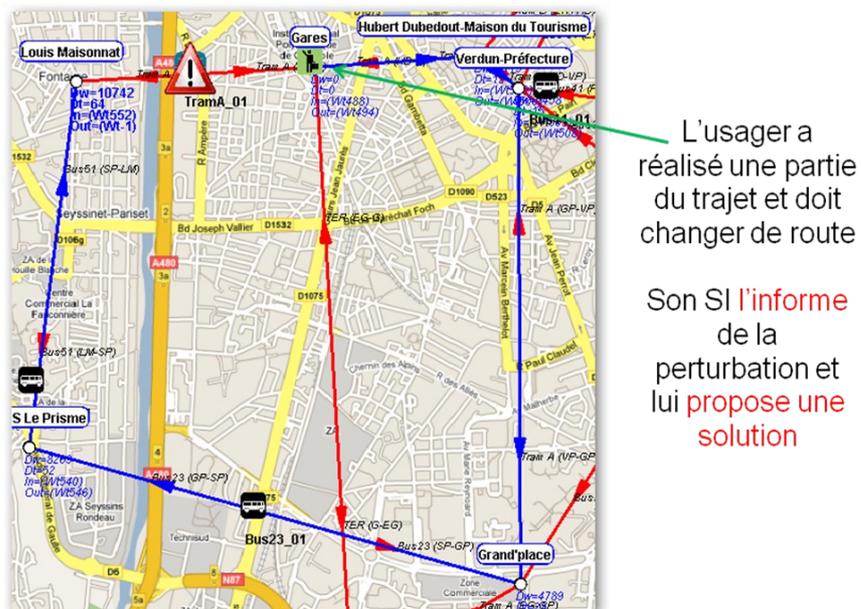


Figure 44 : Recalcul d'itinéraire suite à une perturbation

3.3.5.2. Portage sur téléphone mobile

Le prototype présenté dans la section précédente ayant permis de valider l'utilisation des systèmes à base de réception/diffusion dans le cadre du projet DéSIT, il a ensuite été porté sur un dispositif mobile par deux binômes d'étudiants encadrés par Stéphane Coulondre, membre du laboratoire LIRIS de Lyon, et partenaire du projet DéSIT. La Figure 45 présente trois photos de l'implémentation réalisée : la photo de gauche montre comment l'utilisateur est appelé à définir son profil (ici en termes de moyens de transport préférés, niveau de mobilité et type d'alertes souhaité) alors que les deux autres photos montrent respectivement une alerte de type SMS et une alerte de type SMS en mode vibreur, destinées par exemple à des personnes mal-entendantes.

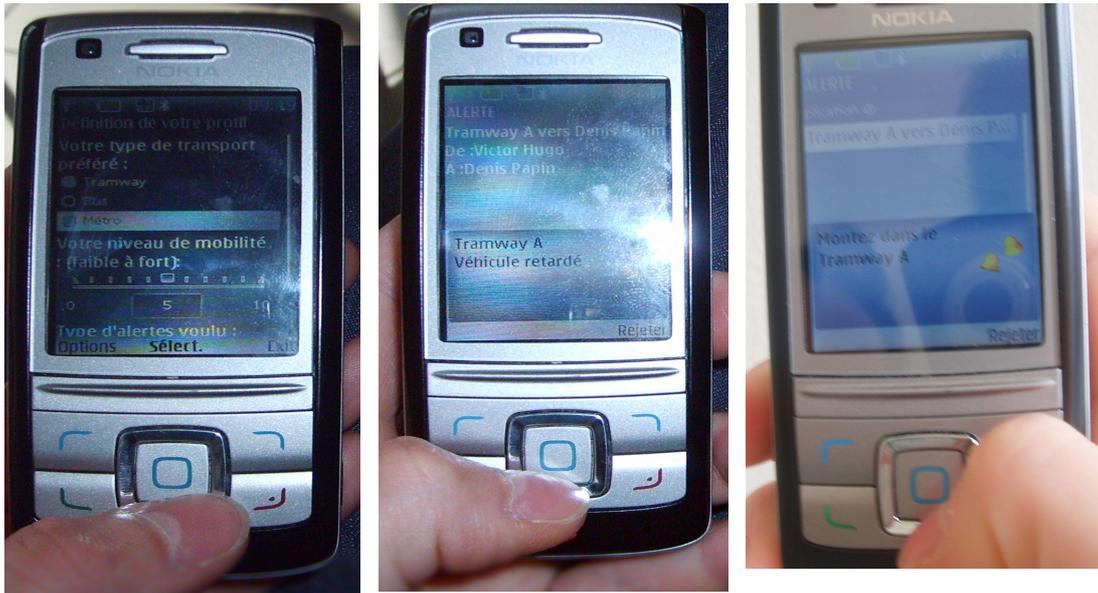


Figure 45 : Le prototype final sur téléphone mobile, du projet DéSIT

3.4. UNE MÉTHODE PARTICIPATIVE POUR LA MODÉLISATION DES PROCESSUS MÉTIER

Le travail présenté ici a été réalisé dans le cadre de la thèse de Marco Santorum que je co-encadre avec Dominique Rieu avec l'aide de la plateforme MarveLIG, pépinière d'expérimentations scientifiques du laboratoire LIG, et de Nadine Mandran, ingénieure en méthodes de qualité au sein du LIG. Il a fait l'objet des publications (Santorum et al., 2010) et (Santorum et al., 2011).

L'approche BPM (Business Process Management) a pour objectif principal de permettre aux utilisateurs et aux gestionnaires métier de mieux collaborer avec les équipes techniques afin de rendre les processus de l'organisation exécutables et contrôlables et ainsi d'aligner les services rendus par le système d'information avec la stratégie de l'organisation. Dans cette approche, le fait que le processus métier devienne soudainement au coeur de la politique d'urbanisation de l'entreprise,

impliquant de manière équivalente les différents acteurs de l'organisation, a pendant un temps, laissé penser qu'il devenait simple de modéliser les processus métier. Or, dans l'approche BPM, le cycle de vie des processus comprend trois phases principales : la modélisation des processus, la génération automatique de processus exécutables, et finalement le suivi et l'optimisation de l'activité des processus métier. Si l'exécution et le pilotage des processus métier sont effectivement relativement bien outillés aujourd'hui dans les outils BPM, le problème reste entier du point de vue de la modélisation des processus : la difficulté essentielle est toujours de trouver le temps de « bien » modéliser un processus métier et de maintenir à jour les modèles de processus en fonction des évolutions organisationnelles de l'entreprise. De plus en plus d'organisations ressentent le besoin de maintenir une cartographie à jour des processus métier, mais personne ne dispose du temps, des compétences et de la motivation nécessaires pour établir cette cartographie. Dans certains cas, une personne est déléguée pour réaliser cette tâche, mais le temps nécessaire pour réaliser une cartographie complète de processus métier est conséquent et il est rare que celle-ci reste à jour très longtemps.

Dans les travaux de thèse de Marco Santorum, l'objectif est de définir une méthode de conception des processus métier qui mettent en jeu tous les acteurs fonctionnels d'un processus métier afin de les impliquer collectivement dans la modélisation et l'amélioration du processus métier. Ainsi, non seulement les acteurs fonctionnels participent à la modélisation du processus (à l'instar des méthodes agiles où les clients sont régulièrement invités à participer aux réunions de suivi des projets), mais en outre, ce sont les acteurs fonctionnels eux-mêmes qui réalisent la cartographie des processus dans lesquels ils sont impliqués.

Pour ce faire, cette méthode, appelée dans sa version déposée sous enveloppe soleau auprès de l'INPI⁸ ISEA (cf. Figure 46), est basée sur le cycle de vie traditionnel des processus métier (modélisation, exécution, pilotage), mais introduit 4 phases originales supplémentaires : l'identification des différents acteurs fonctionnels impliqués dans le processus à modéliser (phase très courte !), la simulation, l'évaluation et l'amélioration.

Les phases de simulation, d'évaluation et d'amélioration sont collaboratives et basées sur l'utilisation de techniques « Serious game ». Elles regroupent l'ensemble des acteurs fonctionnels impliqués dans le processus métier et permettent de motiver cet ensemble d'acteurs à définir de façon collaborative, sans compétences particulières et de façon assez rapide, le modèle du processus métier. L'utilisation de techniques et outils issus du domaine de la gestion de qualité permet en outre d'une part une approche participative de tous les acteurs impliqués, et d'autre part une amélioration « a priori » du processus, avant même son exécution.

⁸M. Santorum, A. front, D. Rieu, N. Mandran, « Modélisation et optimisation participatives des processus métier », Enveloppe soleau déposée auprès de l'INPI, juillet 2010.

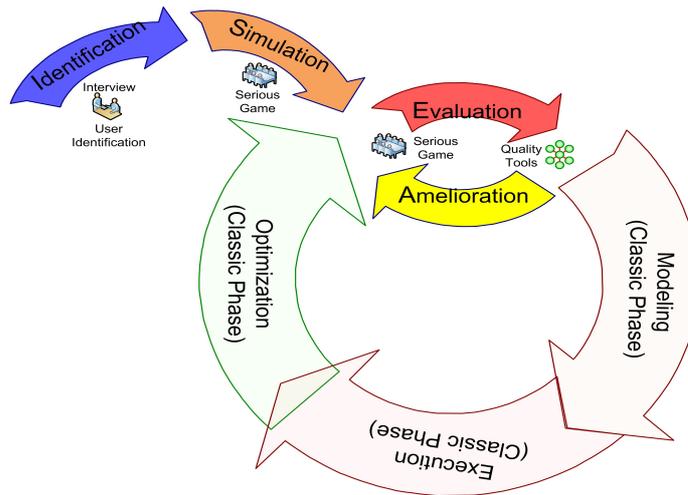


Figure 46 : La méthode ISEA

Dans la suite de cette section, je décris quelques éléments importants de cette méthode.

3.4.1. La phase de simulation

Dans la phase de simulation, l'objectif est d'obtenir, de manière ludique et coopérative, une description des activités réalisées et des documents échangés lors d'un processus métier en faisant jouer, via le jeu de rôles, les différents acteurs fonctionnels intervenant dans un processus métier afin qu'ils établissent, ensemble, la description du processus. Tous les acteurs fonctionnels internes identifiés lors de la phase d'identification sont « convoqués » pour une réunion d'une durée de deux heures environ. Les acteurs externes au processus (par exemple une agence de voyage) considérés comme des « boîtes noires » dont il est en général impossible de modifier l'organisation, ne sont pas convoqués.

Chaque participant joue son propre rôle dans le jeu et doit décrire ses activités quotidiennes. Chaque fois qu'un participant doit intervenir, il note toutes les actions qu'il doit effectuer sur un post-it jaune autocollant avec un feutre de couleur qui l'identifie. Il pose ensuite le post-it sur une grande nappe de papier blanche et fait une ou plusieurs flèches en indiquant à qui il passe le relais (autrement dit, la ou les personne(s) qui doi(ven)t assurer la suite du déroulement du processus). Une activité peut être composée de plusieurs actions ; une action est constituée d'un verbe conjugué à la première personne du singulier (par exemple *je demande*), d'un moyen (par exemple *par mail*) et si besoin d'un document (par exemple *un devis*). Certaines actions correspondent à des créations de documents. Dans ce cas, le participant remplit une petite fiche descriptive (nom du document, contraintes d'utilisation du document, type (électronique ou papier)...), colle une pastille de couleur sur la fiche du document (une couleur unique par document) et la pose sur le post-it. Si un participant a besoin d'un document préalablement créé, il colle sur son post-it une pastille de la couleur du document (voir Figure 47).

