



HAL
open science

Contribution à une méthodologie d'automatisation des processus par caractérisation des ressources

Kawtar Ougaabal

► **To cite this version:**

Kawtar Ougaabal. Contribution à une méthodologie d'automatisation des processus par caractérisation des ressources. Automatique / Robotique. Université de Bordeaux, 2020. Français. NNT : 2020BORD0303 . tel-03245715

HAL Id: tel-03245715

<https://theses.hal.science/tel-03245715v1>

Submitted on 2 Jun 2021

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

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

THÈSE PRÉSENTÉE
POUR OBTENIR LE GRADE DE
DOCTEUR DE
L'UNIVERSITÉ DE BORDEAUX

ECOLE DOCTORALE DES SCIENCES PHYSIQUES ET DE L'INGÉNIEUR
SPÉCIALITÉ PRODUCTIQUE

Par Mme Kawtar OUGAABAL

**Contribution à une méthodologie d'automatisation des
processus par caractérisation des ressources**

Sous la direction d'Yves DUCQ
Co-directeur : Grégory ZACHAREWICZ
Co-encadrant : Said TAZI

Soutenue le 14/12/2020

Membres du jury :

M. VALLESPER, Bruno	Professeur des Universités, Université de Bordeaux	Président
M. EL MHAMED, Abderrahman	Professeur des Universités, Université de Paris 8	Rapporteur
M. OUZROUT, Yacine	Professeur des Universités, Université Lumière Lyon 2	Rapporteur
M. MATHIEU, Jean-Michel	Directeur d'agence, EXAKIS NELITE	Examineur
M. DUCQ, Yves	Professeur des Universités, Université de Bordeaux	Directeur
M. ZACHAREWICZ, Grégory	Professeur de l'IMT, IMT - Mines d'Alès	Co-directeur
M. TAZI, Said	Maitre de conférences, UPPA	Co-encadrant

Titre : Contribution à une méthodologie d'automatisation des processus par caractérisation des ressources

Résumé :

Pour automatiser avec succès leur processus, les entreprises font appel aux SSII pour les aider à identifier les problèmes, ou limites, et proposer des solutions pour améliorer la productivité de leurs workflows. De ce fait, les SSII ont besoin d'une méthodologie et un outil proposant simultanément la modélisation et la simulation des processus. Afin d'atteindre l'objectif fixé par le client et faciliter la communication pendant l'étape d'analyse, les langages à utiliser doivent être compréhensibles par toutes les parties prenantes, de l'analyste métier au consultant IT. Pour cela, une méthodologie basée sur les principes de l'approche MDE est nécessaire pour séparer les besoins opérationnels des besoins techniques, tout en mettant en évidence les liens entre les deux. La méthodologie doit également utiliser un langage de modélisation qui permet de représenter graduellement les besoins opérationnels : modéliser le besoin à un haut niveau d'abstraction puis le détailler graduellement. Pour finir, le même langage de modélisation doit pouvoir présenter les exigences non-fonctionnelles avec la possibilité de les évaluer en utilisant la simulation. Dans ce but, nous proposons une méthodologie par caractérisation des ressources basée sur les principes d'analyse de l'approche MDSEA en utilisant l'outil eBPMN qui permet à la fois de modéliser et de simuler le processus à automatiser. Il s'agit d'une thèse CIFRE effectuée chez EXAKIS NELITE (SSII).

Mots clés : Automatisation des processus, Modélisation, Simulation, Ressource

Title: Contribution to a methodology for process automation by resource characterization

Abstract:

To successfully improve their business information systems and process automation, companies call on computer engineering and services societies (IT companies) to better help their clients to identify process automation barriers and other opportunities to overcome productivity gaps. IT companies require a tool merging process modeling and process simulation simultaneously. To reach the goal set by the client, the modeling simulation tool methodology should be understandable by all stakeholders, from the business analyst to the IT consultant, to increase communication and facilitate the analysis. For this, a tool based on an MDE approach is required. This tool can separate the operational needs from the technical needs, whilst still being able to highlight any links between the two. In addition, it couples the methodology with a modeling language allowing to model different levels of precision: capturing first high-level activities and then concentrating down to lower levels. Finally, the same modeling language includes the representation of non-functional requirements with simulation means to assess them. For that purpose, this thesis proposes an MDE analysis methodology with a tool merging modeling and simulation at the same phase, using a resource-perspective approach. Our work is based on Model Driven System Engineering Architecture (MDSEA) and eBPMN language.

Keywords: Process automation, Modeling, Simulation, Resources

Remerciement

Je vais enfin pouvoir répondre à la question « Alors cette thèse, ça avance ? » par « C'est finiiii ».

Tout d'abord, je remercie Dieu de m'avoir guidé vers cette aventure de doctorat CIFRE, une expérience à la fois challengeante et enrichissante au niveau professionnel et personnel. Merci de m'avoir donné la force de mener à terme ce travail grâce à l'aide et l'encouragement de plusieurs personnes.

Je vais commencer par remercier mon petit frère Abdelmoncef, je le remercie pour son aide, ses encouragements, sa confiance et je le remercie surtout pour les fous rires qu'il m'a apportés. Je remercie profondément et avec tout mon amour mes parents, pour leur amour, leur encouragement, pour les sacrifices consentis à mon égard, et leur éducation qui m'a permis entre autres de persévérer pour atteindre tout objectif fixé, en l'occurrence, la thèse. Que Dieu puisse vous accordez santé et longue vie.

Je tiens à exprimer toute ma gratitude et ma profonde reconnaissance à mes encadrants : M. Yves DUCQ, M. Gregory ZACHAREWICZ et M. Said TAZI. Je les remercie pour le temps qu'ils m'ont accordé tout le long de cette période de thèse et l'accompagnement apporté pour mener à bien les objectifs fixés. Je les remercie pour l'apprentissage scientifique et humain, grâce à eux j'ai appris que « le mieux est l'ennemi du bien » et que l'important est d'avancer. Sans eux ce travail ne serait pas ce qu'il est !

Je tiens à remercier profondément et chaleureusement M. Andrea D'AMBROGIO et son équipe de nous avoir fourni l'outil eBPMN et de nous avoir aidé à l'installer. Leur outil a été d'une aide considérable à l'avancement de la thèse.

Je tiens à remercier respectueusement M. Bruno VALLESPER d'avoir présidé mon jury de thèse et M. Abderrahman EL MHAMED et M. Yacine OUZROUT d'avoir accepté d'évaluer et de rapporter ce travail. Merci à eux pour leurs remarques et critiques constructives. Je tiens à remercier profondément M. Jean-Michel MATHIEU de m'avoir donné l'opportunité de faire une thèse CIFRE au sein de son agence et d'avoir accepté d'examiner ce travail de recherche.

Faire une thèse CIFRE dans une SSII n'est pas simple à premier abord. Surtout quand on a un caractère perfectionniste. De ce fait, je tiens à remercier profondément et chaleureusement ma coach Christine GALLO pour son accompagnement à débloquer toute impasse, à concilier mes différents domaines de vie et pour sa bienveillance sans faille !

Je tiens à remercier tous les membres de l'IMS, doctorants, chercheurs et personnels pour leur gentillesse et leur soutien en cas de besoin. Plus précisément Mme Valérie ABEL et Mme Fabienne PREVOT pour la gestion administrative. Je remercie également tous mes collègues d'Exakis Nelite. Cyril, pour son humour, son soutien et son aide à cadrer la problématique de la thèse surtout au début. Thao, pour son soutien, pour son adaptabilité et le fait d'avoir été compréhensive en tant que chef de projet. Rémi, pour son soutien et sa flexibilité à l'aménagement de mes horaires de travail. Asma, pour son soutien et son humour. Sans oublier Mélanie et tous mes collègues ayant contribué à la gestion de la partie administrative.

Mes remerciements vont également à mes amis pour leur aide et soutien. Zineb, pour son soutien et ses relectures au tout début de ma thèse. Simon, pour son soutien, son aide et pour les bons moments passés ensemble. Je le remercie profondément pour sa gestion de la partie

matérielle le jour de ma soutenance ce qui a permis de supprimer une partie considérable du stress. Sara, pour son aide, son soutien, ses relectures alors qu'elle avait ses propres occupations et sa propre thèse et surtout pour les fous rires qu'elle m'a apportés. Maroua, pour ses relectures alors qu'elle avait un poste très prenant, pour son soutien sans faille, (merci d'avoir toujours trouvé les bons mots) et sans oublier son humour sarcastique. Meriem, pour son soutien, pour ses relectures alors qu'elle avait un bébé à bas âge, pour ses bons conseils (sa piste de coach m'a fait gagner beaucoup de temps) et son humour. Khadija, pour son aide, son amitié, son soutien et ses relectures de dernières minutes. Je remercie également Touria, Leila, Abedlfatah, Ayoub et Amine pour leurs encouragements et leur humour.

Pour finir, je remercie les chercheurs scientifiques rencontrés lors des différentes conférences, pour leurs questions et leurs remarques pertinentes. Je remercie l'ANRT et la région aquitaine pour le financement de ma thèse. Je remercie également toutes les personnes que j'ai pu oublier qui ont contribué de près comme de loin à la réalisation de cette thèse.

Table des matières

INTRODUCTION	1
Chapitre I. CONTEXTE ET PROBLEMATIQUE.....	3
I. Structure organisationnelle d'une entreprise et système d'information	4
1. Structure organisationnelle d'une entreprise	4
2. Qu'est qu'un système d'information d'entreprise.....	6
II. Démarche d'automatisation des processus par une SSII	7
1. Demande client et recueil des besoins.....	8
2. L'analyse fonctionnelle et technique du besoin.....	9
3. Elaboration et intégration de la solution	10
4. Méthodologies de gestion de projet de développement d'un système par une SSII	11
III. Questionnement	13
Chapitre II. ETAT DE L'ART.....	15
I. Les processus métiers.....	16
1. Gestion des processus métiers.....	16
2. Workflow	17
3. Modélisation des processus métiers.....	19
4. Simulation des processus métiers.....	23
II. Extensions des langages semi-formels.....	27
1. Extension orientée domaine spécifique.....	29
2. Extension orientée activité, données et flux de contrôle	30
3. Extension orientée ressource.....	33
III. Ingénierie dirigée par les modèles.....	35
1. MDE.....	35
2. MDA.....	37
3. MDI.....	39
4. MDSEA.....	40
IV. Conclusion	42
1. Réponse aux questions de recherche	42
2. Synthèse	46
Chapitre III. CONTRIBUTION CONCEPTUELLE.....	47
I. Méthodologie proposée	48
1. Choix des langages de modélisation et simulation	48
2. Méthodologie.....	52
II. Distinction des ressources	54

1.	Grille des performances des ressources	55
2.	Ressources humaines	57
3.	Ressources matérielles.....	59
4.	Ressources IT	63
III.	Agrégation des performances des ressources	64
1.	Méthodes utilisées	64
2.	Application sur l'agrégation des performances des ressources.....	68
IV.	Interface graphique	72
V.	Analyse des propriétés non-fonctionnelles en se basant sur un chemin	75
VI.	Conclusion	78
Chapitre IV. CONTRIBUTION OPERATIONNELLE.....		79
I.	Implémentation des contributions dans eBPMN	80
1.	Outils d'implémentation d'eBPMN	80
2.	Implémentation de la distinction des ressources	81
3.	Implémentation de la ressource interface graphique.....	83
4.	Implémentation des options d'affichage des résultats de simulation	84
5.	Agrégation des performances des ressources	86
II.	Cas d'étude	88
III.	Conclusion	98
CONCLUSION ET PERSPECTIVES		99
DIFFUSIONS DES RESULTATS		103
REFERENCES		104