3.4.2. Phase d'évaluation

La phase d'évaluation a pour but d'identifier les difficultés rencontrées par les acteurs lorsqu'ils réalisent le processus, puis de proposer des actions d'amélioration possibles. Dans cette étape, il est primordial que les acteurs puissent échanger leurs idées. Les acteurs commencent par s'appropriier la représentation du processus obtenue (cf. Figure 48). Ensuite, ils identifient les difficultés associées aux différentes parties du processus de manière individuelle. Il s'agit de repérer les problèmes existants dans les processus pour les analyser, les hiérarchiser et trouver des pistes d'amélioration. Après ce travail individuel, l'animateur propose de grouper les difficultés similaires par catégories. Ce travail collectif permet aux participants de discuter sur les difficultés rencontrées et de les regrouper. Une fois ce regroupement terminé, l'animateur propose un libellé consensuel afin d'identifier la difficulté de façon générale. Ce groupe de difficultés est par la suite appelé difficulté (consensuelle). Dans le processus de gestion de la paye dans le cadre de l'Université Pierre Mendès France, de nombreuses difficultés ont été répertoriées telles que par exemple : « Problèmes d'organisation de l'espace de travail », « Logiciels non intégrés », etc.

Un brainstorming silencieux est ensuite imposé par l'animateur afin de réfléchir aux actions d'amélioration possibles pour résoudre les difficultés. La question du brainstorming est « Quelles actions faut-il mener pour améliorer les difficultés ? ». Chaque participant remplit un post-it par action d'amélioration proposée. Puis, comme pour les difficultés, l'animateur propose le regroupement des actions d'amélioration (cf. Figure 49, gauche) dans un diagramme des affinités. Quand le regroupement est terminé, l'animateur propose un libellé consensuel afin d'identifier chaque action d'amélioration (consensuelle). Les acteurs peuvent proposer autant d'actions d'amélioration que nécessaire.

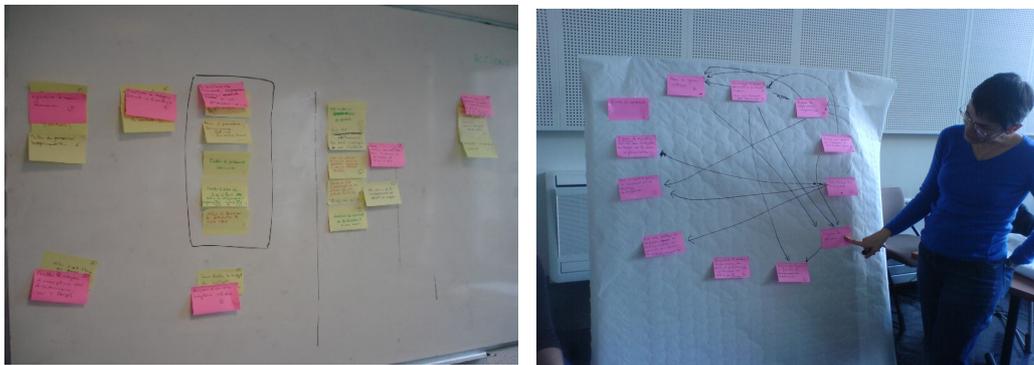


Figure 49 : Diagramme d'affinités des actions d'amélioration et diagramme des relations

Finalement, la dernière étape consiste à déterminer les priorités entre les différentes actions d'amélioration à réaliser. En utilisant un diagramme des relations (voir Figure 49, droite), les participants sont invités à répondre à la question suivante : « Pour faire l'action B, l'action A est-elle utile ? ». Par exemple : « Pour réorganiser l'espace de travail, recruter du nouveau personnel est-il utile ? ». En fonction de la majorité des réponses, l'ordre de mise en œuvre des actions d'amélioration est ainsi

déterminé. L'ordre obtenu est partiel, certaines actions pouvant être indépendantes des autres.

3.4.3. Phase d'amélioration

La phase d'amélioration a pour but de « rejouer » le processus selon les différentes actions d'amélioration proposées. Ainsi, une fois les actions d'amélioration identifiées et organisées, le processus est rejoué autant de fois que nécessaire afin d'identifier quantitativement l'impact des actions d'amélioration sur le processus. A partir de et en cohérence avec l'ordre de priorité des actions établi précédemment, une action d'amélioration est choisie. Une discussion est lancée afin de préciser concrètement la mise en œuvre et les conséquences de cette action d'amélioration.

Les participants rejouent le processus « par écart » : à partir de la représentation initiale du processus obtenue lors de la phase de simulation, chacun des participants modifie ses activités en imaginant que l'action d'amélioration est réalisée. Les activités initiales du processus peuvent donc être modifiées ou supprimées, et de nouvelles activités peuvent apparaître. Si une nouvelle difficulté est mise en évidence par un participant, l'animateur peut décider de l'ajouter dans la liste des difficultés existantes. Chaque participant vote sur son degré de satisfaction de l'action d'amélioration. La note obtenue est pondérée par le niveau d'importance des difficultés.

L'un des objectifs est d'améliorer au plus tôt les processus. Cette amélioration a priori ne supprime évidemment pas les phases de contrôle et d'optimisation « classiques » qui suivent traditionnellement la modélisation des processus, mais peut considérablement les réduire.

A la fin de la phase d'amélioration, la représentation simplifiée telle que présentée en Figure 48, mais qui a été rejouée en fonction des différentes actions d'amélioration, est traduite en un « vrai » modèle BPMN qui pourra être fourni en entrée de la phase de modélisation classique, afin d'être complété par un spécialiste BPMN dans le but de le rendre exécutable. La génération de ce modèle BPMN à partir de la représentation simplifiée fait actuellement l'objet du projet de master recherche de Mayra Carrion-Toro que je co-encadre avec Dominique Rieu, elle est basée sur des transformations de modèles dans une approche IDM.

3.4.4. Une méthode outillée

La Figure 50 montre une copie d'écran de l'outil ISEAsy qui permet d'utiliser la méthode sous sa forme informatisée. L'outil ISEAsy est déployé sous la forme d'une application web accessible en réseau et de façon simultanée par différents joueurs. Sur l'écran représenté en Figure 50 apparaissent les principaux concepts utilisés dans la méthode : les avatars pour représenter les joueurs (à gauche), ainsi que les post-its, les acteurs externes, les documents, les horloges (synonymes d'événements temporels), les boucles et les stops.



Figure 50 : Copie d'écran de l'outil ISEasy

3.4.5. La construction de la méthode par expérimentations

Un point important à souligner à propos de la méthode ISEA est qu'elle a été construite et validée grâce à de nombreuses expérimentations utilisateurs menées avec les acteurs fonctionnels eux-mêmes afin d'identifier au mieux leurs attentes. En effet, afin de vérifier que la méthode participative ISEA serait bien acceptée par les acteurs fonctionnels (non informaticiens) et qu'elle permettrait d'effectuer une modélisation des processus métier de façon ludique et rapide, nous avons très vite réalisé des expérimentations avec différentes équipes d'acteurs fonctionnels : le Laboratoire d'Informatique de Grenoble, les services centraux et informatiques de l'Université Pierre Mendès France, un service pédagogique de l'Université Joseph Fourier et un service pédagogique de l'école d'ingénieurs Grenoble-INP. Nous avons réalisé un protocole d'expérimentation précis, en demandant aux acteurs de suivre ce protocole. Mais nous nous sommes vite aperçus que certaines phases que nous avions prévues et qui nous semblaient inévitables, n'étaient pas adaptées aux attentes des acteurs fonctionnels.

L'exemple le plus flagrant concerne l'identification des buts lors de la phase d'amélioration. Il nous semblait en effet important, lors de l'émergence d'une difficulté dans la phase d'évaluation, de réfléchir sur le but intentionnel qui serait atteint si cette difficulté était résolue. Pour cela, nous avons défini le protocole présenté dans le Tableau 7, et nous avons demandé à une équipe du LIG de suivre ce protocole lors de phase d'amélioration pour le processus Gestion des Missions au LIG.

Expérience N°3

Sujets : un missionnaire, 3 assistantes (Valérie, Béatrice, Amélie), le chef d'équipe

Hypothèses :

1. Il est possible de déterminer les buts à atteindre en partant des difficultés rencontrées dans le processus actuel.
2. Il est possible de mesurer le niveau d'accomplissement des buts à atteindre en partant des difficultés rencontrées dans le processus actuel.

Matériel en début de séance :

- Doc 1 : 5 pochettes contenant le diagramme, les documents, les difficultés, les activités d'amélioration
- Doc 2 : 2 exemplaires des difficultés
- Doc 3 : 5 exemplaires Buts / activités d'amélioration
- Vidéo, portable, 5 feutres de couleurs

Matériel en cours de séance :

- Doc 4 : 10 exemplaires (un que les participants peuvent conserver dans leur pochette, un sur lequel ils notent les buts et nous le rendent) du fichier excel des buts avec colonnes supplémentaires permettant de mesurer le niveau d'accomplissement (A à D : tout à fait atteint, plutôt atteint, plutôt pas atteint, pas du tout atteint), et une colonne d'importance (important : O/N)
- Doc 5 : une diapositive sous forme électronique contenant les activités.

Activité 1 : Mise en situation (10 min)

Distribuer le Doc1

Revoir le diagramme (discussion collective) et les difficultés proposées lors de la séance 2

Activité 2 : Etablir les buts (40 min) – Hypothèse 1

Distribuer la liste des difficultés (Doc 2)

Travail par binôme ou trinôme sur les difficultés pour en faire des buts (15 min) :

Consigne : pour chaque difficulté, pourquoi est-ce important de la résoudre ?

En même temps, saisie par l'animateur des buts dans un fichier excel.

Synthèse et échange sur les buts identifiés.

Impression de la liste des buts (Doc 4)

Activité 3 : Vote sur les buts (niveau d'accomplissement et importance) Hypothèse 2 (10 min)

Réflexion individuelle sur les feuilles pour noter le processus. Chaque participant doit préciser pour chaque difficulté, ce qu'il pense du niveau d'accomplissement du but : pour chaque couple (difficulté/but), répondre aux questions suivantes :

- si la difficulté D était résolue, le but B serait-il : tout à fait atteint, plutôt atteint, plutôt pas atteint, pas du tout atteint ?
- Le but B vous semble-t-il important à atteindre (Oui/Non) ?

Les notes de chaque participant sont conservées et une note globale est calculée : cette note correspond au niveau d'accomplissement des buts à atteindre et permet de déterminer l'importance liée au fait de résoudre chaque difficulté par rapport aux buts du processus.

Tableau 7 : Un échec d'un protocole d'expérimentation pour la construction de la méthode ISEA

Force a été de constater que pour des acteurs fonctionnels, la notion de but n'était pas du tout facile à appréhender. Les acteurs fonctionnels confrontés à des difficultés n'étaient pas capables d'exprimer « pourquoi » il fallait résoudre cette difficulté, ils souhaitaient directement dire « comment » il fallait résoudre cette difficulté... Nous

avons donc revu le protocole en supprimant tout simplement les activités correspondant à l'identification des buts. La méthode ISEA présentée en Figure 46 constitue la version finale de la méthode validée par l'ensemble des expérimentations menées dans divers établissements universitaires grenoblois : le Laboratoire d'Informatique de Grenoble, les services centraux et informatiques de l'Université Pierre Mendès France, un service pédagogique de l'Université Joseph Fourier et un service pédagogique de l'école d'ingénieurs Grenoble-INP.

Un autre échec a été constaté avec le premier outil développé pour supporter la méthode ISEA. Ce prototype d'outil de modélisation était basé sur un moteur virtuel 3D tel que présenté dans la Figure 51. L'idée était ici de bénéficier des techniques issues des jeux vidéos pour permettre à un acteur fonctionnel de prendre le rôle d'un avatar qui se déplacerait dans le monde virtuel pour y collaborer avec les autres acteurs fonctionnels afin de poser des activités, de créer des documents, etc. L'objectif était évidemment de motiver l'acteur.

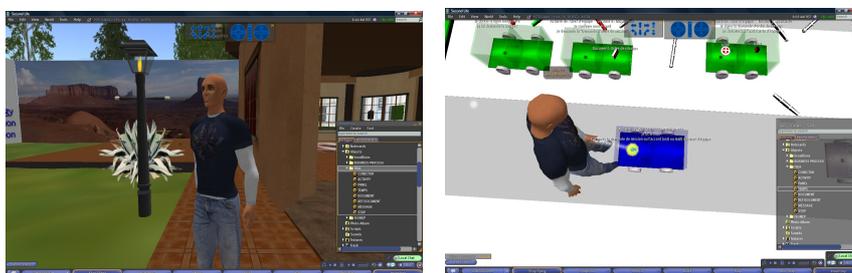


Figure 51 : Un participant au serious game et ses activités

Malheureusement, cet outil a été abandonné à l'issue des expérimentations réalisées, les acteurs fonctionnels des processus universitaires étant généralement des femmes, âgées d'une quarantaine d'années en moyenne et peu habituées et encore moins assidues à l'utilisation des jeux vidéos. Nous avons donc remplacé ce jeu vidéo par l'interface web simple et intuitive présentée en Figure 50 reprenant autant que faire se peut les symboles des post-its et documents réels.

3.5. SYNTHÈSE

Dans le cadre de l'ingénierie des méthodes, j'ai contribué à des recherches selon trois objectifs principaux :

- la définition d'une approche pour la construction de métamodèles de processus permettant à des organisations de définir le métamodèle de processus adapté à leur organisation ou à leurs projets, l'approche par métamodélisation présentant l'avantage de fournir un cadre structurant rigoureux. Cette approche a été définie dans le cadre de la thèse de Charlotte Hug, elle est guidée par un graphe conceptuel qui permet aux ingénieurs de méthodes de construire leur métamodèle de processus par assemblage de fragments de métamodèles de processus prédéfinis, et est instrumentée dans une approche IDM par des transformations de modèles.

– la définition de nouvelles méthodes de conception pour des nouveaux types de systèmes d'information. Ces méthodes de conception sont la plupart du temps basées sur des méthodes existantes et adaptent ou ajoutent certaines phases spécifiques liées aux nouvelles technologies ou formes d'organisation visées. Dans la thèse de Luz-Maria Priego-Roche, l'accent est plus particulièrement mis sur la phase intentionnelle qui permet de définir toutes les caractéristiques qu'un système d'information d'une organisation ouverte doit couvrir. Cette phase intentionnelle peut ainsi compléter le processus de développement Symphony, pauvre dans sa version originale sur le niveau intentionnel. Dans les travaux de thèse d'Ansem Ben Cheikh, une branche contextuelle ainsi que de nouveaux métamodèles sont ajoutés aux branches classiques d'une démarche en Y telle que Symphony (branche fonctionnelle, branche technique et branche centrale). Ils permettent à un concepteur de contextes d'être guidé lors de la définition du ou des contexte (s) à prendre en compte dans un système ubiquitaire sensible au contexte.

– la définition d'une méthode participative dont le but est de motiver l'ensemble des acteurs fonctionnels d'un processus métier à construire eux-mêmes et de façon collaborative la cartographie des processus métier d'une organisation en vue de les améliorer. Cette méthode est définie dans le cadre de la thèse de Marco Santorum, elle a été construite et validée par des expérimentations avec des acteurs fonctionnels de processus universitaires ; elle est applicable à n'importe quel type d'organisation.

4.1. BILAN

La Figure 52 dresse un panorama chronologique des différentes activités de recherche que j'ai menées depuis ma thèse soutenue en décembre 1997. J'ai principalement orienté mes recherches autour de 3 thèmes principaux :

- une meilleure spécification des produits (patrons, composants métier et frameworks) en vue de leur réutilisation,
- une meilleure formalisation des processus en vue de leur adaptation aux différents projets des organisations et aux organisations elles-mêmes. Cette formalisation a été initialisée dans le cadre de la collaboration avec la société Umanis autour de la méthode Symphony, et poursuivie dans le cadre des travaux sur la méta-modélisation des processus,
- la proposition de méthodes spécifiques, qui prend actuellement deux formes différentes : proposition de méthodes destinées à des types de systèmes d'information spécifiques tels que les systèmes d'information pervasifs ou encore les systèmes d'information des entreprises ouvertes, et proposition de méthodes utilisant des approches motivantes et participatives telles que les serious games.

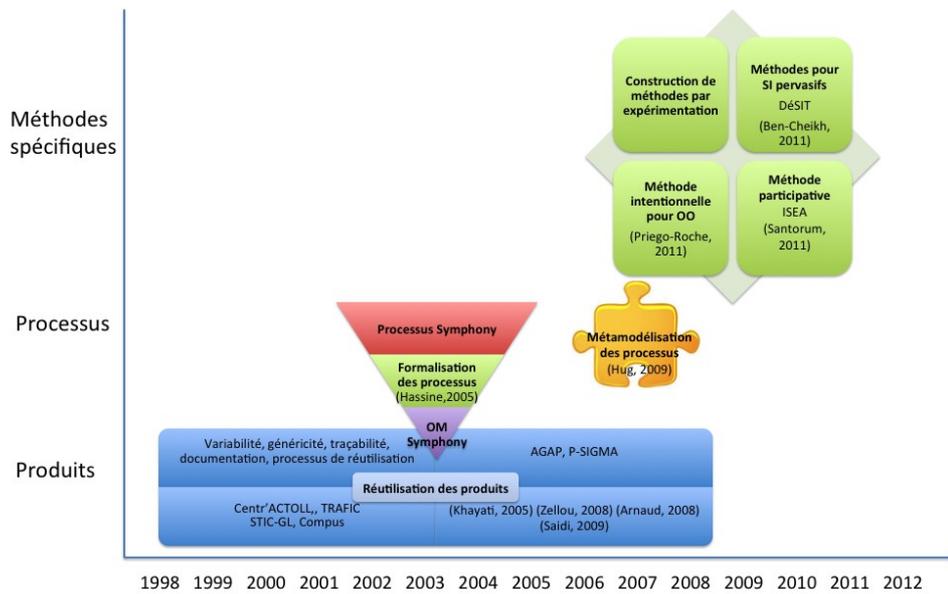


Figure 52 : Chronologie de mes différents thèmes de recherche

Les annexes A, B et C présentent de plus un résumé des différentes thèses, projets de DEA et de M2R et mémoires d'ingénieurs CNAM que j'ai co-encadrés ou encadrés ainsi que des différents projets auxquels j'ai participé.

4.2. PERSPECTIVES

Depuis la fin des années 2000, de nouveaux outils d'échange sont apparus sous le terme générique du web 2.0, du web social ou plus récemment de l'informatique sociale (en anglais, social computing). Ces outils visent le même objectif : permettre à des individus ou à des organisations de partager et de diffuser des connaissances et de créer des liens par le biais d'interactions sociales ou professionnelles : pour résumer, il s'agit d'une forme de socialisation de la connaissance. L'un des principaux challenges à l'heure actuelle est d'imaginer comment ces nouveaux outils pourraient devenir partie intégrante du système d'information des entreprises et des organisations. (Grim-Yefsah et al., 2010) propose par exemple une approche visant à améliorer la qualité d'un processus métier par l'analyse et l'utilisation des réseaux sociaux au sein des entreprises.

D'un autre côté, l'avènement et la vulgarisation des dispositifs mobiles (téléphones portables, assistants personnels...) modifient l'utilisation traditionnelle d'un système d'information : les systèmes d'information, jusqu'alors très « statiques », doivent désormais pouvoir s'adapter au dispositif utilisé, au profil de l'utilisateur et au contexte d'utilisation. Si d'un point de vue technologique, de telles technologies sont désormais relativement matures, l'un des enjeux sera, de la même manière que pour les outils du web 2.0, d'utiliser ces technologies pour concevoir des systèmes d'information plus adaptés aux besoins des organisations et de leurs acteurs.

La chronologie proposée par (Nurcan et Rolland, 2005) peut ainsi dorénavant être complétée par une sixième génération de systèmes d'information, que j'appellerai ici la *génération de la socialisation et de la mobilité* (cf. Figure 53). L'objectif commun est de tirer parti des nouvelles habitudes de travail individuelles pour accroître encore l'efficacité, voire l'efficacité des systèmes d'information et combiner les systèmes d'information personnels avec ceux des organisations.

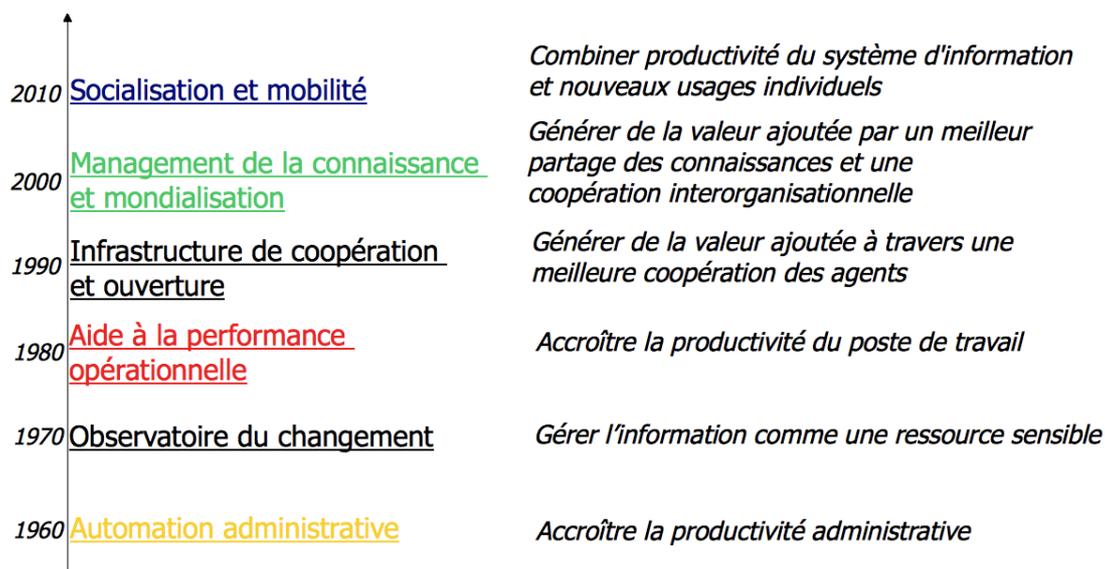


Figure 53 : Evolution des systèmes d'information

Ces nouveaux usages individuels ouvrent la voie vers des systèmes d'information intégrant les besoins de l'organisation et ceux des individus qui la composent (dans la suite, j'utiliserai le terme de « système d'information social »). Il est alors primordial d'impliquer les différentes parties prenantes dans la conception de ces systèmes d'information sociaux. Pour ce faire, les méthodes de conception des systèmes d'information doivent devenir plus attractives et encore plus coopératives.

Mes perspectives de recherche actuelles visent ainsi les trois axes suivants :

- d'une part, **l'ingénierie de méthodes de conception de systèmes d'information sociaux** visant à proposer des méthodes pour la conception de systèmes d'information sociaux,
- d'autre part, **l'ingénierie de méthodes de conception sociales** visant à intégrer dans les méthodes de conception, les nouvelles technologies d'échange afin de concevoir des systèmes d'information de manière attractive et coopérative,
- enfin, **l'ingénierie sociale de méthodes de conception**. Le but est ici d'intégrer les nouvelles technologies d'échange lors de la définition des méthodes, afin de permettre de construire des méthodes adaptées aux besoins des différents acteurs qui utiliseront la méthode.

Chacun de ces trois axes prend bien évidemment appui sur les avancées des recherches initialisées dans les dernières thèses que j'ai co-encadrées, qu'elles soient récemment soutenues ou encore en cours.

4.2.1. Vers des méthodes de conception de systèmes d'information sociaux

L'usage grandissant des nouvelles technologies d'échange ainsi que des dispositifs mobiles a provoqué une véritable mutation des systèmes d'information. Ainsi, les systèmes d'information doivent aujourd'hui être sensibles au contexte non seulement spatio-temporel et matériel, mais aussi humain : il n'est plus envisageable aujourd'hui de construire des systèmes d'information sans que ceux-ci puissent s'adapter à l'utilisateur, à ses préférences, à son contexte d'utilisation (contexte en mobilité, mais aussi contexte professionnel ou personnel). Il est donc nécessaire de combiner les différents systèmes d'information personnels avec ceux des organisations, mais également entre eux.