Liste des figures

Figure 1: Organisation pyramidale.....	4
Figure 2: Organisation transversale.....	5
Figure 3: Ensemble des opérations d'un SI	6
Figure 4: L'approche systémique selon (Le Moigne, 1973).....	7
Figure 5: Les étapes générales de développement d'une application.....	8
Figure 6: Les types de besoins d'une application.....	9
Figure 7: Cycle en cascade.....	12
Figure 8: Cycle en V	13
Figure 9: Définition d'un processus.....	16
Figure 10: Cycle BPM (Wil M. P. van der Aalst et al., 2003)	17
Figure 11: Dimensions des systèmes Workflow (Zacharewicz, 2006)	18
Figure 12: Méta modèle du Workflow selon (Hollingsworth, 1995)	19
Figure 13: Les composants d'un processus de simulation	24
Figure 14: Description graduelle du besoin dans la démarche d'IDM	36
Figure 15: Les niveaux de modélisation de MDA.....	38
Figure 16: Transformation dans l'approche MDA (Hardebolle, 2008).....	39
Figure 17: Modèle de référence de MDI	40
Figure 18: Méthodologie MDSEA	41
Figure 19: Diagramme de classe de la représentation des ressources dans BPMN (Domingos et al., 2016)	49
Figure 20: Représentation générale de l'approche d'automatisation des processus par caractérisation des ressources	52
Figure 21: Représentation de la méthodologie d'automatisation des processus par caractérisation des ressources	54
Figure 22: Types des ressources.....	55
Figure 23: Décomposition séquentielle	65
Figure 24: Décomposition alternative	65
Figure 25: Décomposition en parallèle.....	66
Figure 26: Réduction séquentielle	66
Figure 27: Réduction en OU	66
Figure 28: Réduction en ET	67
Figure 29: Etape de décomposition/agrégation des ressources	69
Figure 30: Décomposition d'une tâche en sous-tâches atomique	70
Figure 31: Décomposition en trois tâches atomiques.....	71
Figure 32: Composition de la tâche utilisateur.....	72
Figure 33: Exemple d'une interface graphique.....	74
Figure 34: Représentation conceptuelle de la ressource Interface Graphique.....	74
Figure 35: Choix des chemins de simulation en utilisant la probabilité.....	76
Figure 36: Choix des chemins de simulation en explicitant la décision de l'utilisateur	76
Figure 37: Les options d'affichage des résultats.....	77
Figure 38: Déroulement de la simulation de l'outil eBPMN	80
Figure 39: Ajout des classes de distinction des types de ressources dans PyBPMN	81
Figure 40: Exemple d'utilisation des syntaxes de distinction des ressources	82
Figure 41: Extension du méta-modèle PyBPMN avec la classe PyInterface	83
Figure 42: Exemple d'utilisation de PyInterface	84

Figure 43: Exemple de choix du chemin à simuler en indiquant la décision de l'utilisateur.....	85
Figure 44: Extension d'eBPMN avec la classe « Path ».....	86
Figure 45: Exemple d'utilisation de syntaxe d'agrégation de ressource	87
Figure 46:Extension d'eBPMN avec la classe «PerformanceAgregation ».....	87
Figure 47: Modélisation du processus à un haut niveau d'abstraction	90
Figure 48: Atteignabilité des chemins.....	91
Figure 49: Modélisation détaillée du processus	91
Figure 50: Tâches associées aux mesures non-fonctionnelles.....	92
Figure 51: Résultat global en utilisant l'agrégation des performances des ressources	93
Figure 52: Résultat global de la simulation des ressources juniors après la suppression des tâches inutiles	93
Figure 53: Résultat de simulation global pour les ressources humaines expérimentées	94
Figure 54: Résultat de chaque type de ressource pour le cas des ressources humaines juniores	94
Figure 55:Résultat de chaque type de ressource pour le cas des ressources humaines expérimentées .	95
Figure 56: Résultat de la simulation de la ressource composée interface graphique	97
Figure 57: Résultat de la simulation de la ressource composée interface graphique après les améliorations	97
Figure 58: Résultat global du chemin analysé après les améliorations des interfaces graphiques.....	97

Liste des tableaux

Tableau 1: Extension des langages semi-formels pour intégrer les caractéristiques de domaines spécifiques	30
Tableau 2: Extension des propriétés fonctionnelles des langages semi-formelles	31
Tableau 3: Extension des propriétés non- fonctionnelles des langages semi-formelles.....	33
Tableau 4: Extension des propriétés fonctionnelles et non-fonctionnelles des ressources des langages semi-formelles	35
Tableau 5: Comparaison des langages de modélisation des processus	50
Tableau 6: Description des tâches du langage BPMN	50
Tableau 7: Comparaison des outils de simulation industriels	51
Tableau 8: Grille de performance des ressources humaines.....	59
Tableau 9: Grille des performances des ressources matérielles	62
Tableau 10: Grille de performance des ressources IT	64
Tableau 11: Les règles d'agrégation de performance	71
Tableau 12: Grille de performance des évaluateurs	90
Tableau 13: Détermination des mesures non-fonctionnelles des tâches de l'évaluateur	92
Tableau 14: Description et amélioration d'une interface graphique	96

Table des sigles et abréviations

SSII: Société De Service et D'Ingénierie Informatique

MDE: Model Driven Engineering

MDA: Model Driven Architecture

MDI: Model Driven Interoperability

MDSEA: Model Driven Service Engineering Architecture (MDSEA)

BPMN: Business Process Model and Notation

EPC: Event-driven Process Chain

UML AD: Unified Modeling Language Activity Diagram

IDEF: Integration Definition

IT: Information Technology

BPM: Business Process Management

SI : Système D'Information

IDM: Ingénierie Dirigée par les Modèles

PyBPMN: Performability-enabled BPMN

INTRODUCTION

L'évolution de l'environnement économique actuel, caractérisée par des marchés en constante transformation, où la réactivité, la flexibilité et l'innovation sont désormais les nouveaux critères de succès, incite les entreprises et plus précisément les industries à intégrer de nouveaux concepts et pratiques visant à adapter leur façon de travailler à un marché où le client est le seul décideur. La devise « le client est roi » s'applique pleinement. Autrement dit, pour assurer sa pérennité, l'entreprise doit personnaliser ses produits et ses services en fonction de ses clients. Par conséquent, la mise en place d'une organisation qui part de l'écoute et besoins des clients et qui maîtrise les activités nécessaires à sa satisfaction, nécessite d'évoluer d'une organisation verticale caractérisée par une amélioration du service sans se préoccuper des autres services à une organisation transversale « organisation par processus », qui permet une vision et une amélioration globale du système.

La clarification des interactions entre les différents intervenants est l'un des avantages les plus significatifs de l'approche processus qui facilite la communication au sein de l'entreprise. De nos jours, la fluidité de circulation de l'information est devenue primordiale d'où l'émergence des technologies de l'information et de la communication qui œuvrent à optimiser le flux d'information en automatisant les processus métier de l'entreprise.

L'automatisation des processus métiers de l'entreprise vise à augmenter les gains de productivité procurés par la gestion par processus.

En outre, avec l'émergence des nouvelles technologies (Big Data, Cloud...), les entreprises évoluent dans des environnements IT de plus en plus complexes ce qui requiert, du fait de la multiplicité des systèmes utilisés, d'assurer leur interconnexion, d'où l'importance d'adopter et de formaliser les bonnes pratiques pour automatiser les processus et assurer leur interopérabilité tant en termes informatiques qu'en termes de pratiques. Ainsi, le numérique n'est plus une option à se permettre mais une nécessité qui touche toutes les sociétés et impactent leur compétitivité et leur survie. L'entreprise doit désormais faire face à un défi majeur, elle doit organiser le décloisonnement des métiers, favoriser la collaboration, faciliter l'accès à l'information et transformer la relation avec ses clients.

Afin de tirer profit des avantages de la transformation numérique, les entreprises ont recours à l'automatisation des processus, puisqu'elle est l'approche la plus pertinente pour maîtriser les activités dans le but d'augmenter la performance, réduire les coûts d'exploitation et satisfaire les clients. L'automatisation des processus de l'entreprise est appelée « Workflow » et est considéré comme l'une des briques majeures d'un système d'information. L'un des principaux défis à relever par ce type de projet est l'alignement et la compatibilité des besoins réels de l'entreprise et la solution développée.

Dans le but d'améliorer avec succès leur système d'information et de réussir leur projet d'automatisation des processus, les entreprises font appel aux sociétés de service en ingénierie informatique (SSII). Afin d'aider leurs clients à automatiser leur processus en détectant les anomalies qui y sont présentes et en identifiant les opportunités pour améliorer leur productivité, les SSII ont besoin d'une approche et d'un outil qui permettent de représenter les différents types d'exigences requises. C'est la problématique liée à notre travail de thèse CIFRE effectué chez EXAKIS NELITE.

Notre mémoire se décompose en quatre parties.

Dans le premier chapitre, nous allons présenter le contexte de notre étude. Nous allons commencer par définir les deux principales structures organisationnelles d'une entreprise et leur impact sur les changements à mettre en place. Nous allons par la suite définir le système d'information et son importance au sein d'une entreprise. Ensuite, nous allons détailler les différentes démarches utilisées par les SSII pour automatiser les processus métier, du recueil du besoin jusqu'à l'élaboration et l'intégration des solutions. Nous allons également présenter les méthodologies de gestion de projet utilisées pour gérer ces étapes. Pour finir, à travers ces notions, nous allons établir la problématique de cette thèse composée d'un ensemble de questionnements de recherche qui vont orienter l'état de l'art du deuxième chapitre.

Au cours du deuxième chapitre, nous présenterons le travail bibliographique que nous avons effectué. Dans un premier temps, nous allons présenter les concepts de base du domaine des processus métiers, et les différentes techniques et langages de modélisation et de simulation. Dans un deuxième temps, nous détaillons les travaux les plus représentatifs de la littérature traitant le problème d'extension des langages de modélisation des processus métiers. Dans un troisième temps nous allons présenter les différentes approches permettant de lier les différents niveaux de modélisation. Finalement, ce chapitre essaiera de répondre aux questions établies dans le premier chapitre faisant émerger une problématique de recherche qui fera l'objet de notre contribution.

Le troisième chapitre a pour objectif de présenter la contribution conceptuelle de la méthodologie d'automatisation des processus par une caractérisation des ressources que nous proposons. Dans un premier temps nous allons présenter nos choix de langage de modélisation et de simulation utilisés, puis, nous présenterons notre contribution à l'approche MDSEA (Model Driven System Engineering Architecture). Dans un deuxième temps, nous allons présenter les enrichissements apportés au langage de simulation choisi pour l'adapter à notre méthodologie.

Le quatrième chapitre explique la mise en œuvre technique de nos propositions sur un cas d'étude d'un processus d'un des clients d'Exakis Nelite qui illustrera l'application de notre méthodologie par la caractérisation des ressources que nous proposons.

Enfin, nous concluons ce mémoire et identifions les perspectives de recherches envisageables.

Chapitre I. CONTEXTE ET PROBLEMATIQUE

I. Structure organisationnelle d'une entreprise et système d'information	4
1. Structure organisationnelle d'une entreprise	4
2. Qu'est qu'un système d'information d'entreprise.....	6
II. Démarche d'automatisation des processus par une SSII	7
1. Demande client et recueil des besoins.....	8
2. L'analyse fonctionnelle et technique du besoin.....	9
3. Elaboration et intégration de la solution	10
4. Méthodologies de gestion de projet de développement d'un système par une SSII	11
III. Questionnement	13

Dans ce premier chapitre, nous allons présenter le contexte général de la thèse relatif à l'automatisation des processus métiers. Nous allons commencer par définir les différentes structures d'une entreprise et leur rôle dans la mise en place des outils technologiques. Nous allons par la suite donner la définition d'un système d'information et les démarches adoptées par les SSII pour son automatisation. Ce chapitre nous conduira à poser un certain nombre de questions de recherche qui ont orienté la conduite de notre état de l'art.

I. Structure organisationnelle d'une entreprise et système d'information

1. Structure organisationnelle d'une entreprise

L'entreprise se définit par ce qui la constitue et par sa structure. Une entreprise quelle qu'elle soit doit structurer les différentes parties qui la composent. La structure organisationnelle représente la configuration adoptée par l'entreprise pour organiser son travail. Cela consiste à identifier ses fonctionnalités ainsi que leur mode de fonctionnement, leurs rôles, leurs interactions et la façon dont elles sont réparties, regroupées et coordonnées. Ainsi, le choix d'une structure organisationnelle a un impact considérable sur les aspects relationnels et la répartition des responsabilités dans un organisme. Pour (Hellriegel and Slocum, 2006), trois besoins doivent être satisfaits par une structure d'organisation :

- Définir un mécanisme de communication de sorte à faciliter la circulation de l'information et la prise de décision
- Définir clairement les rôles d'autorité et de responsabilité
- Préciser les interactions et les coordinations souhaitées entre les fonctionnalités

Différents types de structure d'entreprise existent dans la littérature. Nous allons définir les deux principales distinctions existantes : la structure verticale et la structure horizontale.

1.1. Structure verticale

La structure verticale ou l'organisation pyramidale consiste à diviser l'activité de l'entreprise par service avec la direction générale au sommet suivie par les multitudes de fonctions possibles, appelées services (Figure 1): service de production, service de finance, etc. (Rodet, 2009). Pour chaque service on retrouve le même schéma de structure, une direction qui gère les collaborateurs des sous-services. Ainsi, il s'agit d'une structure qui repose sur la hiérarchisation des différents services et leurs responsables dans le but de donner une représentation plus claire et opérationnelle.

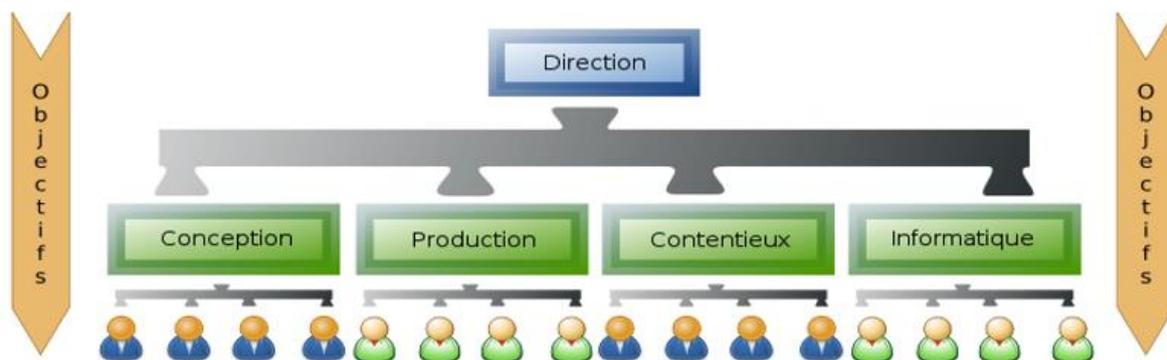


Figure 1: Organisation pyramidale

Parmi les avantages de la structure verticale :

- Le fonctionnement est simple, chacun se situe précisément par rapport à l'autre
- Elle permet d'identifier facilement le champ d'intervention
- Elle facilite la gestion des départements

Parmi les inconvénients de cette approche :

- Une organisation rigide
- Une centralisation excessive : manque de vision globale sur les projets à traiter
- Une lenteur de transmission de l'information

1.2. Structure horizontale

La structure horizontale vise à décloisonner les silos de travail découlant de l'organisation traditionnelle et privilégie une gestion horizontale comme le décrit la Figure 2. Son objectif est de lier et faire communiquer efficacement et harmonieusement les différentes parties d'une entreprise entre elles (Rodet, 2009). Les débuts de la structure horizontale proviennent de la promotion à l'autonomie des collaborateurs d'une entreprise. De manière plus récente, la structure horizontale est forgée par le domaine des Technologies de l'Information et de la Communication (TIC). L'élaboration des projets d'intégration des TIC amène les entreprises à penser leur organisation de travail de façon à favoriser la communication et les échanges entre ses différents acteurs et parties. Parmi les principaux avantages de cette structure :

- Limiter les interfaces et ainsi améliorer la communication et la productivité interne
- Améliorer la réactivité du fait de la vision globale procurée

Parmi les principaux inconvénients de la structure transversale :

- La difficulté à découper pertinemment les activités de l'entreprise de manière horizontale
- La Division des pouvoirs décisionnels peut mener à des problèmes d'encadrement

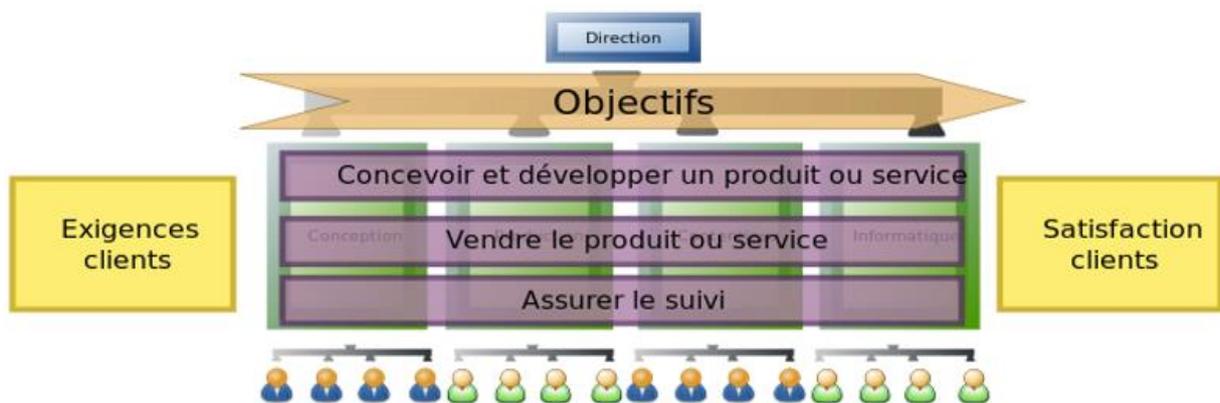


Figure 2: Organisation transversale

1.3. Changement dans une structure verticale

(Rodet, 2009) a décrit dans son article le déroulement du changement dans une structure pyramidale. On peut transposer cette description sur la mise en place de nouveaux outils technologiques au sein d'une entreprise. Le changement dans une structure pyramidale est généralement initié par la hiérarchie. La direction générale ou la direction de service vont définir à partir des objectifs qu'ils souhaitent atteindre et en fonction des représentations qu'ils se font sur le fonctionnement du service, les spécificités et les caractéristiques des outils technologiques à mettre en place. Généralement, ces décisions se prennent entre les responsables hiérarchiques impliquées exclusivement. Les utilisateurs de ces technologies ont uniquement une vision partielle et morcelée de la technologie à mettre en place. La vision présentée est une représentation subjective de la hiérarchie, ce qui peut générer la mise en place d'outils incompatibles avec leurs besoins et leur travail effectif.

1.4. Changement dans une structure horizontale

Dans une structure transversale, l'accent est mis sur la gestion par processus. Contrairement à la structure verticale qui raisonne en fonction des services, des postes et des tâches. Pour passer vers une approche dirigée client, l'approche processus est la meilleure solution à adopter. Elle permet aux entreprises de raisonner de façon transversale en se focalisant sur les résultats et la création de valeur ajoutée pour faire face à un marché évolutif et répondre à des clients de plus en plus exigeants. Elle représente également la référence de fonctionnement adoptée dans l'ISO 9000 pour faciliter l'approche client et elle est considérée comme l'un des principes forts de la démarche qualité. L'approche processus consiste à identifier les processus de l'organisme et de clarifier leurs interactions. Le concept est de considérer l'entreprise comme un processus en soi ou plus exactement une série de processus cohérents et liés, permettant de réaliser un produit ou un service visant des clients et des parties intéressés. De ce fait, dans le cadre de la mise en place de nouveaux outils technologiques, l'avantage principal de cette structure est le décloisonnement des services qui réunit les différents métiers de l'entreprise à réaliser une finalité partagée. Ainsi, toutes les parties prenantes ont une vue d'ensemble des activités réalisées ce qui facilite la prise de décision et l'amélioration continue des processus. Cela permet de faciliter l'implication des différents acteurs concernés par le changement, en identifiant les besoins des outils à mettre en place avec eux.

De ce fait, l'approche processus est l'une des solutions qui permet aux entreprises de raisonner de façon transversale en se focalisant sur la valeur ajoutée et les besoins clients (Barouch, 2013). Par conséquent, les entreprises se dirigent de plus en plus vers l'automatisation de leur système d'information en adoptant l'approche processus. La mise en place des nouvelles technologies en raisonnant par processus facilite la flexibilité, la capacité à créer des coopérations transversales entre ses différents métiers en interne, le développement des partenariats et d'échanges avec ses clients et ses fournisseurs. Dans le paragraphe qui suit nous allons présenter la définition d'un système d'information.

2. Qu'est qu'un système d'information d'entreprise

L'évolution d'un système d'information est un enjeu de taille pour les entreprises. Il est considéré comme un moyen pertinent pour la réduction des coûts, l'amélioration de la qualité et l'accroissement de la réactivité face à un contexte économique en constante évolution. Dans la littérature, plusieurs définitions sont données au système d'information (SI). D'après (O'Brien and Marion, 1997), un SI est un ensemble de ressources qui collecte, traite et distribue l'information au sein d'une entreprise. De ce fait, un système d'information utilise plusieurs ressources : humaines (utilisateurs), matérielles (machines) et logicielles (applications informatiques). La définition la plus citée et celle de Robert Reix « Un système d'information est un ensemble organisé de ressources : matériel, logiciel, personnel, données, procédures, etc. permettant d'acquérir, de traiter, de stocker des informations (sous forme de données, textes, images, sons, etc.) dans et entre des organisations » (Figure 3).