Les premiers travaux nécessaires à court terme concernent donc la proposition de métamodèles complètement génériques pour l'adaptation au contexte. Ainsi, tous les travaux visant le développement de SI adaptables au contexte définissent leur propre métamodèle, ce qui conduit à l'heure actuelle à l'existence d'une pléthore de métamodèles spécifiques selon la nature de l'adaptation au contexte visée : adaptation de la présentation, adaptation des informations délivrées, etc. Il devient donc indispensable de disposer de métamodèles génériques d'adaptation au contexte. Dans les travaux de thèse d'Ansem Ben-Cheikh, un premier pas a été fait dans ce sens, mais les expérimentations que nous avons réalisées avec des concepteurs issus de

différentes spécialités (GL, IHM, SI entres autres) ont montré la difficulté d'obtenir des métamodèles suffisamment génériques et acceptés par toutes les spécialités.

Une autre perspective envisageable à court terme concerne l'extension de la méthode E-CARE pour l'adaptation des processus métier en fonction du contexte. En effet, la méthode E-CARE (thèse d'Ansem Ben-Cheikh) est actuellement destinée à la conception de systèmes d'information pervasifs utilisant une technologie à base de « diffusion/réception de flux ». Les systèmes d'information visés ont essentiellement pour but la diffusion d'informations adaptées à l'utilisateur en fonction de son contexte et de ses préférences. Un domaine d'application privilégié est par exemple le domaine des transports publics, et plus particulièrement l'information des voyageurs. Néanmoins, nous n'avons que très peu abordé dans la méthode E-CARE l'adaptation des processus métier en fonction du contexte. En effet, le domaine des transports publics comme de nombreux autres domaines d'application (santé, éducation, commerce, etc.) engendre une autre problématique importante liée à la prise en compte du contexte : l'adaptation des processus métier. Ainsi, il n'est pas difficile d'imaginer qu'en fonction du contexte, certains rôles sont amenés à évoluer, des droits d'accès à être modifiés, des activités à être supprimées ou au contraire rajoutées, etc. Je souhaite donc poursuivre et adapter les résultats de la thèse d'Ansem Ben-Cheikh afin de prendre en compte la variabilité des processus métier selon le contexte. J'ai démarré ces travaux dans le cadre d'une collaboration avec Christine Verdier et Aurélien Faravelon (Faravelon et al., 2011) où nous mettons en oeuvre une approche de type « Privacy by design » afin de définir dans un processus métier, les différentes politiques d'accès à prendre en compte et à faire évoluer en fonction du contexte.

La proposition de métamodèles génériques de contexte et la prise en compte de l'adaptation des processus métier seront de premières briques pour la définition de méthodes de conception de systèmes d'information sociaux, mais de nombreuses questions devront également trouver des réponses à l'ère de l'informatique sociale : quelles conséquences liées à l'intégration des nouveaux outils d'échange dans les systèmes d'information ? comment intégrer les systèmes d'information personnels entre eux et avec ceux des organisations tout en respectant la séparation des préoccupations entre vie personnelle et vie professionnelle et la confidentialité des informations ? ...

4.2.2. Vers des méthodes de conception sociales de systèmes d'information

Concernant l'axe « Ingénierie de méthodes de conception sociales », les travaux liés à la méthode ISEA (thèse de Marco Santorum) apportent également de nombreuses perspectives de recherche à moyen terme.

D'une part, la méthode ISEA est actuellement dédiée à la cartographie et à l'évolution de processus métier existants pour lesquels les acteurs fonctionnels ont une bonne connaissance du processus analysé. Néanmoins, elle n'est pas utilisable en l'état pour des processus métier nouveaux ou mal définis comme les processus métier des organisations ouvertes. La thèse de Luz-Maria Priego-Roche ayant pour but d'identifier les besoins nécessaires à la conception d'un système d'information pour

une entreprise ouverte, il apparaît naturel de combiner les résultats de cette thèse avec ceux de la méthode ISEA, afin de définir une méthode de modélisation des processus métier peu ou pas définis, par exemple ceux des organisations ouvertes qui se mettent en place en réponse à une opportunité du marché. Ainsi, les modèles intentionnels définis dans la thèse de Luz-Maria Priego-Roche seront une aide fondamentale pour que les acteurs des différentes organisations impliquées puissent concevoir et optimiser de manière collaborative, les processus métier liés à l'organisation ouverte. De plus, les nouvelles technologies d'échange telles que les réseaux professionnels ou les forums pourront permettre aux différentes organisations et acteurs impliqués de se connaître et d'échanger autour de la mise en place des processus métier à mettre en place. Nous débuterons des recherches sur cet aspect dans le cadre du projet « Analyse et modélisation du processus de recyclage des Déchets d'Equipements Electriques et Electroniques, en vue d'optimiser sa gouvernance ». Ce projet, accepté dans le cadre de la SFR INNOVACS, débutera en octobre en collaboration avec les laboratoires G-SCOP (Laboratoire des Sciences pour la Conception, l'Optimisation et la Production) et CERAG (Centre d'Etudes et de Recherches Appliquées à la Gestion). Son domaine d'application concerne le traitement des Déchets d'Equipements Electriques et Electroniques pour lequel le nombre d'organisations impliquées et d'acteurs concernés est important et les processus mal définis.

D'un point de vue outillage, nous avons démarré cette année dans le cadre d'un projet de TER, une recherche sur l'utilisation des outils de type table virtuelle dans la méthode ISEA. Les outils de type table virtuelle, plus connus sous le nom de « Tabletop », sont des dispositifs modernes se présentant sous la forme de tables sensibles au toucher. Cette technologie, relativement nouvelle, est encore peu répandue principalement à cause de son prix élevé : les deux principales tables virtuelles, la DiamondTouch (commercialisée en 2006 par Mitsubishi Electric Research Laboratories) et la Surface (développée par Microsoft entre 2008 et 2011), coûtent actuellement respectivement environ 10000 \$ et 7500 \$. Nous avons néanmoins eu l'opportunité de tester l'utilisation de la DiamondTouch pour la méthode ISEA en développant une application dédiée (stage de TER de Martin Matuszek, soutenu en juin 2011). Cette application n'a malheureusement pas pu être terminée pour des raisons techniques et fonctionnelles. En particulier, il n'est actuellement pas possible de faire collaborer plus de quatre acteurs autour d'une même table, ce nombre étant insuffisant dans le cadre des processus métier étudiés dans la méthode ISEA (processus universitaires où le nombre d'acteurs impliqués est généralement bien supérieur à quatre). Néanmoins, les premiers prototypes développés ont montré des fonctionnalités intéressantes pour la méthode ISEA, en particulier le fait que les acteurs, regroupés autour de la table virtuelle, puissent toucher et manipuler des objets virtuels (par exemple des post-its virtuels) pratiquement de la même façon que dans la version manuelle de la méthode ISEA. Ainsi, une telle table virtuelle peut se substituer aux équipements traditionnels (nappe et post-its papier tels que ceux présentés en Figure 47, mais aussi ordinateur individuel et interface web telle que celle présentée en Figure 50) tout en stimulant, par le biais de sa nouveauté technologique, la collaboration entre les différents acteurs. Il est donc intéressant de poursuivre ces travaux au fur et à mesure des avancées technologiques sur les tables virtuelles. Evidemment, des expérimentations devront être menées afin de tester l'acceptation de tels dispositifs par les acteurs fonctionnels des processus métier.

De par son originalité, l'approche « Serious Game pour la modélisation » est un axe de recherche intéressant à poursuivre. Ainsi, ma « casquette » d'enseignante en modélisation m'amène à penser que cette approche, qui a déjà largement fait ses preuves pédagogiquement dans de très nombreux domaines (mathématiques, sciences physiques, langues vivantes...), mais paradoxalement très peu encore en informatique, est intéressante dans le domaine de l'enseignement de la modélisation. Une telle approche pourrait ainsi faciliter et motiver « nos étudiants » dans l'apprentissage de la modélisation. Ainsi, je suis convaincue qu'un outil de type Serious Game tel que celui développé dans le cadre de la thèse de Marco Santorum cf. Figure 51) et pour l'instant abandonné car non destiné au public avec lequel nous avons expérimenté la méthode ISEA (des femmes, secrétaires ou administratives, âgées d'une quarantaine d'années en moyenne et peu habitués à l'utilisation de jeux vidéos), serait un atout majeur dans le cadre d'un cours destiné à des étudiants jeunes et assidus aux jeux vidéos. Le challenge consiste ici à bien définir les objectifs et le scénario pédagogique à mettre en œuvre ; une collaboration avec l'équipe Metah du LIG spécialiste des Environnements Informatiques pour l'Apprentissage Humain pouvant tout à fait être envisagée dans ce but.

De manière plus générale, l'ingénierie de méthodes sociales de conception concerne l'intégration des nouveaux outils tels que les forums de discussion, les messageries instantanées, les blogs ou encore les jeux vidéos dans les méthodes de conception. Le but est ici encore de motiver et de faire collaborer les concepteurs d'applications sur tous les aspects liés au développement d'une nouvelle application. En effet, les visites de stagiaires et d'alternants que j'ai pu effectuer ces trois dernières années, m'ont permis de suivre l'évolution des méthodes agiles dans les entreprises. Aujourd'hui, l'acceptation de ces méthodes dans les entreprises est évidente, elle est liée d'une part à la fréquence des réunions de projets organisées quasi quotidiennement sous la forme de brainstormings collaboratifs entre les différents collaborateurs impliqués, d'autre part à la validation régulière des besoins du client grâce à différentes maquettes. L'utilisation d'une méthode agile dans une entreprise renforce ainsi l'implication des différents collaborateurs qui mettent tout en œuvre pour respecter le planning et les contraintes établies par l'ensemble de l'équipe. En adéquation avec la génération « Socialisation et mobilité », je souhaite aller plus loin dans l'implication et la motivation des différents acteurs en étudiant comment intégrer l'ensemble des technologies de communication telles que la messagerie instantanée, les réseaux sociaux, les jeux vidéos en réseau, etc. dans les méthodes de conception, le but étant de rendre le développement d'une application encore plus collaboratif et plus attractif pour l'ensemble des acteurs impliqués. Ceci rejoint l'idée évoquée dans de nouvelles approches du domaine de l'Interface Homme-Machine (Klöckner et al., 2010), visant à rajouter des propriétés « hédoniques » (stimulation, beauté, etc.) à l'application afin de déclencher des émotions positives telles que de la joie ou du plaisir chez l'utilisateur de l'application. Cette approche, basée sur des théories psychologiques, soutient le fait que l'utilisateur de l'application, en éprouvant plus de joie et de plaisir à son travail, sera plus motivé, plus concentré et ainsi plus créatif. C'est aussi le pari que je tenterais de franchir dans les prochaines années, en introduisant des propriétés attractives et stimulantes dans les méthodes de conception de systèmes d'information ou de processus métier.

4.2.3. Vers une ingénierie sociale de méthodes de conception de systèmes d'information

L'axe *Ingénierie sociale de méthodes de conception* fait suite aux travaux menés dans le cadre de la construction de la méthode ISEA et a pour objectif la définition de méthodes adaptées aux besoins des organisations et de leurs acteurs de manière consensuelle et participative. En effet, les expérimentations menées lors de la construction de la méthode ISEA avec les différents acteurs de différentes organisations (acteurs fonctionnels, mais aussi décideurs et chefs de projets) nous ont permis de définir une méthode adaptée aux besoins et aux compétences de ces acteurs. Ainsi, la méthode ISEA pourrait être considérée comme une sorte de DSL (Domain Specific Language), spécifique à l'organisation visée, et composée de modèles et de processus adaptés aux acteurs de l'organisation et définis de manière consensuelle. Néanmoins, l'approche utilisée est restée principalement basée sur des techniques issues du domaine de la gestion de la qualité, permettant ainsi une collaboration active entre les différents acteurs sous la forme de focus group.

Il apparaît important aujourd'hui d'élargir cette approche en se basant en particulier sur les possibilités offertes par les nouvelles technologies d'échange. L'idée sera ici d'étudier comment intégrer les nouveaux outils d'échange (réseaux sociaux, forums, serious game, etc.) afin de motiver les différents acteurs d'une organisation à définir eux-mêmes, via une approche participative et consensuelle, une méthode adaptée à leur organisation et à leurs habitudes de travail.

Beaucoup de pas restent évidemment à franchir pour atteindre cet objectif ambitieux. D'autres aspects doivent également être intégrés, comme par exemple une meilleure intégration dans les méthodes de conception de la gestion de projets des entreprises et de la gestion de la qualité, une meilleure gestion des informations et des connaissances des organisations, la proposition d'outils adaptés supportant les méthodes sans oublier une meilleure gestion de la sûreté et à la sécurité des données liées aux organisations et à leurs acteurs. Néanmoins, la proposition de techniques simples et attractives permettant aux différents acteurs d'une organisation de définir eux-mêmes, de façon consensuelle et motivante, des modèles et des méthodes adaptés à leurs habitudes de travail et aux besoins de leurs organisations, sera peut-être un premier pas pour passer de l'ère « ingénierie des modèles » à l'ère « industrie des modèles », et pourquoi pas, d'une ingénierie des méthodes encore souvent pauvre et archaïque vers une véritable « Industrie des méthodes ».

-A-

Achilleos et al., 2010

A. Achilleos, K. Yang, N. Georgalas. Context modeling and a context-aware framework for pervasive service creation : A model driven approach. *Pervasive and Mobile Computing*, vol. 6, pp. 281-296, 2010.

Alexander et al., 1977

C. Alexander, S. Ishikawa, M. Silverstein, *A Pattern Language : Towns, Buildings, Construction*. Oxford University Press, 1977.

Alexander, 1979

C. Alexander, *The Timeless Way of Building*, Oxford University Press, 1979.

Appleton, 1997

N. Appleton, *Patterns and Software – Essential Concept and Terminology*, <http://www.enteract.com/~bradapp/docs/patterns-intro.html>, 1997.

Annoni et al., 2006

E. Annoni, F. Ravat, O. Teste, G. Zurfluh, *Méthode de Développement des Systèmes d'Information Décisionnels : Roue de Deming*, Congrès INFORSID, 2006.

Ambler, 1998

S.W. Ambler, *Process Patterns : building Large Scale Systems using Object Technology*, SIGS Books, Cambridge University Press, Décembre 1998.

Arnaud, 2008

N. Arnaud, *Fiabiliser la réutilisation des patrons par une approche orientée complétude, variabilité et généricité des spécifications*, Thèse de doctorat de l'Université Joseph Fourier, soutenue le 13 octobre 2008.

Arnaud et al., 2004

N. Arnaud, A. Front, D. Rieu, "Une approche par métamodélisation pour l'imitation des patrons", Congrès INFORSID'04, Biarritz, Mai 2004.

Arnaud et al., 2005

N. Arnaud, A. Front, D. Rieu, « Deux opérations pour l'intégration d'imitations de patrons », Congrès INFORSID'05, Grenoble, France 24-27 Mai 2005.

Arnaud et al., 2006a

N. Arnaud, A. Front, D. Rieu, "Expression et usage de la variabilité dans les patrons de conception", Congrès INFORSID'06, Hammamet, 1-3 Juin 2006.

Arnaud et al., 2006b

N. Arnaud, A. Front, D. Rieu, "Processus d'imitation pour patrons de conception à variantes", 2èmes Journées IDM'06, Lille, 26-28 Juin 2006.

Arnaud et al., 2007a

N. Arnaud, A. Front, D. Rieu, *Expression et usage de la variabilité dans les patrons de conception*, Revue ISI-RSTI série ISI, *Conception des systèmes d'information : patrons et spécifications formelles*, Vol. 12, numéro 4, Hermès, 2007.

Arnaud et al., 2007b

N. Arnaud, A. Front, D. Rieu, *Expressing variability for patterns re-use*. Poster dans RCIS'2007, The first International Conference on Research Challenges in Information Science, 2007.

Arnaud et al., 2009

N. Arnaud, A. Front, D. Rieu, S. Dupuy-Chessa, Un processus d'imitation de patrons supporté par l'approche IDM, 5èmes journées Ingénierie Dirigée par les Modèles – IDM'2009, Mars 2009.

Australian Standard, 2004

Australian Standard metamodel for software development methodologies, AS 4651—2004, August 2004.

Ayed et al., 2007

D. Ayed, D. Delanote, Y. Berbers. Mdd approach for the development of context-aware applications. In CONTEXT'07. Springer. 2007.

-B-**Bachmann et Bass, 2001**

F. Bachmann, L. Bass. Managing variability in software architecture. *Software Engineering Notes*, vol. 26, n°3, pp. 126-132. 2001.

Barnett et al., 1994

W. Barnett, A. Presley, M. Johnson, D.H. Liles, « An architecture for the Virtual Enterprise », IEEE International Conference on Systems, Man, and Cybernetics, Humans, Information and Technology, Oct 1994, vol. 1, 2-5, p. 506 - 511, 1994.

Barthez et al., 2000

B. Barthez, A. Front-Conte, D. Rieu, *Intégrations d'imitations de patrons pour la spécification de cadrage*, Congrès INFORSID'00, Lyon, Mai 2000.

Basole et Rouse 2008

R. Basole, W. Rouse, « Complexity of service value networks : Conceptualisation and empirical investigation », *IBM Systems Journal*, vol. 47, n° 1, p. 53-70, 2008.

Beck, 1999

K. Beck. *Extreme Programming Explained: Embrace Change*. Addison-Wesley Professional, Longman Publishing Co., Inc. Boston, Massachusetts, 1999.

Ben-Cheikh et al., 2009a

A. Ben-Cheikh, D. Rieu, A. Front, "A method for business process reverse-engineering based on a multi-view metamodel", Conference on ENTERprise Information Systems (CENTERIS), Ofir, Portugal, Octobre 2009.

Ben-Cheikh et al., 2009b

A. Ben Cheikh, Y. Tourasse, A. Front, S. Coulondre, J-P. Giraudin, *Vers une approche orientée processus métier pour les Systèmes d'Information de Transport*, Première Conférence Francophone sur les Technologies de l'Information, de la Communication et de la Géolocalisation dans les Systèmes de Transports.

Ben-Cheikh et al., 2009c

A. Ben Cheikh, D. Rieu, A. Front, « Une méthode de rétro-ingénierie des processus métier basée sur un métamodèle multi-vues », INFORSID'09, Toulouse, 26-29 mai 2009.

Ben-Cheikh et al., 2010a

A. Ben-Cheikh, A. Front, S. Coulondre, J-P. Giraudin, “Event Based Modeling for Context-Reactive Information Systems”, 6th IEEE International Conference on Signal Image Technology and Internet Based Systems (SITIS'10), Kuala-Lumpur, Malaisie, Décembre 2010.

Ben-Cheikh et al., 2010b

A. Ben-Cheikh, A. Front, S. Coulondre, J-P. Giraudin, “Event-based modeling for ubiquitous information systems”, UbiMob'10, 6^{èmes} journées francophones Ubiquité et Mobilité, Lyon, 7-9 juin 2010.

Ben-Cheikh et al., 2010c

A. Ben-Cheikh, A. Front, S. Coulondre, J-P. Giraudin, “Une modélisation événementielle des SI ubiquitaires”, INFORSID'10, Marseille, 25-28 mai 2010.

Berrut et Front, 2000

C. Berrut, A. Front-Conte, *Patterns Retrieval System: a first attempt*, 5th International Conference on Applications of Natural Language to Information Systems fKetf(NLDB'2000), Versailles, Juin 2000.

Boehm, 1988

R. Boehm, *A Spiral Model of Software Development and Enhancement*, IEEE Computer, vol.21, n°5, May 1988.

Booch, 1991

G. Booch, *Object Oriented Analysis and Design with Application*, Benjamin / Cummings, 1991.

Borchers, 2000

J.O. Borchers. A Pattern Approach to Interaction Design. Symposium on Designing Interactive Systems, pp. 369-378. 2000.

Borne et Revault, 1999

I. Borne, N. Revault, *Comparaison d'outils de mise en œuvre de design patterns*. L'Objet, Vol.5, n°2, 1999).

-C-**Casanave, 1996**

C. Casanave, Business Object Architectures and Standards, Data Access Corporation, Miami, USA, 1996.

Cauvet, 2006

C. Cauvet, *Modélisation des processus d'ingénierie des systèmes d'information*, chapitre dans *Encyclopédie de l'informatique et des systèmes d'information*, Vuibert, 2006.

Ceri et Fraternali, 1997

S. Ceri, P. Fraternali, *Designing Database Applications with Objects and Rules – The IDEA Methodology*, Addison-Wesley.

Cheng et al., 2005

S. Cheng, X. Xu, G. Wang, Q. Li, An agile method of modeling business process simulation for virtual enterprises, IEEE International Conference on e-Business Engineering, 2005.

Clauss, 2001

M. Clauss, *Generic modeling using UML extensions for variability*, *OOPSLA 2001, Workshop on Domain Specific Visual Languages*, pages 11-18, Septembre 2001.

Coad et al., 1996

D.Coad, P. North, M. Mayfield, *Objects Models – Strategies, Patterns and Applications*. Yourdon Press Computing Series, 1996.

Cocquebert et al., 2010

E. Cocquebert, D. Trentesaux, C. Tahon, *WISDOM: A website design method based on reusing design and software solutions*. *Information and Software Technology, Volume 52, Issue 12*, pp. 1272–1285, December 2010.

Couturier, 2007

Couturier, Un système de patterns dédiés à l'analyse, à la conception et à l'implémentation des Systèmes d'Information Coopératifs, *Ingénierie des Systèmes d'Information*, Vol. 12, No. 4, 2007, pp. 45-68.

Cummins, 1999

F. Cummins, *OMG Business Object Concept – BOTF*, White paper, EDS-BOM/99-12-42, 1999.

Curtis et al., 1992

B. Curtis, M.I. Kellner, J. Over, *Process modeling*, *Commun. ACM, ACM*, 1992, n° 35.

Czarnecki et Eisenecker, 2000

K. Czarnecki, U.W. Eisenecker, *Generative Programming – Methods, Tools and Applications*, Addison-Wesley, 2000.

-D-**Davidow et al., 1992**

Z.H. Davidow, M.S. Malone, *The virtual corporation : Structuring and revitalizing the corporation for the 21st century*, T. Harper Business, NY, 1992.

Davis, 2001

M. Davis, "Struts, an OpenSource MVC implementation. Manage complexity in large Web with this servlets and JSP framework", <http://www-106.ibm.com/developerworks/library/j-struts/>, février 2001.

Dupuy et al., 2000

S. Dupuy, A. Front-Conte, C. Saint-Marcel, « Chapitre 6 : Using UML with a Behaviour-Driven Method », dans "Software Specification Methods - An Overview Using a Case Study", Springer, 2000.

-E-**Eden, 2000**

A.H. Eden, *Precise Specification of Design Patterns and Tool Support in their Application*, Thèse de Doctorat, Tel Aviv, 1997.

-F-

Fahkri et Cauvet, 2011

M. Fahkri, C. Cauvet, Méthodes d'ingénierie de systèmes d'information à base de services, *Revue Ingénierie des Systèmes d'Information* vol. 16, n° 2, pp. 43-66, 2011.

Faravelon et al., 2011

A. Faravelon, C. Verdier, **A. Front**, *Towards a Business-Centric Definition of Security Policies. 5th IEEE Int Conf on Research Challenges in Information Science (RCIS'11)*, 2011.

Finkelstein et al., 1990

A. Finkelstein, J. Kramer, M. Goedicke, *ViewPoint oriented software development, in 3rd International Workshop on Software Engineering and Its Applications*, pp. 374—384, 1990.

Fowler, 1997

M. Fowler, *Analysis Patterns: reusable objects models*, Addison-Wesley, Reading MA, 1997.

Front, 1997

A. Front, « *Développement de systèmes d'information à l'aide de patrons – Application aux bases de données actives* » Thèse de doctorat de l'Université Joseph Fourier, Grenoble, soutenue en Décembre 1997.