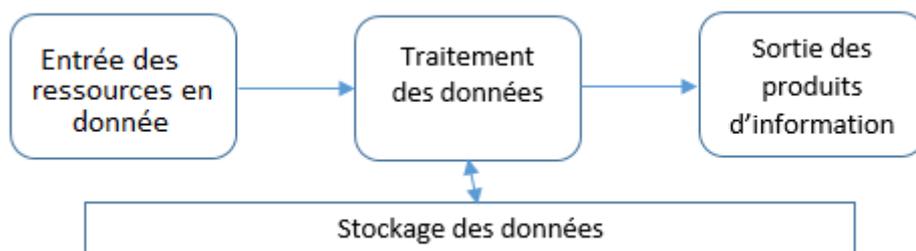


Figure 3: Ensemble des opérations d'un SI

L'approche de (Le Moigne, 1973) fait partie des travaux les plus cités dans le domaine de la systémique de l'organisation où l'entreprise est considérée comme un système composé d'éléments en interaction permanente. Son objectif est de mettre en évidence le rôle de fondement du système d'information au sein d'une entreprise (Figure 4)

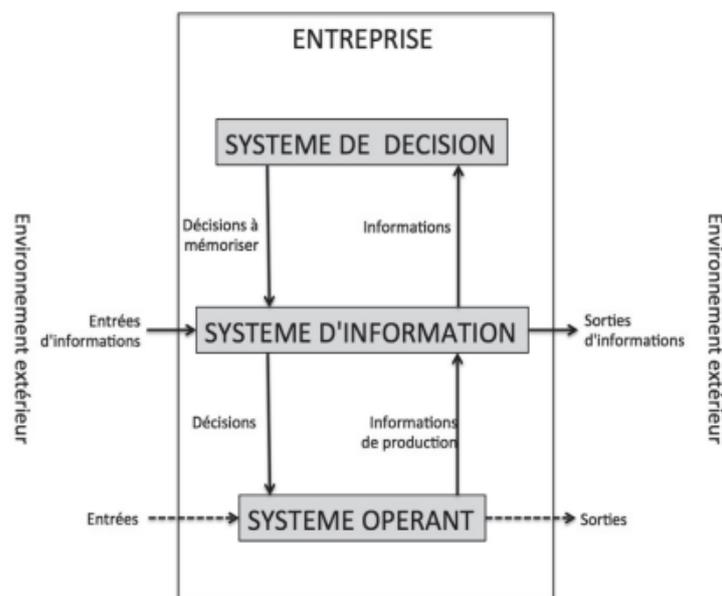


Figure 4: L'approche systémique selon (Le Moigne, 1973)

- **Système décisionnel** : il est également nommé système de pilotage, c'est lui qui prend les décisions, fixe les objectifs et les moyens de les atteindre. Il analyse le fonctionnement interne de l'entreprise et assure sa régulation. Il communique également avec les autres systèmes via les flux d'information
- **Système opérant** : il est également nommé système de production et est l'élément qui réalise toutes les opérations de production physique des biens et des services. Il communique avec les autres systèmes via les flux physiques et informationnels
- **Système d'information** : il joue le rôle d'interface entre le système opérant et le système décisionnel. Pour ce faire, il recueille les informations, les mémorise, les traite et les communique aux deux systèmes

De ce fait, le système d'information joue le rôle du pivot dans l'organisation, il assure la coordination et la cohésion de toutes ses entités. Son informatisation se traduit par l'automatisation du traitement d'information au niveau de la collecte, la communication et le stockage. L'automatisation des systèmes d'information couvre plusieurs domaines : le développement d'intranet, le développement de site web transactionnel, la mise en place des ERP, etc. Dans notre travail de recherche, nous étudions la partie d'automatisation des processus métiers. Puisqu'il s'agit d'une thèse Cifre qui s'est déroulée au sein d'une SSII, nous allons présenter dans le chapitre suivant les étapes d'automatisation des processus par une SSII.

II. Démarche d'automatisation des processus par une SSII

L'automatisation des processus se fait souvent dans le cadre d'un contexte contractuel entre un client et un fournisseur externe où la maîtrise d'ouvrage doit définir clairement son besoin et ses objectifs pour qu'ensuite la maîtrise d'œuvre puisse formaliser les caractéristiques du besoin et de sa solution. Dans ce chapitre nous allons décrire les principales étapes adoptées pour automatiser un processus par une

SSII, du recueil du besoin jusqu'à l'implémentation et la maintenance de la solution retenue (Figure 5). La description est basée sur le déroulement des étapes existantes chez Exakis Nelite et sur les références suivantes (Pétin, 2007; Bourlier, 2015; Amokrane, 2016; Compliance Consulting, 2017; Yende, 2019). Nous allons également présenter brièvement les approches adoptées pour gérer ces étapes.

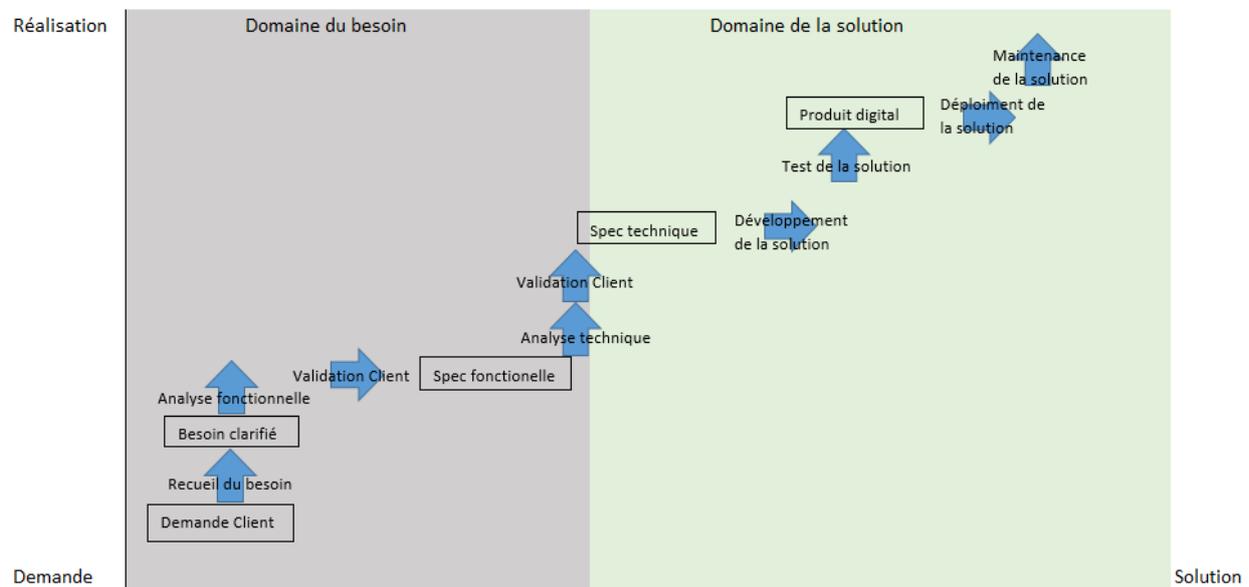


Figure 5: Les étapes générales de développement d'une application

1. Demande client et recueil des besoins

La demande des clients ou le besoin client représente l'expression initiale qui permet de développer le système demandé. Les besoins sont souvent définis de façon informelle dans le but de simplifier la communication avec les parties prenantes.

Le recueil du besoin est une étape qui consiste dans un premier temps à décomposer les fonctionnalités du système à développer, telles que pratiquées par le client, afin d'identifier les activités à automatiser et les acteurs impliqués dans leur réalisation. Cela consiste à décrire le système en mettant au clair ses concepts et fonctionnalités opérationnelles, ses événements déclencheurs et centraux, leurs utilisateurs internes et externes ainsi qu'identifier leurs interactions. Généralement, ces éléments sont regroupés en deux catégories : la première consiste à identifier les besoins métiers et utilisateur et sont nommés besoins fonctionnels et la deuxième consiste à identifier les objectifs et besoins business et stratégiques de l'entreprise et sont nommés les besoins non-fonctionnels (Figure 6). (ISO15288, 2008) appuie cette catégorisation et définit ces besoins comme étant des exigences. Les exigences traduisent les besoins et représentent les éléments requis de l'application et par conséquent les contraintes à satisfaire lors de sa conception et son développement. La définition des deux types d'exigences donnée par (ISO15288, 2008) est :

- **Les exigences fonctionnelles** sont les exigences qui définissent les fonctionnalités et les services qu'un système doit fournir. Elles traduisent les opérations à automatiser liées au fonctionnement de l'entreprise
- **Les exigences non-fonctionnelles** caractérisent et précisent comment le système doit satisfaire les exigences fonctionnelles en termes de performance, fiabilité, convivialité, maintenabilité, etc. Elles concernent également les exigences liées au projet de développement comme le coût

et les délais de réalisation. Leur éclaircissement est important car elles constituent la base sur laquelle le choix de conception et d'implémentation est fait.

L'étape du recueil des besoins consiste à restructurer le cahier des charges en identifiant les besoins du système à automatiser en distinguant clairement les exigences fonctionnelles, les exigences non-fonctionnelles et le périmètre du projet. Les supports généralement utilisés sont sous forme de représentation orale uniquement, représentation textuelle ou représentation graphique simple de l'enchaînement du processus à automatiser. Cette étape permet également de faire un tour d'horizon des solutions possibles pour répondre au besoin en écartant celles qui ne correspondent pas aux exigences fonctionnelles ou non-fonctionnelles. Le niveau de pertinence et d'exhaustivité de l'expression du besoin dépend principalement du budget et du temps alloués à la réalisation du projet. Les acteurs possibles de l'analyse du besoin sont le client/décideur (se charge de mettre au clair les critères budgétaires et temporels à prendre en considération), le représentant métier (se charge de transmettre les besoins des utilisateurs finaux), l'expert ou analyste métier (se charge de mettre en œuvre le recueil du besoin, le traduire en fonctions à remplir qu'elles soient fonctionnelles ou non-fonctionnelles) et l'expert technique (se charge d'identifier les fonctions qui ne peuvent pas être remplies pour les écarter de l'analyse). Généralement, la communication peut être difficile entre les différentes parties prenantes car chaque acteur a un vocabulaire spécifique issu de son propre domaine de compétences. De plus, souvent, le client vient avec une technologie précise pour implémenter sa solution, et de ce fait, les besoins à étudier sont plus orientés sur « comment » la solution sera implémentée plutôt que de clarifier de « quoi » a-t-il besoin.

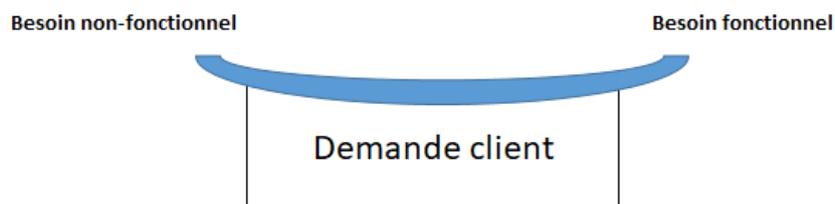


Figure 6: Les types de besoins d'une application

2. L'analyse fonctionnelle et technique du besoin

L'étape d'analyse du besoin consiste à traduire le besoin client détaillé pendant l'atelier du recueil du besoin de manière concrète, factuelle et fidèle. De plus, dans le contexte économique actuel, les enjeux de cette étape sont multiples. Il faut à la fois prendre en considération la satisfaction des parties prenantes, la satisfaction de l'utilisateur final et la conformité aux exigences définies (fonctionnelles et non-fonctionnelles).

2.1. Analyse fonctionnelle

L'analyse conceptuelle du besoin permet de manière générale de déterminer « qui fait quoi et quand » dans le système en détaillant le comportement des acteurs, leurs interactions et les objectifs à atteindre pour chaque étape d'exécution. L'analyse constitue le socle de cette étape. Elle repose sur une étude approfondie et méticuleuse des besoins clients identifiés. Elle permet de bien déterminer les exigences et de clarifier leur catégorie. De plus, elle permet également de faire émerger de nouvelles interrogations et ambiguïtés à résoudre. Les démarches méthodiques adoptées pour ce type d'analyse dépendent de la nature du projet ainsi que du budget et temps alloués.