Front, 1999

A. Front-Conte, "SCalP, un langage de patrons pour les aspects réactifs des systèmes d'information", *Revue l'Objet*, numéro spécial "Patrons Orientés Objets", volume 5, numéro 2, Hermès, pp 147-175, Juin 1999.

Front et al., 1999

A. Front, J-P. Giraudin, D. Rieu, C. Saint-Marcel, *Réutilisation et patrons d'ingénierie*, Chapitre 4 du livre "Génie Objet - Analyse et Conception de l'Evolution", Editions Hermès, 1999, pp. 91-136.

Front-Conte et al., 2001a

A. Conte, J-P. Giraudin, I. Hassine, D. Rieu, *Un environnement et un formalisme pour la définition, la gestion et l'application de patrons*, *Revue ISI*, numéro spécial « *Formalismes et Modèles pour les Systèmes d'Informations* », Vol. 6, numéro 2, Hermès.

Front-Conte et al., 2001b

A. Conte, M. Fredj, J.P. Giraudin, D. Rieu, *P-Sigma : un formalisme pour une représentation unifiée de patrons*, Congrès INFORSID'01, Genève, Mai 2001.

Front-Conte et al., 2001c

A. Conte, I. Hassine, J.P. Giraudin, D. Rieu, *AGAP : un Atelier de Gestion et d'Application de Patrons*. Congrès INFORSID'01, Genève, Mai 2001.

Front-Conte et al., 2002

A. Front-Conte, M. Fredj, I. Hassine, J-P. Giraudin, D. Rieu "AGAP: an environment to design and apply patterns", *IEEE International Conference on Systems, Man and Cybernetics (IEEE SMC)*, Hammamet, Tunisia, October 6-9, 2002.

Front-Conte et al., 2002b

A. Front-Conte, M. Fredj, I. Hassine, JP. Giraudin, D. Rieu, "A tool and a formalism to design and apply patterns", 8th International Conference on Object-Oriented Information Systems (OOIS 2002), Montpellier, France, September 2-5, 2002.

Front-Conte et al., 2002c

A. Front-Conte, J-C. Freire Junior, J-P. Giraudin, I. Hassine, D. Rieu, "A tool and a formalism to design and apply patterns", Second Latin American Conference on Pattern Languages of Programming (SugarLoafPLoP 2002), Itaipava, Rio de Janeiro, Brazil, August 5-7, 2002.

-G-

Gamma et al., 1994

E. Gamma, R. Helm, R. Johnson, J. Vlissides. Design Patterns : Elements of Reusable Object-Oriented Software. Addison-Wesley Publishing Company, 1994.

Grim-Yefsah et al., 2010

M. Grim-Yefsah, C. Rosenthal-Sabroux, V. Thion-Goasdoué, Évaluation de la qualité d'un processus métier à l'aide d'informations issues de réseaux informels, Revue Ingénierie des Systèmes d'Information, vol. 15, n° 6, pp. 63-83, 2010.

Gzara et al., 2000

L. Gzara, D. Rieu, M. Tollenaere, Patterns Approach to Product Information Systems Engineering, Requirements Engineering Journal, Vol. 5, n° 3, pp. 157-179, 2000.

Guzélian et al., 2004

G. Guzélian, C. Cauvet, P. Ramadour, Conception et réutilisation de composants : une approche par les buts, Congrès INFORSID 2004, pp. 179-174, 2004.

-H-

Harel, 1987

D. Harel, *Statecharts: A Visual Formulation for Complex Systems*, Science of Computer Programming, vol. 8, n° 3, 1987.

Harmsen, 1997

A.F. Harmsen, *Situational Method Engineering*. Moret Ernst & Young, 1997.

Hassine, 2005

I. Hassine, *Spécification et formalisation des démarches de développement à base de composants métier : la démarche Symphony*, Thèse de doctorat de l'Institut National Polytechnique de Grenoble, soutenue le 24 septembre 2005.

Hassine et al., 2005a

I. Hassine, E. Jausseran, D. Rieu, A. Front, "Objets métier et composants techniques dans la méthode Symphony", Revue ISI-RSTI série ISI, numéro spécial "Méthodes avancées de développement des systèmes d'information", volume 10 – n°6/2005.

Hassine et al., 2005b

I. Hassine, D. Rieu, F. Bounaas, E. Jausseran, A. Front, J-P. Giraudin, « Chapter 4 : Méthodes de développement centrées composants », dans "Ingénierie des Composants - Concepts, techniques et outils", Vuibert, pp 87-131, 2005.

Hassine et al., 2004

I. Hassine, D. Rieu, A. Front, L. Tastet, "An Information System Development Tool Based on Pattern Reuse", Poster Conférence ICEIS'04, Porto, Avril 2004.

Hassine et al., 2003a

I. Hassine, D. Rieu, A. Front-Conte, F. Bounaas, "Intégration des processus métier pour une meilleure spécification des composants métiers", Conférence CopSTIC'03, Rabat, Maroc, 11-13 Décembre 2003.

Hassine et al., 2003b

I. Hassine, D. Rieu, A. Front-Conte, F. Bounaas, "Modélisation et Formalisation d'une démarche de développement à base de composants métier", ICSSEA'2003 -16th International Conference Paris, France, 2-4 Décembre 2003.

Henderson-Sellers et Gonzales-Perez, 2011

B. Henderson-Sellers, C. Gonzalez-Perez, Towards the Use of Granularity Theory for Determining the Size of Atomic Method Fragments for Use in Situational Method Engineering, IFIP WG8.1 working conference, ME'2011, Paris, 20-22 avril 2011.

Henricksen et Indulska, 2006

K. Henricksen, J. Indulska. Developing context-aware pervasive computing applications : models and approach. Pervasive and Mobile Computing, vol. 2. 2006.

Hug, 2009

C. Hug, « Méthode, modèles et outils pour la métamodélisation des processus d'ingénierie des systèmes d'information », thèse de doctorat de l'Université Joseph Fourier, soutenue le 20 octobre 2009.

Hug et al., 2007a

C. Hug, A. Front, D. Rieu, « Ingénierie des processus : une approche à base de patrons », INFORSID'07, Perros-Guirec, 22-25 mai 2007.

Hug et al., 2007b

C. Hug, A. Front, D. Rieu, « Ingénierie des processus - Un patron pour la métamodélisation de concepts et de catégories de concepts », 1^{er} atelier MADSI (Méthodes Avancées de Développement de Systèmes d'Information), organisé dans le cadre du congrès INFORSID'07, Perros-Guirec, 22 mai 2007.

Hug et al., 2008a

C. Hug, A. Front, D. Rieu, Ingénierie des processus : une approche à base de patrons, Revue ISI-RSTI, série ISI, Vol. 13, numéro 4, 2008.

Hug et al., 2008b

C. Hug, A. Front, D. Rieu, A Process Engineering Method Based on Ontology and Patterns, ICSOFT 2008, 3rd International Conference on Software and Data Technologies, Porto, Portugal.

Hug et al., 2008c

C. Hug, A. Front, D. Rieu, A Process Engineering Method based on a Process Domain Model and Patterns, International workshop MoDISE-EUS 2008 (Model Driven Information Systems Engineering: Enterprise, User and System Models), held in conjunction with CAISE 2008, Montpellier, France.

Hug et al., 2009

C. Hug, A. Front, D. Rieu, Brian Henderson- Sellers, « A Method to build Information Systems Engineering Process Metamodels », Journal of Systems and Software. Mai 2009.

Hug et al., 2010a

C. Hug, N. Mandran, **A. Front**, D. Rieu, "Qualitative Evaluation of a Method for Information Systems Engineering Processes", 4th International Conference on Research Challenges in Information Science (RCIS), Nice, Mai 2010.

Hug et al., 2010b

C. Hug, **A. Front**, D. Rieu, "A Method and a Tool based on a Conceptual Graph for Information Systems Engineering Processes", 5th International Conference on Evaluation on Novel Approaches to Software Engineering (ENASE), Athènes, Grèce, Juillet 2010.

Humphrey et Kellner, 1989

W.S. Humphrey, M.I. Kellner, *Software Process Modeling: Principles of Entity Process Models*, ICSE '89: *Proceedings of the 11th International Conference on Software Engineering*, ACM Press, Pittsburgh, PA, USA, May 15-18 1989.

-I-**ISO, 2007**

ISO/IEC 24744:2007, Software Engineering - Metamodel for Development Methodologies, 2007.

-J-**Jarke et al., 1992**

M. Jarke, J. Mylopoulos, J.W. Schmidt, Y. Vassiliou, « DAIDA: An Environment for Evolving Information Systems », *ACM Transactions on Information Systems*, January 1992, vol. 10, n° 1, p. 1-50.

Johnson, 1992

R.E. Johnson, *Documenting Frameworks using Patterns*, OOPSLA'92, 1992.

Juras et al., 2006

D. Juras, D. Rieu, S. Dupuy-Chessa, **A. Front**, "Conception collaborative pour les Systèmes Mixtes", Congrès INFORSID'06, Hammamet, 1-3 Juin 2006.

-K-**Kang et al., 1990**

K. Kang, S. Cohen, J. Hess, W. Novak, S. Peterson, *Feature-Oriented Domain Analysis (FODA) feasibility study*, Technical report CMU/SEI-90-TR-21, Software Engineering Institute, Carnegie Mellon University, November 1990.

Kanter, 1994

R. M. Kanter , « Collaborative Advantage : The art of alliances », *Harvard Business Review*, vol. , p. 96-108, July-August, 1994.

Kapitsaki et al., 2009

G. Kapitsaki, D. Kateros, G. Prezerakos, I. Venieris. Model-driven development of composite context-aware web applications. *Information and Software technology*, vol. 51, pp. 1244-1260. Elsevier, 2009.

Khayati, 2005

O. Khayati, « Modèles formels et outils génériques pour la gestion et la recherche de composants », thèse de doctorat de l'Institut National Polytechnique de Grenoble, soutenue le 17 décembre 2005.

Khayati et al., 2003a

O. Khayati, **A. Front-Conte**, J.P. Giraudin, "Intégration de la notion de rôle dans un modèle de composant métier", Conférence CopSTIC'03, Rabat, Maroc, 11-13 décembre 2003.

Khayati et al., 2003b

O. Khayati, **A. Front-Conte**, J.P. Giraudin, "Vers un modèle de recherche de composants", Colloque ALCAA2003, 4-5 Septembre, 2003, Bayonne, France, pp 130-141.

Khayati et al., 2004a

O. Khayati, **A. Front**, J.P. Giraudin, "Intégration de la notion de rôle dans un modèle de composants métier", Revue AMADEIA, Maroc. pp 75-86, 2004.

Khayati et al., 2004b

O. Khayati, **A. Front**, J.P. Giraudin, "A metamodel for a metatool to describe and manage components", Conference IEEE-ICTTA, Dams, Syrie April 19-23, 2004, pp 403-405.

Khayati et al., 2004c

O. Khayati, **A. Front**, J.P. Giraudin, "Un métamodèle pour Systèmes d'Accès à des Bases de Composants Hétérogènes", Conference CARI'04, Hammamet, Tunisie, 22-25 novembre 2004.

Khayati et al., 2005a

O. Khayati, **A. Front** et J-P. Giraudin, "Une approche de réingénierie des systèmes d'information basée sur la génération et l'appariement de spécifications de composants", Revue électronique internationale e-TI (e-revue en Technologies de l'Information), 1er numéro, 2005, <http://www.revue-eti.net>.

Khayati et al., 2005b

O. Khayati, **A. Front**, J.P. Giraudin, « Génération et appariement de spécifications formelles de diagrammes de classes pour la recherche de composants », Congrès INFORSID'05, Grenoble, France 24-27 Mai 2005.

KLC, 2004

Quel avenir pour la fonction SI ?, Les Synthèses de KLC, mai 2004.

Klöckner et al., 2010

K. Klöckner, K. Kohler, D. Kerkow, S. Niebuhr, C. Nass, « Aligning Business Goals and User Goals by Engineering Hedonic Quality », The ACM SIGCHI Symposium on Engineering Interactive Computing Systems (EICS'10), Berlin, Allemagne, 19-23 Juin 2010.

Kruchten, 2000

P. Kruchten, *The Rational Unified Process: An Introduction*, Addison-Wesley.

Kumar et Welke, 1992

K. Kumar et R.J. Welke. Methodology EngineeringR: a proposal for situation-specific methodology construction. W. W. Cotterman et J. A. Senn (Eds.), *Challenges and strategies for research in systems development*, John Wiley & Sons, pp. 257-269, 1992.

Kunz et Rittel, 1970

W. Kunz, H.W.J. Rittel, *Issues as elements of information systems*, Working Paper n° 131, Heidelberg-Berkeley, 1970.

-L-

Lahna et al. 2003a

B. Lahna, O. Roudies, J-P. Giraudin, **A. Front-Conte**, « Patterns pour l'intégration des vues dans l'ingénierie des systèmes d'information », Conférence CopSTIC'03, Rabat, Maroc, 11-13 Décembre 2003.

Lahna et al., 2003b

B. Lahna, O. Roudies, J-P. Giraudin, **A. Front-Conte**, "Classification des vues à base de treillis", Conférence CopSTIC'03, Rabat, Maroc, 11-13 Décembre 2003.

Lahna et al., 2004

B. Lahna, O. Roudies, J-P. Giraudin, **A. Front Conte**, "A lattice based approach to views integration in information systems", Eighth Maghrebian Conference on Software Engineering and Artificial Intelligence, Tunisie, Mai 2004.

Lamsweerde, 2001

A. Lamsweerde, *Goal-Oriented Requirements Engineering: A Guided Tour, Invited Paper for RE'01 - 5th IEEE International Symposium on Requirements Engineering*, Toronto, August, 2001, pp. 249-263

Lamsweerde et al., 1998

A. Lamsweerde, R. Darimont, E. Letier, « Managing conflicts in goal-driven requirements engineering », *IEEE transactions on Software engineering*, Vol. 24 No. 11, p. 908-926, 1998.

-M-

Maiden, 1998

N. Maiden, « Crews-Savre : Scenarios for acquiring and Validating requirements », *Journal for Automate Software Engineering*, vol. 5, n° 4, p. 419-446, 1998.

Maamar et al., 2011

Z. Maamar, N. Faci, L. Krug Wives, H. Yahyaoui, H. Hacid, *Towards a Method for Engineering Social Web Services*, Conférence Method Engineering, Paris, 20-22 Avril 2011.

Marshal et al., 1999

P. Marshal, J. McKay, J. Burn, « The tree S's of Virtual Organisations : Structure, Strategy and Success Factors », *School of Management Information Systems*, Edith Cowan University, 1999.

Martin et Odell, 1996

J. Martin, J.J. Odell, *Object Oriented Methods: Pragmatic Considerations*. Prentice-Hall – Englewood Cliffs, New Jersey, USA, 1996.

Mazzechi, 2001

M. Mazzechi, « The Virtual Organisation », *7 th International Conference on Concurrent Enterprising*, Bremen, Vol. 24 No. 11, p. 331-335, 27-29 June, 2001.

Mc Dermid et Ripken, 1984

J. McDermid et K. Ripken. *Life cycle support in the ADA environment*. University Press, 1984.

Mirbel and Ralyté, 2006

I. Mirbel, J. Ralyte, *Situational method engineering : combining assembly-based and roadmap-driven approaches*, Requirement Engineering Journal, vol. 11, n°1, 2006, pp. 58--78.

-N-**Nurcan et Rolland, 2005**

S. Nurcan, C. Rolland, *50 ans de Système d'Information : de l'automatisation des activités individuelles à l'amélioration des processus et la création de valeur ajoutée*, 50 ans de l'IAE de Paris, 2005.

-O-**OMG, 2002**

MetaObjectFacility (MOF) Specification, Version 1.4, April 2002.

OMG, 2005

Software Process Engineering Metamodel Specification, Version 1.1, January 2005.

OMG, 2006

Unified Modeling Language: Superstructure, Version 2.1, April 2006.

OOSPICE, 2002

OOSPICE: Software Process Improvement and Capability Determination for Object Oriented/ Component Based Software Development, <http://oospice.com>, 7 October 2002.

OPF, 2005

OPF: Open Process Framework, <http://www.opfro.org/>, 16 décembre 2005.

-P-**Pagel et Winter, 1996**

B. Pagel, M. Winter, *Towards Pattern-Based Tools*, EuroPLoP, 1996.

Perez-Medina et al., 2007

J-L. Perez-Medina, S. Dupuy-Chessa, **A. Front**, "A survey of Model Driven Engineering Tools for User Interface Design", 6th International Workshop on Task Models and Diagrams, TAMODIA 2007, actes publiés par LNCS, éditeurs Marco Winckler, Hilary Johnson, P. Palanque.

Pernici, 2010

B. Pernici, *Advanced Information Systems Engineering*, 22nd International Conference, CAiSE 2010, Hammamet, Tunisia, June 7-9, 2010.

Pernici, 2011

Barbara Pernici: Panel on Green and Sustainable IS. [CAiSE 2011](#), London, United Kingdom, 20-24 June, 2011.

Pohl et al., 2005

K. Pohl, G. Bockle, F.V.D Linden, *Software Product Line Engineering : Foundations, Principles and Techniques*, Springer-Verlag, 2005.

Potts et Bruns, 1988

C. Potts, G. Bruns, *Recording the Reasons for Design Decisions, ICSE '88: Proceedings of the 10th International Conference on Software Engineering*, IEEE Computer Society Press, Singapore, April 1988.

Priego-Roche, 2011

L-M. Priego-Roche, « Modélisation intentionnelle des organisations ouvertes », thèse de doctorat de l'Université de Grenoble, soutenue le 13 avril 2011.

Priego-Roche et al., 2009a

L-M. Priego, D. Rieu, **A Front.**, "A 360 Vision for Virtual Organizations Characterization and Modelling: Two Intentional Level Aspects", 9th IFIP Conference on e-Business, e-Services and e-Society, I3E 2009, Springer-Verlag (Lecture Notes in Computer Science), Nancy, 2009.

Priego-Roche et al., 2009b

L-M. Priego-Roche, D. Rieu, **A. Front.**, « Caractérisation et Modélisation Intentionnelles des Organisations Virtuelles », INFORSID'09, Toulouse, 26-29 mai 2009.

Priego-Roche et al., 2010

L-M. Priego-Roche, D. Rieu, **A. Front.**, "Vers une caractérisation intentionnelle des organisations virtuelles », *Revue Ingénierie des Systèmes d'Information (RSTI-ISI)*, Vol. 15, n°3, Hermès, 2010.

-R-**Ralyté et Rolland, 2001**

J. Ralyté, C. Rolland, *An Assembly Process Model for Method Engineering, Proceedings of the 13th Conference on Advanced Information Systems Engineering CAISE'2001*, Springer-Verlag, LNCS 2068, 2001.

Ralyté et al., 2007

J. Ralyté, S. Brinkkemper et B. Henderson-Sellers (Eds.). *Situational Method Engineering: Fundamentals and Experiences*, Proceedings of the IFIP WG 8.1 Working Conference, *Situational Method Engineering, Springer*, 2007.

Reix, 1998

R. Reix, *Systèmes d'information et management des organisations*, Vuibert, 1998.

Rieu et al., 1999

D. Rieu, J.P. Giraudin, C. Saint-Marcel, **A. Front-Conte**, *Des opérations et des relations pour les patrons de conception*, Congrès INFORSID'99, Toulon, Juin 1999.

Rieu et al., 2002

D. Rieu, **A. Front-Conte**, J.P. Giraudin, *Pattern-Based Environments for Information Systems Development "*, International Conference, The Sciences of Design, Lyon, 15-16 mars 2002.

Robinson et al., 2005

P. Robinson, Y. Karabulut, J. Haller, « Dynamic Virtual Organization Management Service Oriented Enterprise Applications », IEEE International Conference on Collaborative Computing : Networking, Applications and Worksharing, 2005.

Rocques et Vallée, 2000

P. Rocques, F. Vallée, *UML en action : de l'analyse des besoins à la conception en Java*, Eyrolles, 2000.

Rolland, 2005

C. Rolland, *L'ingénierie des méthodes : une visite guidée, revue électronique internationale e-TI*, Premier numéro, 2005.

Rolland et al., 1995

C. Rolland, C. Souveyet, M. Moreno, *An Approach for defining ways-of-working*, *Information System Journal*, vol. 20, n° 4, 1995.

Rolland et al., 1999

C. Rolland, N. Prakash, A. Benjamen, *A Multi-Model View of Process Modelling*, *Requirements Engineering*, Springer-Verlag London Limited, vol. 4, n° 4, 1999.

Rolland et al., 2000

C. Rolland, S. Nurcan et G. Grosz. A decision-making pattern for guiding the enterprise knowledge development process. *Information & Software Technology*, 42 (5), pp. 313-331, 2000.

Royce, 1970

W. W. Royce. Managing the Development of Large Software Systems. *IEEE Wescon*, pp 1-9, 1970.

-S-**Saad et al., 2003**

I. Saad, F. Basséras, C. Rosenthal-Sabroux, Les démarches de gestion des connaissances Vers une meilleure conception et exploitation des systèmes d'information coopératifs dans l'entreprise étendue. *Revue Ingénierie des systèmes d'information*, vol. 8, n°2, pp. 33-56, 2003.

Saidi, 2009

R. Saidi, « Conception et usage des composants métier processus pour les systèmes d'information »; thèse en co-tutelle entre l'INPG et l'Université Mohammed V-Agdal de Rabat, soutenue le 26 septembre 2009.

Saidi et al., 2007

R. Saidi, M. Fredj, S. Mouline, A. Front, D. Rieu, « Towards Managing Variability Across Business Component Development », *IEEE International Conference on Information Reuse and Integration*, IRI. Las Vegas, USA, pp. 594-599, 2007.

Saidi et al., 2008a

R. Saidi, M. Fredj, S. Mouline, A. Front, D. Rieu, Variability Modelling for Business Component Customization, *The IEEE Symposium on Computers and Communications (ISCC'08)*, Marrakech, Maroc.

Saidi et al., 2008b

R. Saidi, A. Front, D. Rieu, M. Fredj, S. Mouline, From a Business Component to a Functional Component using a Multi-View Variability Modelling, *MoDISE-EUS'2008 workshop in conjunction with The 20th International Conference on Advanced Information Systems Engineering (CaiSE'08)*, Montpellier, France.

Saidi et al., 2008c

R. Saidi, M. Fredj, S. Mouline, **A. Front**, D. Rieu, Spécification de composants métier : une approche par expression de variabilité multi-vue, Congrès INFORSID'08, Fontainebleau, 27-30 mai 2008.

Saidi et al., 2009a

R. Saidi, M. Fredj, **A. Front**, S. Mouline. « Variabilité dans les composants métiers multivues », Revue RSTI, série ISI, Vol. 14, numéro 2, 2009.

Saidi et al., 2009b

R. Saidi, **A. Front**, D. Rieu, M. Fredj, S. Mouline, Component-Based Development : Extension with Business Component Reuse, 3rd IEEE International Conference on Research Challenges in Information Science (RCIS), Avril 2009.

Saidi et al., 2011

R. Saidi, **A. Front**, D. Rieu. « Réutilisation orientée métier », Revue Ingénierie des Systèmes d'Information (RSTI-ISI), Vol. 16, n°2, Hermès, 2011.

Schwaber et Beedle, 2001

K. Schwaber et M. Beedle. *Agile Software Development with SCRUM*. Prentice Hall, Upper Saddle River, New Jersey, 2001..

Santorum et al., 2010

M. Santorum, **A. Front**, D. Rieu, N. Mandran, “Approche de gestion des processus universitaires basée sur les jeux”, INFORSID'10, Marseille, 25-28 mai 2010.