Vient ensuite la phase de modélisation qui permet de représenter graphiquement les exigences établies, dans le but de mettre au clair les besoins graphiquement. Pour résumer, l'analyse conceptuelle est un processus cognitif qui s'inscrit dans une démarche de clarification et centralisation des besoins exprimés

par le commanditaire. Le résultat de cette analyse apporte des clarifications et explications suffisantes des besoins qui servira de base pour définir les solutions techniques à mettre en œuvre. Le résultat de cette étape permet d'alimenter un document de cadrage fonctionnel ou la spécification fonctionnelle pour centraliser les informations décrivant les caractéristiques du produit digital à développer et ainsi faciliter l'analyse technique.

2.2. Analyse technique

Les fonctions opérationnelles et leurs exigences clairement identifiées, définies et validées par le commanditaire, constituent la base sur laquelle l'analyse technique s'appuie. L'analyse technique permet d'étudier les solutions d'implémentation possibles en réfléchissant aux différents moyens existants pour réaliser les différentes fonctions du système. Elle consiste à déterminer la conception de l'architecture matérielle, l'architecture logicielle de l'application, conception des interfaces, etc. Partant du fait, qu'aucune application ne peut répondre et ne peut regrouper toutes les contraintes d'un système à la fois, plusieurs solutions sont envisagées, de la plus simple à la plus élaborée et de la plus générale à la plus spécifique. La solution technique choisie avec le client est souvent un compromis entre toutes les contraintes déterminées au début du projet. Le résultat final de cette analyse sert de référence tout au long du développement du système. Cela consiste à décrire et détailler la solution retenue dans un document de « spécification technique » (décrire l'architecture de la solution, ses interfaces, ses interactions, etc.)

Les résultats des deux étapes sont consignés dans un document de « spécification fonctionnelle » et un document de « spécification technique ». Il est à noter que l'étape d'analyse est un processus continu et itératif, où les échanges et les demandes de validation sont récurrent car plus on avance dans l'analyse, plus on clarifie et affine le besoin du produit digital à réaliser. Comme l'étape du recueil de besoin, dans l'étape d'analyse, les supports textuels, images graphiques et dessins graphiques sont les plus utilisés.

3. Elaboration et intégration de la solution

L'élaboration ou l'implémentation de la solution vient après la validation des spécifications techniques par le client. L'implémentation de la solution est généralement divisée en deux parties, la partie de programmation et la partie des tests.

3.1. Développement et test de la solution

Codage ou phase d'implémentation de la solution choisie suite à la validation des résultats de l'analyse technique par le commanditaire consiste à produire le code source de l'outil à réaliser, c'est-à-dire traduire la conception technique choisie dans le langage de programmation choisi interprétable par l'environnement d'exécution du client.

La phase de test débute une fois la phase de programmation terminée et consiste à évaluer si la solution développée satisfait la spécification élaborée lors de la phase d'analyse. Le test s'intéresse à la partie fonctionnelle de la solution (le comportement des fonctions de la solution) et à sa partie technique. L'objectif est double : vérifier que l'outil est conforme aux exigences établies et minimiser l'apparition des anomalies fonctionnelles ou techniques de la solution en production chez le client. Pour ce faire, des jeux de scénario de test sont définis en se basant sur les spécifications fonctionnelles et techniques validées. Le plan de test est rédigé par l'analyste fonctionnel (métier) et l'analyste technique et est validé par le client. La phase de test est généralement divisée en trois parties :

- Les tests unitaires consistent à évaluer les résultats techniques de chaque composant de la solution, individuellement pour évaluer s'ils sont conformes à la conception technique détaillée et ainsi détecter le plus tôt possible d'éventuelles erreurs
- Les tests fonctionnels permettent de tester toutes les règles métiers établies dans la spécification pour évaluer la conformité des comportements fonctionnels attendus de la solution
- Les tests d'intégration consistent à tester le bon fonctionnement de la mise en œuvre conjointe des différents composants élémentaires de la solution dans un environnement semblable à celui de la production

Même si l'application a été testée en s'appuyant sur un plan de test approuvé par le client, il arrive qu'en production, les utilisateurs constatent des écarts importants entre leur besoin effectif et l'application mise à leur disposition.

3.2. Maintenance de la solution

Le déploiement de la solution développée consiste à mettre en place toutes les opérations nécessaires à son fonctionnement dans la plateforme de production. La première étape consiste à installer la solution dans son environnement d'exploitation chez le client en y transférant tous ses composants élémentaires et nécessaires à son fonctionnement (bibliothèques, exécutables, etc.). Puis vient l'étape de la configuration où les paramètres requis au fonctionnement de la solution sont configurés en fonction de l'environnement de production du client. Lors de cette phase, plusieurs paramètres entrent en considération : les caractéristiques des technologies utilisées pour développer la solution, les caractéristiques de l'environnement de production du client ainsi que les moyens d'accès aux réseaux de cet environnement. De ce fait, le déploiement d'une solution requiert un plan d'action préalablement défini et qui prend en compte les différentes caractéristiques et risques à maîtriser. En outre, le plan inclut également la formation technique et fonctionnelle de la solution pour que les administrateurs puissent se l'approprier plus facilement.

Après le déploiement, la solution est opérationnelle dans son environnement de production, son comportement est surveillé et est modifié si nécessaire. Il s'agit de la phase de maintenance où il est parfois essentiel de modifier la solution développée pour corriger des anomalies, pour améliorer ses caractéristiques non-fonctionnelles, ou pour répondre à de nouveaux besoins. La maintenance est catégorisée en quatre types :

- La maintenance corrective fixe les anomalies de la solution développée (bugs) où elle ne fonctionne pas comme prévu selon la spécification fonctionnelle ou technique
- La maintenance adaptative fixe les changements requis pour que la solution fonctionne correctement dans la nouvelle version de l'environnement d'exploitation (mis à jour ou changement de l'environnement de production)
- La maintenance perfective fixe les changements liés à la satisfaction des nouveaux besoins identifiés. Il s'agit des besoins non mandatés dans les spécifications fonctionnelles et techniques
- La maintenance préventive peut être définie comme étant un mécanisme mis en place pour empêcher la survenue des anomalies

4. Méthodologies de gestion de projet de développement d'un système par une SSII

Pour pouvoir gérer les différentes étapes d'automatisation du processus, plusieurs approches de gestion de projet ont été proposées : cycle en cascade, cycle en V, le cycle en spirale, « Dual vee cycle », les

méthodes agiles, etc. Nous allons présenter brièvement, les méthodes les plus courantes, à savoir : Cycle en cascade, Cycle en V et méthodes agiles.

4.1. Cycle en cascade

Le cycle en cascade est une méthode d'organisation des projets de développement d'un système qui comporte 7 phases (Figure 7) incluant : Exigences du système, exigences logicielles, analyse, conception, implémentation, test, exploitation. Son principe est un enchaînement linéaire d'étapes avec un retour possible aux précédentes si nécessaire, chaque étape se terminant à une date limite définie. Une nouvelle phase ne peut débuter que si la précédente est complètement finalisée. Elle convient aux projets dont les besoins sont clairement définis et stables. Dans ce cycle, le client ne valide le produit développé qu'à la fin du cycle.

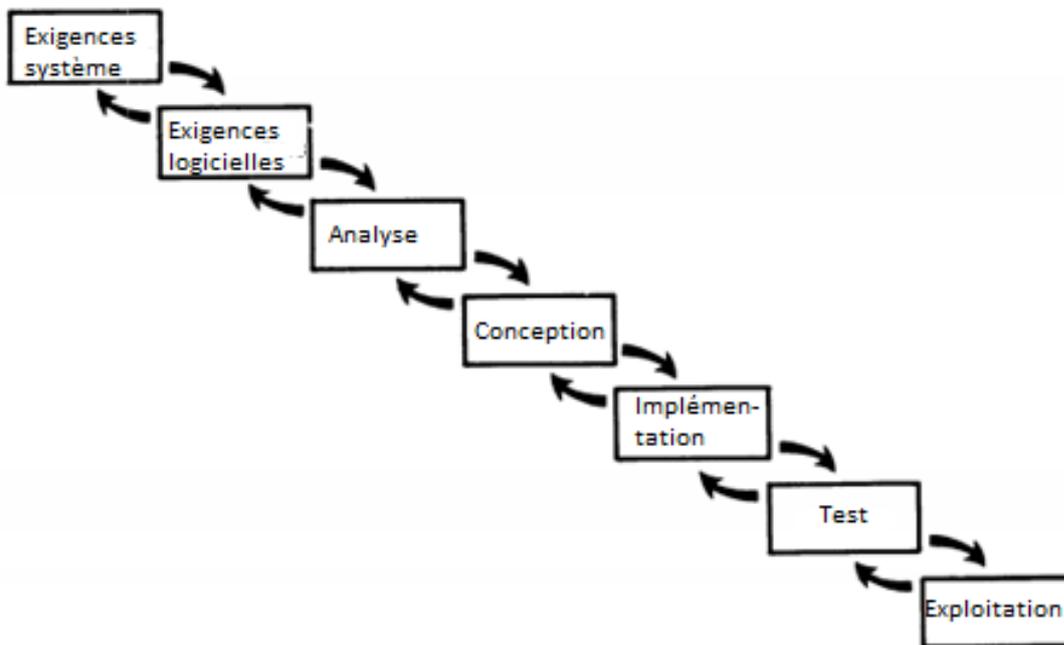


Figure 7: Cycle en cascade

4.2. Cycle en V

Pour pallier le manque de réactivité du cycle en cascade, le cycle en V a été proposé où chaque livrable doit être testé. Le cycle en V est une méthode d'organisation des projets de développement d'un système. Il peut être distingué par trois parties principales : la phase de conception, la phase de réalisation et la phase de validation. Comme le montre la Figure 8, la branche descendante du cycle contient toutes les étapes de la conception du projet, et la branche montante toutes les étapes de test du projet. Quant à la pointe du V, elle représente le codage du projet. En effet chaque étape de la conception (branche descendante) correspond à une étape de test (branche montante). En cas de dysfonctionnement, cette méthodologie permet d'éviter un retour aux étapes antérieures. Les étapes de la partie ascendante renvoient de l'information aux étapes adjacentes en cas d'erreurs détectées pour avoir une meilleure qualité du logiciel. A l'instar du cycle en cascade, le cycle en V convient aux projets clairement définis et le client ne valide le produit final qu'à la fin du cycle.