Santorum et al., 2011

Approche de gestion des processus basée sur les jeux - Auteurs : Marco Santorum, Agnès Front, Dominique Rieu, Nadine Mandran. Revue RTSI, série ISI, Vol. 16, N°1, XVI(I):33--59, 2011.

Schmidt, 1997

D.C. Schmidt, *Applying design patterns and frameworks to develop object-oriented communication software*, in Salus P. Ed., Handbook of Programming Languages, MacMillan Computer Publisher, 1997.

Serral et al., 2010

E. Serral, P. Valderas, V. Pelechano, V. Towards the model driven development of context-aware pervasive systems. Pervasive and Mobile Computing, vol. 6, pp. 254-280, 2010.

-T-**TRAFIC, 2005**

Collectif TRAFIC, « Intermodalité, réutilisation et coopération dans le domaine des transports – Le projet TRAFIC », Atelier Systèmes d'Information Transport, Congrès INFORSID'05, Grenoble, France, 24 Mai 2005.

Travica, 1997

B. Travica, « The Design of the Virtual Organisation. : a research model », Associations of Information Systems Conference, Indianapolis, vol. , p. 15-17, August, 1997.

Tripathy et al., 2007

A. Tripathy, S.D. Eppinger, « A System Architecture Approach to Global Product Development », , MIT Sloan Research Paper, March, 2007.

Turner et Bedell, 2003

J. Turner et K. Bedell, "STRUTS", édition CampusPress, juin 2003.

-V-

Van der Maßen et Lichter, 2002

T. Van der Maßen, H. Lichter, *Modeling variability by UML use case diagrams, International Workshop on Requirements Engineering for Product Lines (REPL'02)*, Septembre 2002.

Van Grup, 2000

J. Van Grup, *Variability in Software Systems, the key to software reuse*, Licentiate Thesis, University of Groningem, Sweden, 2000.

Vieira et al., 2010

V. Vieira, P. Tedesco, A. Salgado. Designing context sensitive systems : An integrated approach. In Expert Systems with Applications. Elsevier. 2010.

-W-

Wartik et Prieto-Diaz, 1992

S. Wartik, R. Prieto-Diaz, Criteria for Comparing Reuse-Oriented Domain Analysis Approaches, *International Journal of Software Engineering and Knowledge Engineering*, vol. 2, n° 3, pp. 403-431, 1992.

-Y-

Yang et al., 2006

Z. Yang, J.B. Zhang, C.P. Low, « Towards Dynamic Integration on Collaborative Virtual Enterprise Using Semantic Web Services », *IEEE International Conference on Industrial Informatics*, 2006.

Yu, 1997

E. Yu, « Towards Modeling and Reasoning Support from early-phase requirements Engineering », *IEEE 3th International Symposium on Requirements Engineering*, Annapolis MD, vol. , p. 226-235, 5-8 Janvier, 1997.

Yu et Mylopoulos, 1993

E. Yu, J. Mylopoulos, *An Actor Dependency Model of Organizational Work With Application to Business Process Reengineering, Proceedings of Conference on Organizational Computing Systems*, November, 1993, Milpitas, Calif., USA, Simon Kaplan, ed., ACM Press, pp. 258-268.

-Z-

Zellou, 2008

A. Zellou, « Contribution à la réécriture LAV dans le contexte de WASSIT, un framework d'intégration de ressources », thèse de l'Ecole Mohamedia d'Ingénieurs de Rabat, Maroc, soutenue le 5 avril 2008.

Zellou et al., 2004

A. Zellou, D. Chiadmi, A. Front, JP Giraudin , « The e-government Platform Based on Reused Components & Mediation”, , *IEEE International Conference on Information &*

Communication Technologies: from theory to applications. ICTTA'04, Label. April 19 - 23, 2004, Damascus, Syria.

Zellou et al., 2006

A. Zellou, D. Chiadmi, **A. Front**, « Mediation Systems: A Survey », IEEE International conference on information & Communication Technologies : from theory to applications, April 24 - 28, 2006, Damascus, Syria.

Ziadi et Jézéquel, 2005

T. Ziadi, J.M. Jézéquel, *Manipulation de lignes de produits logiciels : une approche dirigée par les modèles, Ingénierie Dirigée par les Modèles (IDM'05)*, Mai 2005.

ANNEXE A : THÈSES ENCADRÉES

	<i>Titre de la thèse</i>	<i>Etablissement</i>	<i>Date</i>	<i>Co-encadrement</i>
I. Hassine	Spécification et formalisation des démarches de développement à base de composants métier : la démarche Symphony	INPG, Grenoble	24/09/05	D. Rieu
O. Khayati	Modèles formels et outils génériques pour la gestion et la recherche de composants	INPG, Grenoble	17/12/05	J-P. Giraudin
A. Zellou	Contribution à la réécriture LAV dans le contexte de WASSIT, un framework d'intégration de ressources	EMI, Rabat, Maroc	05/04/08	D. Chiadmi
N. Arnaud	Fiabiliser la réutilisation des patrons par une approche orientée complétude, variabilité et généricité des spécifications	UJF, Grenoble	13/10/08	D. Rieu
R. Saidi	Conception et usage des composants métier processus pour les systèmes d'information ; thèse en de Rabat, soutenue le, co-dirigée avec Dominique Rieu et Mounia Fredj (ENSIAS, Rabat, Maroc).	Co-tutelle INPG et l'Université Mohammed V-Agdal, Rabat, Maroc	26/09/09	D. Rieu M. Fredj
C. Hug	Méthode, modèles et outils pour la méta-modélisation des processus d'ingénierie des systèmes d'information	UJF, Grenoble	20/10/09	D. Rieu
L-M. Priego-Roche	Modélisation intentionnelle des organisations ouvertes	Université de Grenoble	13/04/11	D. Rieu
A. Ben-Cheikh	Démarche d'ingénierie de systèmes d'information ubiquitaires »	Université de Grenoble	En cours	J-P. Giraudin, S. Coulondre
M. Santorum	Méthode de gestion des processus métier basée sur les jeux	Université de Grenoble	En cours	D. Rieu

ANNEXE B : DEA, M2R ET MÉMOIRES D'INGÉNIEURS CNAM ENCADRÉS

	<i>Type de stage</i>	<i>Titre</i>	<i>Année</i>	<i>Co-encadrement</i>
B. Barthez	Mémoire d'ingénieur CNAM	Vers un environnement d'aide à la conception centré patron.	2001	D. Rieu
O. Khayati	DEA MATIS	Vers des systèmes de patrons formalisés et outillés.	2001	D. Rieu
C. Descombes	Mémoire d'ingénieur CNAM	AGAP : un générateur d'applications pour favoriser la réutilisation des patrons.	2003	D. Rieu
L. Tastet	Mémoire d'ingénieur CNAM	AGAP : un Atelier de Gestion et d'Application de Patrons.	2004	D. Rieu
N. Arnaud	DEA MATIS	Imitation et Intégration de patrons pour la conception de Systèmes d'Information.	2004	D. Rieu
O. Meyssonier	Mémoire d'ingénieur CNAM	Mise en œuvre d'un EAI pour le pilotage d'applications télécoms pour la livraison des services xDSL.	2004	Olivier Orand, France Télécom R&D
E. Jausseran	Mémoire d'ingénieur CNAM	Démarche Symphony étendue, formalisation et expérimentation sur un Système d'Information Hospitalier.	2005	D. Rieu
C. Hug	M2R SI	Vers un méta-modèle de processus dédié à l'ingénierie des systèmes d'information.	2006	D. Rieu
T.T. Le Thi	M2R SI	Intégration d'une méthode d'ingénierie des besoins dans une méthode de développement centrée composants.	2006	D. Rieu
R. Benhamada	Mémoire d'ingénieur CNAM	Conception de services innovants par utilisation des BPM.	2006	F. Herledan (France Télécom R&D).
T. Montanier	Mémoire d'ingénieur CNAM	Etude et mise en œuvre d'un métamodèle pour l'imitation de patrons.	2006	D. Rieu N. Arnaud
A. Ben-Cheikh	M2R Informatique	Une méthode de rétro-ingénierie des processus métier basée sur un	2008	D. Rieu

		méta-modèle multi-vues.		
S. Hachani	M2R Informatique	Prise en compte de la personnalisation dans les méthodes de conception de SI.	2009	S. Dupuy-Chessa
N. Pignier	Mémoire d'ingénieur CNAM	Développement d'une bibliothèque de patrons de transformation de modèles.	Non soutenu pour raisons personnelles	J-M/ Favre
Y. Tourasse	Mémoire d'ingénieur CNAM	Instrumentation d'une démarche d'ingénierie de systèmes d'information transports pervasifs, sécurisés et personnalisés.	2009	
M. Carron-Toro	M2R Informatique	Processus métier : de l'intuition au formel.	2011	D. Rieu

ANNEXE C : PARTICIPATION À DES PROJETS

<i>Titre</i>	<i>Type</i>	<i>Domaine</i>	<i>Période</i>	<i>Implication</i>
DéSIT	Région Rhône-Alpes, cluster TTS	Systèmes d'information Transport, Personnalisation, Adaptation	2008-2011	Coordinateur
COMPUS	PAI VOLUBILIS	Réutilisation	2006-2009	Co-encadrement de la thèse de Rajaa Saidi
COCIVI	Institut IMAG	Systèmes interactifs post-WIMP	2006-2008	Participation
TRAFIC	Région Rhône Alpes, thématique prioritaire "Aide à la décision : transports".	Systèmes d'information Transport, Multimodalité, Coopération	2003-2006	Responsable de lots et de livrables.
STIC-GL	Réseau franco-marocain	Réutilisation, composants	2002-2004	Co-encadrement de la thèse d'Ahmed Zellou
Initiative Centr'ACTOLL	Ministère de l'Industrie dans le domaine des logiciels de péage du futur..	Systèmes d'information transport, Réutilisation	2001-2003	Participation

Résumé

Depuis plusieurs années, l'avènement des nouvelles technologies de l'information et de la communication a considérablement modifié le rôle des systèmes d'information dans le fonctionnement des organisations. Les systèmes d'information doivent désormais s'adapter au profil de l'utilisateur, à son contexte d'utilisation et au dispositif utilisé. De plus, les systèmes d'information se doivent de supporter, voire de faciliter, les évolutions inéluctables et les changements de tous ordres des organisations. Nous présentons un ensemble de contributions visant à proposer des méthodes permettant de développer et de faire évoluer des systèmes d'information en fonction des besoins et des technologies. Deux axes principaux sont étudiés. D'une part, l'ingénierie de systèmes d'information par réutilisation vise à proposer une meilleure spécification des produits du développement en vue de leur réutilisation ainsi qu'une meilleure formalisation des processus de développement en vue de leur adaptation aux différents projets d'une organisation ou aux organisations elles-mêmes. D'autre part, du point de vue de l'ingénierie des méthodes, nous proposons des méthodes de développement spécifiques destinés à des types de systèmes d'information particuliers ou utilisant des approches innovantes et participatives telles que les serious games. Enfin, en adéquation avec la nouvelle génération de systèmes d'information appelée "Socialisation et Mobilité", nous proposons des perspectives de recherche autour de trois axes principaux : l'ingénierie de méthodes de conception de systèmes d'information sociaux, l'ingénierie de méthodes de conception sociales et l'ingénierie sociale de méthodes de conception.

Abstract

For several years, the advent of new information and communication technologies has changed the role of information systems in organizations' functioning. Information systems must now adapt to the user's profile, to his context of use and to the device he uses. In addition, information systems have to support or facilitate organizations' inevitable evolutions and changes. We present a set of contributions which aim to propose methods to develop and make information systems evolve according to the evolution of needs and technology. Two main areas are studied. First, engineering by reuse of information systems aims to provide a better specification of the development products for their later reuse and to better formalize the development processes for their adaptation to different projects of an organization or to organizations themselves. On the other hand, in terms of engineering methods, we propose some methods dedicated to specific types of information systems, and some using innovative and participative approaches such as serious games. Eventually, in line with the new generation of information systems called "Socialization and Mobility", we offer research opportunities in three main areas: engineering methods for designing social information systems, engineering of social design methods and social engineering of design methods.