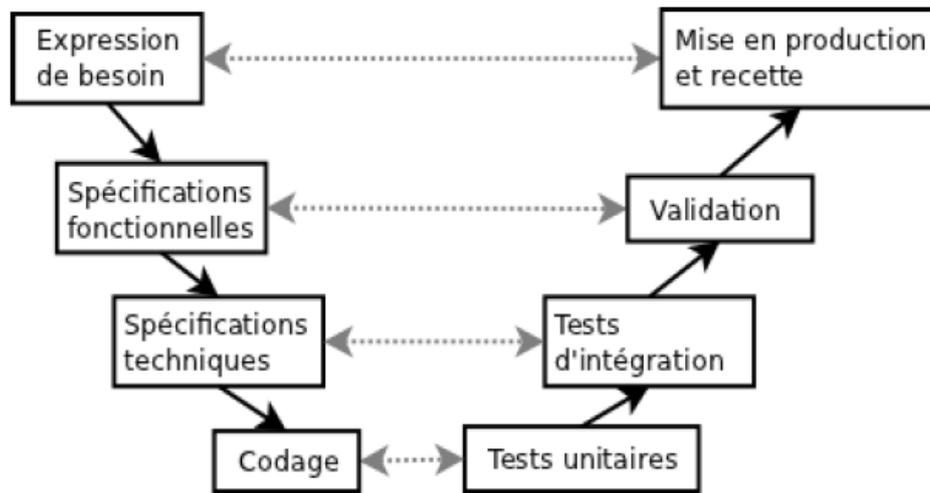


Figure 8: Cycle en V

4.3. Méthodes agiles

Les méthodes agiles se veulent plus souples et flexibles que les approches précédemment citées, leur but est de placer les besoins du client au centre du projet. En 2001, un consensus a été établi, en formalisant les principes essentiels de la méthode en un manifeste nommé « Le manifeste Agile ». Malgré les valeurs présentées par ce manifeste, on ne trouve pas de définition commune de la méthode dans le domaine du génie logiciel. Elle est généralement définie comme étant une méthode qui repose sur le principe de l'adaptation au changement en offrant une flexibilité d'adaptation tout au long du projet ce qui permet de faire évoluer le projet dans des directions différentes selon les exigences ou problématiques rencontrées. Il s'agit d'une méthode itérative et participative, où l'avancement du développement de l'application se fait en validant élémentairement chacun de ses composants. Les deux méthodes agiles les plus utilisées sont « Extreme Programming » et « Scrum » (Tarhan and Yilmaz, 2014).

III. Questionnement

Pour résumer le contexte de notre étude, auparavant, les entreprises adoptaient une structure organisationnelle verticale caractérisée par une division classique par fonction, où chacune est dirigée par un chef fonctionnel alors que l'ensemble est coordonné par la direction générale. Parmi les inconvénients de cette approche, on trouve le cloisonnement de l'information et le ralentissement du processus décisionnel du fait des échelons hiérarchiques. Ces inconvénients entravent la réussite des projets d'automatisation de leur système d'information. C'est la raison pour laquelle les entreprises adoptent de plus en plus un raisonnement par processus pour accroître leur compétitivité. Le système d'information constitue le socle de chaque organisme pour supporter ces processus, d'où le recours des entreprises aux SSII pour les aider à mettre en place les applications adéquates à leur besoin : les outils de workflow. Dans cette thèse, nous nous concentrons sur l'automatisation des processus métiers. Le développement des processus métiers par une SSII suit le cycle classique du développement des applications (intranet, site web, etc.), du recueil du besoin client à la maintenance de la solution produite. Ce cycle de développement est géré par différentes méthodes qui ont pour objectif d'assurer l'organisation et la communication entre les différentes parties prenantes. La conduite de ces projets est souvent dictée par une orientation technique. De ce fait, les solutions décrites sont des fois incompréhensibles et incohérentes ce qui engendre une perte de temps considérable pendant la phase de développement, et des écarts par rapport aux attendus clients. Ces écarts peuvent engendrer des coûts de maintenance importants voir même des demandes de refonte des solutions récemment développées.

Dans ce contexte, notre travail de recherche pointe en particulier les limites des pratiques existantes dans les SSII pour l'automatisation des processus métiers. Cette problématique soulève plusieurs questions de recherche que nous détaillons ci-après :

Une SSII travaillant avec plusieurs clients différents de secteurs différents, la spécification textuelle et graphique du processus à développer constitue la base de communication et validation des objectifs à atteindre.

Q1. Existent-ils d'autres techniques qui permettent de définir un modèle du processus métier applicable à différents domaines d'activité ? Ces techniques sont-elles compréhensibles par les utilisateurs de compétences métiers et techniques différentes ?

Comme vu dans la définition d'un SI, un SI utilise plusieurs types de ressources : humaine, matérielle et logicielle.

Q2. Comment les différents types de ressources sont représentés d'un point de vue opérationnel ?

Dans la pratique courante, l'analyse du processus se fait principalement d'un point de vue fonctionnel alors que les besoins d'automatisation des processus sont fonctionnels et non-fonctionnels.

Q3. Est-ce que la représentation et modélisation des propriétés fonctionnelles uniquement dans la phase d'analyse est suffisante pour étudier les performances du système à concevoir ? Si non, comment prendre en charge les besoins fonctionnels et non-fonctionnels à la fois ?

L'analyse du processus à automatiser se fait principalement en utilisant des supports textuels et graphiques.

Q4. Dans le cas d'analyse d'un processus, le modèle et la représentation statiques sont-ils suffisants ?

Le développement de la solution à automatiser se fait suite à l'analyse technique qui s'appuie sur les résultats de l'analyse fonctionnelle consignés dans le document de « spécification fonctionnelle »

Q5. Comment lier les différents résultats des différents points de vue d'analyse ?

Chapitre II. ETAT DE L'ART

I. Les processus métiers.....	16
1. Gestion des processus métiers.....	16
2. Workflow	17
3. Modélisation des processus métiers.....	19
4. Simulation des processus métiers.....	23
II. Extensions des langages semi-formels.....	27
1. Extension orientée domaine spécifique.....	29
2. Extension orientée activité, données et flux de contrôle	30
3. Extension orientée ressource.....	33
III. Ingénierie dirigée par les modèles.....	35
1. MDE.....	35
2. MDA.....	37
3. MDI.....	39
4. MDSEA.....	40
IV. Conclusion	42
1. Réponse aux questions de recherche	42
2. Synthèse	46

Le système d'information constitue le pivot de communication dans une entreprise, de ce fait, assurer la performance et la fiabilité des processus métiers qui le composent est important pour atteindre les objectifs de gain, de qualité et de réactivité escomptés. Dans ce chapitre, nous allons présenter l'état de l'art que nous avons effectué pour voir jusqu'à quel point la littérature scientifique permet de répondre aux questions posées dans le chapitre « Contexte et problématique ». Pour ce faire, nous avons essayé de couvrir les différents concepts liés à l'automatisation des processus métiers

I. Les processus métiers

1. Gestion des processus métiers

1.1. Qu'est-ce qu'un processus

Plusieurs définitions sont attribuées au mot processus. L'une des définitions données dans le contexte des processus métier est : « Ensemble d'activités corrélées ou interactives qui transforment des éléments d'entrée en éléments de sortie ». Le processus répond aux questions : Quoi ? Pour quelle valeur ajoutée ? (Brandenburg and Wojtyna, 2006). Ainsi, un processus constitue l'ensemble des activités mises en œuvre par l'entreprise depuis l'expression d'un besoin client (entrée) jusqu'à sa satisfaction (sortie) (Debauche and Mégard, 2004). La Figure 9 représente les principaux éléments liés à un processus. L'exécution d'un processus est déclenchée par les éléments et les événements d'entrée. Cette exécution est effectuée par les ressources impliquées afin de fournir des éléments de sortie qui doivent être en adéquation avec les objectifs à atteindre par le processus.

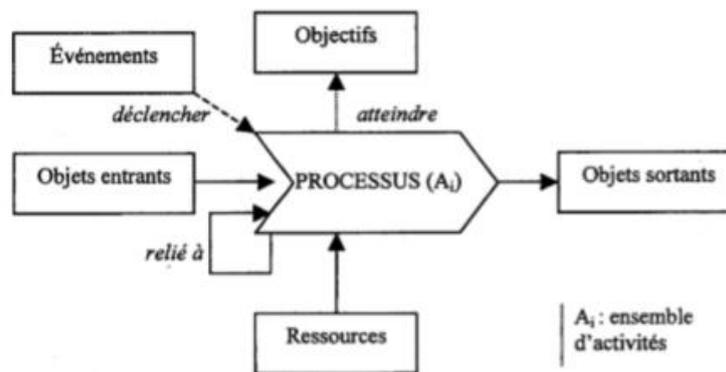


Figure 9: Définition d'un processus

1.2. Business Process Management (BPM)

Actuellement, la circulation de l'information a pris toute son importance, d'où l'émergence des technologies de l'information et de la communication (IT). Pour qu'une entreprise puisse atteindre ses objectifs d'une manière efficace et efficiente, elle doit allier sa vision métier à son système d'information. Le BPM « Business Process Management » est l'une des disciplines les plus répandues dans le domaine. Son objectif est d'obtenir une meilleure vue globale de l'ensemble des processus métiers de l'entreprise et de leurs interactions dans le but de les améliorer avant leur automatisation quand cela est possible (Van der Aalst et al., 2003). BPM est une méthodologie sous forme d'un cycle de vie continu et bouclé comme le montre la Figure 10 divisée en quatre étapes : la conception, la mise en œuvre, la gestion et l'analyse des processus.

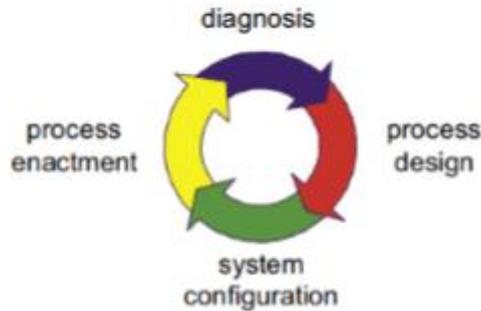


Figure 10: Cycle BPM (Wil M. P. van der Aalst et al., 2003)

Phase de conception « process design » : La première phase consiste à modéliser les processus métiers existants et après leur analyse modéliser les processus métiers cibles

Phase de configuration « system configuration » : Cette phase consiste à transformer les processus modélisés en processus exécutables (Définir le format des messages échangés, les connexions aux services et les applications invoquées.)

Phase d'exécution « process enactment » : Cette phase correspond à l'exécution du processus automatisé

Phase de diagnostic « diagnosis » : Cette étape consiste à utiliser les outils d'analyse et de surveillance pour améliorer les processus

2. Workflow

2.1. Définition

La notion de workflow a émergé à la fin des années 1960 (Dumas et al., 2005), dans le cadre d'évolution des systèmes d'information (Van Der Aalst and Van hee, 2002). Un workflow est traduit littéralement en français par « flux de travail ». Plusieurs définitions ont été proposées dans la littérature au concept de workflow, on cite celle de la Workflow Management Coalition (WFMC) qui est une organisation qui définit les normes concernant le développement des outils de workflow. Un workflow est « l'automatisation partielle ou complète d'un processus, au cours duquel des documents, des informations, des tâches passent d'un participant à un autre, au sein d'un groupe de travail, en conformité avec un ensemble de règles » (WFMC, 1999). Ainsi un workflow consiste à automatiser les flux d'information d'un processus en fournissant à chaque acteur les informations nécessaires à l'exécution des activités qui le composent en suivant des règles procédurales. Les termes workflows et processus peuvent être confondue (Leymann and Roller, 2000), un workflow décrit une suite de tâches exécutées automatiquement par une machine. Un processus quant à lui représente une succession de toutes les tâches, qu'elles soient exécutables ou non par la machine. C'est la raison pour laquelle, le processus est modélisé en premier pour identifier les parties qui peuvent être exécutées par un moteur de Workflow.

2.2. Dimension du workflow

Un workflow a trois dimensions :

La dimension de cas représente une instance (un scénario) du workflow où chaque cas est traité individuellement. Les cas s'influencent entre eux indirectement et uniquement via le partage

des ressources et des données. Un cas représente également un chemin parcouru parmi l'ensemble des chemins du processus pendant la phase d'exécution,

La dimension du processus représente la définition des tâches et leur enchainement (séquentielle, parallèle...),

La dimension des ressources représente les ressources qui exécutent la tâche. Ces ressources représentent un rôle (ressources basées sur des capacités et exigences fonctionnelles) ou une unité organisationnelle (équipe, département, etc.).

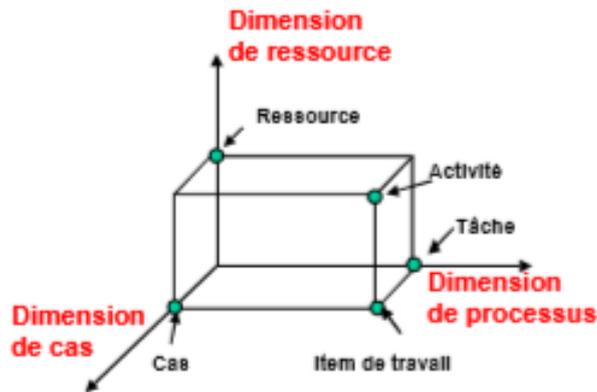


Figure 11: Dimensions des systèmes Workflow (Zacharewicz, 2006)

Le workflow peut être visualisé sous forme d'un nombre de points dans la vue en trois dimensions (Figure 11). Chaque point représente soit un élément de travail (cas + tâche) ou une activité (cas + tâche + ressource)

2.3. Classification des workflows

Plusieurs classifications de workflows existent dans la littérature due à la diversité des critères qui peuvent être pris en considération (domaine d'application, degré de structuration...). La classification proposée par (McReady, 1992) est la plus répandue dans la littérature scientifique, elle a été reprise par (Van Der Aalst, 1998) et depuis elle demeure la plus adoptée. Elle distingue 4 catégories de workflow :

- **Workflow de production** correspond à l'automatisation des processus opérationnels connus et généralement fixes, où la valeur ajoutée de l'entreprise dépend de leur performance. Ce type de workflow est souvent répétitif. La difficulté d'automatisation de ce type de workflow réside dans la structure du processus qui fait appel à des informations provenant des systèmes d'information variés,
- **Workflow administratif** correspond à l'automatisation des processus administratifs. Il s'agit d'un processus facilement automatisable grâce aux règles d'enchainement des tâches qui sont simples et clairement définies. Leur gestion du routage des documents, le remplissage et la circulation des formulaires permettent d'alléger les tâches de bureau et de réduire les erreurs humaines,
- **Workflow ad-hoc** appelé aussi workflow adaptable, correspond à l'automatisation des processus qui ne sont pas fixes et qui doivent s'adapter aux changements de l'environnement. Chaque étape doit être décidée au fur et à mesure, ce qui rend le workflow dynamique,

- **Workflow collaboratif** permet de gérer un travail de groupe pour atteindre un objectif commun. Ce type de workflow est axé sur la communication et le partage d'information et utilise souvent des outils « Groupwares ».

2.4. Concepts fondamentaux des workflows

La Figure 12 représente le diagramme du méta modèle d'un workflow proposé par la WFMC (Hollingsworth, 1995). Elle présente les principaux concepts et terminologies liés au workflow ainsi que leur interaction :

- **Activité** représente une action atomique du workflow qui permet la transformation d'un élément d'entrée en un élément de sortie. Elle peut être manuelle ou automatique
- **Acteur (ressource)** représente une entité du modèle organisationnel participant à l'exécution du workflow. Elle est chargée de réaliser les activités qui lui sont assignées via les rôles qui lui sont définis dans le modèle. Elle peut être humaine ou matérielle
- **Rôle** représente les compétences nécessaires à une ressource pour exécuter une activité du workflow
- **Donnée** représente une information en rapport avec la réalisation d'une activité. Elle est en entrée et en sortie des activités et détermine la prochaine activité à déclencher,
- **Condition de transition** représente le critère de changement d'état ou de progression de l'activité courante à l'activité suivante
- **Application externe** représente l'application informatique externe nécessaire à la réalisation de l'activité

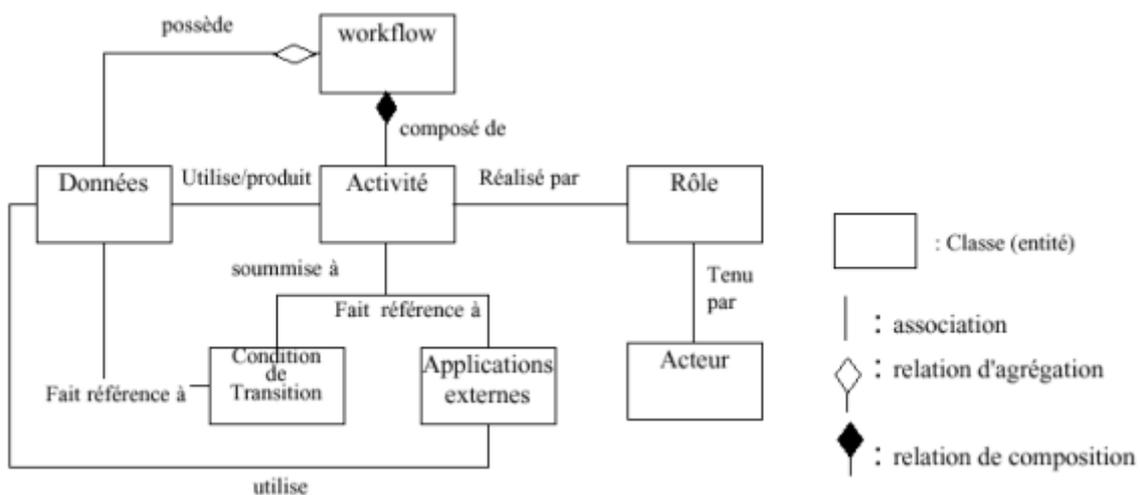


Figure 12: Méta modèle du Workflow selon (Hollingsworth, 1995)

3. Modélisation des processus métiers

3.1. Modélisation d'entreprise

Il existe plusieurs définitions attribuées au mot « modèle » du fait de la diversité des domaines l'utilisant. L'une des définitions communes utilisée dans le cadre de l'ingénierie du logiciel est donnée par (Vernadat, 1999) : Un modèle est « une représentation d'une abstraction d'une partie du monde réel, exprimée dans un langage de représentation ». Ainsi la modélisation est l'outil

conceptuel qui permet de représenter de manière intelligible un besoin qui peut sembler complexe (Varenne and Silberstein, 2013). Ainsi, la modélisation d'entreprise est la représentation du fonctionnement de l'entreprise via des concepts capables de décrire toutes ses dimensions (Doumeingts et al., 2000). L'avantage de la modélisation d'entreprise repose sur le fait que les travaux de réingénierie et de refonte de ses activités nécessitent une compréhension globale et claire de son état de fonctionnement actuel. Ce qui permet d'analyser, identifier les anomalies et prendre les bonnes décisions sur les changements à effectuer. Plusieurs langages et méthodes de modélisation sont proposés dans la littérature pour comprendre et analyser l'état actuel du système afin d'améliorer sa performance. Ils permettent de décrire les différents aspects d'une entreprise quel que soit son secteur d'activité (Kettinger et al., 1997): SADT « Structured analysis and design technique », IDEF « Integration Definition: IDEF0...IDEFx », la méthode et les modèles GRAI et ARIS (Architecture of Integrated Information System), etc. (Vernadat, 1997). Certaines méthodes de modélisation sont générales, d'autres sont orientées objet, d'autres orientées processus, etc. Le choix de l'une des méthodes se fait en fonction de l'objectif à atteindre. Dans nos travaux de recherche nous nous focalisons sur la modélisation orientée processus.

3.2.Modélisation des processus

D'après (Weske, 2007), la modélisation des processus est considérée comme étant la base de BPM. Comme vu dans les paragraphes précédents, elle permet de faire apparaître les éléments constitutifs d'un processus en représentant l'ensemble des interactions des activités effectuées par ses acteurs. Plusieurs recommandations de modélisation des processus sont présentées dans la littérature scientifique pour améliorer sa pertinence et sa compréhension par toutes les parties impliquées. Comme l'indique (Dumas et al., 2005), le périmètre de modélisation des processus peut être classé en deux catégories : la conception organisationnelle « organizational design » consiste en une représentation qui simplifie la complexité des interactions entre les activités et leurs ressources en évitant de décrire les détails d'implémentation ; quant à la conception applicative « application design », elle décrit les détails d'implémentation du processus, cette approche est habituellement utilisée avant la phase d'automatisation et d'implémentation du processus. Généralement, la majorité des chercheurs scientifiques recommande de définir l'objectif de modélisation en premier, puis, modéliser le processus du plus haut niveau d'abstraction au plus bas.

3.2.1. Aspects à modéliser

Selon (Tudor, 2016), la représentation d'une même réalité métier n'est pas unique ou universelle. Il est possible d'avoir des représentations différentes en fonction des points de vue et aspects envisagés. Plusieurs aspects de modélisation sont identifiés dans la littérature scientifique (Curtis et al., 1992) ont identifié 4 principaux aspects : fonctionnel, comportemental, organisationnel et informationnel. (Krogstie, 2012) en a identifié quatre de plus : aspect orienté sur l'objectif et les règles « Goal and rule perspective », aspect orienté objet « Object oriented perspective », aspect orienté communication « Communication perspective » et aspect orienté topologique « Topological perspective ». Les aspects de modélisation identifiés par (Curtis et al., 1992) sont les plus utilisés de par leur simplicité et adaptabilité à tous les secteurs d'activités :

- **L'aspect fonctionnel** concerne l'identification et la distinction des étapes opérationnelles (activités) du processus à modéliser. Cela consiste à représenter la hiérarchie des activités (sous processus), les flux de données associés et également les interdépendances de données entre les activités (data flow)
- **L'aspect comportemental** est considéré comme étant primordial puisqu'il est lié à la dynamique du processus. Il s'agit de modéliser le flux de contrôle : chronologie des activités, leurs flux (séquentiel ou parallèle), les événements qui permettent de déclencher les activités, conditions de déclenchements...
- **L'aspect informationnel** concerne l'ensemble des données et des informations qui sont associées aux activités. Il décrit les interrelations existantes entre les données, leur type et leur structure...
- **L'aspect organisationnel** concerne la description de l'organisation des acteurs de l'entreprise, qu'ils soient des ressources humaines ou unité organisationnelle (département, service, etc.)

3.3. Langage de modélisation des processus

La modélisation d'un processus métier est exprimée à l'aide d'un langage avec lequel on construit, examine et manipule les modèles. Les langages de modélisation peuvent être classés selon trois formalismes : informel, semi-formel et formel (Freire et al., 1997).

- **Langage de modélisation informel** utilise le langage naturel comme la parole et l'écriture pour décrire un processus. La construction du modèle ne suit aucune restriction. L'inconvénient majeur de ce type de langage est que le raisonnement humain est le facteur principal pour l'analyse et la validation ce qui peut engendrer des erreurs de compréhension et de vérification ce qui rend la modélisation imprécise et parfois ambiguë
- **Langage de modélisation semi-formel** utilise comme outil de modélisation un langage textuel ou graphique pour lequel une syntaxe précise est définie pour spécifier les conditions et combinaisons permises. Leur sémantique est faiblement définie mais permet une certaine dose de contrôle et d'automatisation. Même s'il permet une large expressivité et une compréhension plus facile par rapport au langage informel, le manque d'une sémantique complète engendre un manque de précision et de clarté
- **Langage de modélisation formel** est un langage rigoureux basé sur un modèle mathématique, des notations formelles avec une sémantique riche et précise. Leur principal avantage est leur capacité à exprimer une signification précise et leur possibilité de vérifier la cohérence et la complétude d'un système

3.4. Langage de modélisation des processus

3.4.1. BPMN

BPMN « Business Process Modeling Notation » est une notation graphique standardisée pour la modélisation des processus métier. C'est un langage basé sur un simple enchaînement logique de composants pour décrire le besoin à représenter, il permet donc de réaliser des modèles selon une vue comportementale. Le but principal est de fournir une annotation standardisée compréhensible par tous les acteurs d'une entreprise, à la fois pour les utilisateurs métiers (managers, employés, etc.), les analystes métiers et les développeurs. Le langage a été

développé en premier par BPMI (Business Process Management Initiative) en combinant les meilleurs principes des autres normes comme UML Activity Diagram, UML EDOC Business Processes, IDEF, Event-Process-Chains, etc. BPMN est aujourd'hui maintenu par l'Object Management Group (Object Management Group, n.d.) et constitue un standard de facto de la pratique professionnelle (Braun et al., 2014a). Les éléments de modélisation de BPMN peuvent être composés en quatre principales catégories (Kherbouche, 2013):

- Les objets de flux « Flow Objects » : il s'agit des éléments graphiques qui permettent de définir le comportement d'un processus métier : les activités, les événements et les passerelles
- Les objets de connexion « Connecting Objects » : il s'agit des éléments graphiques qui permettent de relier les objets de flux pour représenter le cheminement du processus : les flux de séquence, les flux de message et les associations
- Les couloirs « Swimlanes » : il s'agit des éléments qui permettent de regrouper et structurer tous les composants du diagramme BPMN : Partitions « Pools » et « Lanes »
- Les artefacts « artefacts » : il s'agit des éléments qui permettent de fournir des informations supplémentaires sur le processus : les objets de données représentent les données liées à une tâche, les groupes permettent de regrouper des tâches d'une même catégorie pour faciliter leur repérage et les annotations permettent d'ajouter des commentaires pour expliciter une information

3.4.2. UML

UML « Unified Modeling Language » est un langage de modélisation orientée objet, utilisé principalement dans le développement informatique. Il a été développé dans le but de fournir une notation graphique pour représenter les spécificités d'un système, l'analyser et le documenter. La notation graphique d'UML est standardisée et se compose de treize types de diagrammes qui peuvent être utilisés pour modéliser différents aspects d'un système. Le diagramme d'activité est le plus adéquat et le plus couramment utilisé pour modéliser les processus (Wohed et al., 2005). Il permet de décrire le comportement des processus sous la forme de flux ou d'enchaînement d'activités. Il est sémantiquement proche des diagrammes d'états-transitions où chaque état représente l'exécution d'une activité.

Le diagramme d'activité est composé principalement de :

- Les éléments d'Etat représentent les événements qui doivent avoir lieu pour déclencher l'exécution d'une activité
- Les éléments d'Activité représentent les fonctionnalités à exécuter
- Les éléments d'Action représentent la plus petite unité de fonction exécutable au sein d'une activité
- Les éléments de Transition permettent de relier les différents composants du diagramme
- Les Couloirs permettent d'organiser le diagramme d'activité selon les différents responsables des activités et actions représentées

3.4.3. EPC

EPC « Event-driven Process Chain » est un langage de modélisation graphique développé dans le cadre de l'outil ARIS (Architecture of integrated Information System) (Aris, 2007). Il représente un composant important des concepts de modélisation SAP. Il est également utilisé

pour la description des processus métiers en les représentant comme une succession de fonctions et d'évènements. EPC considère les évènements comme des déclencheurs ou des résultats d'une étape du processus. Les déclencheurs EPC résultant d'une étape du processus sont appelés fonctions. Ainsi, la structure du flux est sous forme de « Evènement-Action-Evènement ». D'un point de vue graphique EPC est un diagramme constitué d'éléments représentant les fonctions (activités), les unités organisationnelles (les participants), les évènements, les opérateurs logiques et des objets divers.

3.4.4. IDEF3

IDEF « Integration DEFinition Methods » est une famille de méthodes développées à l'origine par l'armée américaine pour élaborer l'architecture de systèmes industriels. Elle se compose de quinze langages qui couvrent un large éventail d'utilisation (d'IDEF0 à IDEF14). IDEF3 est le langage de description graphique des processus métiers (Marier, 1996). Elle permet de décrire le processus selon les rapports de priorité et de causalité entre les activités et les évènements de ce processus. Le langage est composé de quatre principaux éléments :

- Les unités de comportements (UDC) représentent les entités ou artefact pouvant être produit par le système
- Les liens utilisés pour relier les UDC. IDEF3 abandonne tous les autres types de flèches indiquant les contraintes sur les activités ou les ressources allouées
- Les connecteurs logiques (ET, OU, et OU exclusif)
- Les références

4. Simulation des processus métiers

4.1. Définition

Comme vu dans le chapitre précédent, la modélisation des workflows est une étape importante dans la réalisation des projets d'automatisation des processus. Elle permet d'améliorer la compréhension de la complexité des interactions des activités et des ressources affectées à chacune d'elle. Or, l'analyse et la vérification d'un modèle statique peut s'avérer compliquée et même parfois fastidieuse d'où l'intérêt d'utiliser la simulation. La simulation ajoute un fonctionnement interactif au modèle, afin de connaître son comportement dynamique et de pouvoir prédire son comportement futur. Elle réagit aux consignes qu'on lui donne (modèle et variables d'entrée) et permet d'étudier et comparer les scénarios possibles pour évaluer et analyser le comportement du processus (van der Aalst, 2015). Elle permet également de fournir des renseignements sur les conséquences des changements et des modifications du processus, avant que ceux-ci ne soient implémentés réellement. Il est important de souligner qu'il s'agit d'un outil d'aide à la décision et non un outil de résolution de problème. La Figure 13 proposé par (Revel et al., 2004) illustre de manière simple les composants d'un processus de simulation itératif. La modélisation fournit un modèle conceptuel, la programmation permet d'ajouter le comportement interactif au modèle, la partie d'expérimentation permet de jouer les différents scénarios de modification envisagée et pour finir l'analyse des résultats qui permet d'évaluer le processus modélisé.

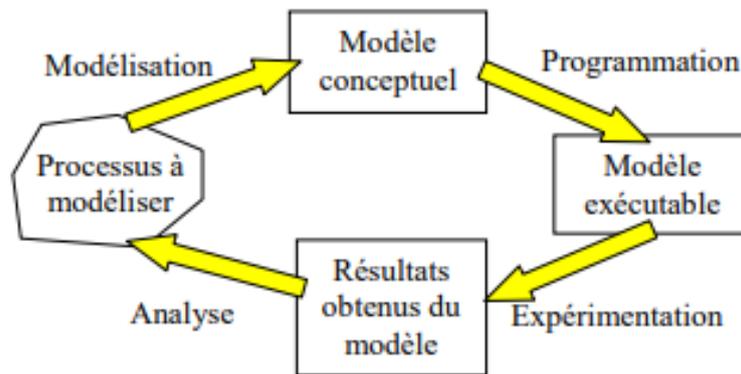


Figure 13: Les composants d'un processus de simulation

Comme évoqué ci-dessus, la simulation est uniquement un outil qui aide l'utilisateur à déduire les solutions les plus adéquates au besoin exprimé. D'où l'importance de vérifier la validité du modèle conceptuel et des données d'entrées utilisées sous peine d'obtenir des résultats incomparables avec ceux obtenus sur le système réel.

4.2.Types d'objectif d'analyse d'une simulation

L'objectif d'analyse de la simulation des processus métiers peut être classé en trois catégories :

- **L'analyse de validation et vérification du modèle** consiste à détecter les erreurs et les failles de conception d'un modèle. Il s'agit d'une analyse qualitative visant à évaluer la bonne utilisation des éléments de modélisation et les flux les liant pour représenter un processus (Verbeek, 2001). Les erreurs détectables dans ce type d'analyse peuvent être classifiées en deux catégories, erreurs de structure et erreurs de syntaxe (Awad and Puhlmann, 2008). Une erreur de syntaxe est une erreur liée au non-respect de la sémantique de la notation du langage de modélisation utilisé (le manque de rigueur dans la prise en considération de la signification d'une notation). L'erreur de structure est une erreur d'agencement et de séquençement des flux des notations utilisées qui peut par exemple induire à une boucle infinie. La vérification structurelle est l'une des problématiques les plus traitées par les chercheurs et est généralement basée sur une des deux approches comme indiqué par (Jander et al., 2011) : la première approche consiste à utiliser des langages de modélisation formels comme Petri nets (Van der Aalst, 2000) alors que la deuxième approche consiste à transformer ou à enrichir un langage semi-formel comme BPMN en un langage répondant aux caractéristiques formelles (M.Dijkman et al., 2007)
- **L'analyse fonctionnelle** consiste à vérifier si le modèle du processus répond à toutes les exigences opérationnelles en jouant les différents cas d'utilisation du processus. Cela consiste à définir les cas d'utilisation fonctionnels sous forme de cas de test. Un cas de test consiste à vérifier que pour une valeur d'entrée donnée, le séquençement des activités et les valeurs des résultats attendus sont correctes. Ainsi, l'objectif est de vérifier la pertinence opérationnelle du modèle en vérifiant la relation entrée/sortie (Jander et al., 2011; Speck et al., 2003)
- **L'analyse de la performance** est utilisée pour évaluer la performance du processus et pour étudier l'impact des changements effectués sur la rentabilité en jouant différents

scénarios. Ce type de simulation est considéré comme une aide intéressante aux décideurs. Parmi les approches d'analyse de la performance les plus utilisées dans le domaine des processus métiers : la simulation des systèmes dynamiques, la simulation par événements discrets, la simulation à base d'agents et la simulation analytique. *La simulation des systèmes dynamiques* permet d'évaluer l'évolution du comportement du processus dans le temps à partir de ses éléments, de leurs interactions et les dynamiques qui les dirigent. Autrement dit, il s'agit d'évaluer la relation causale entre les activités d'un processus et les facteurs qui impactent son exécution (Lianjun An and Jun-Jang Jeng, 2005). Le but de cette approche est de comprendre l'association entre la structure et la conduite de l'ensemble du processus afin de changer la structure pour améliorer la conduite. *La simulation à évènements discrets* consiste à décrire le processus sous forme d'enchaînement d'évènements liés entre eux par des transitions. L'évolution d'un état à un autre se fait conformément à l'arrivée des événements caractérisant le changement d'état du processus. La simulation à évènements discrets génèrent des métriques de performance en se focalisent sur la conduite individuelle d'une activité et des ressources qui l'exécutent (Čerić and Hlupić, 1993). *La simulation à base d'agents* consiste à modéliser le processus sous forme d'agents en décrivant la conduite de chacun d'eux ainsi que leur interaction. Un agent peut représenter un organisme, des ressources matérielles et immatérielles et toute entité capable d'atteindre un objectif (Abar et al., 2017; Roy et al., 2004). L'un des objectifs majeurs de cette simulation est d'évaluer comment la conduite d'un agent peut influencer celle des autres agents et celle du processus en entier. *La simulation analytique* permet d'évaluer la performance des processus en utilisant des méthodes analytiques qui se basent sur les équations des théories régissant l'évaluation de la performance d'un système comme « Stochastic Petri-net », « Stochastic Activity Networks ». Le choix des variables à modéliser et simuler se fait en référence et se limite à la méthode choisie

Chacune de ces techniques représentent des avantages et des inconvénients en fonction du domaine d'activité et l'objectif de l'analyse de la performance souhaité. (Borshchev and Filippov, 2004; Greasley, 2009) proposent des guides pour choisir au mieux le type de simulation convenable pour atteindre les objectifs souhaités. Bien que, dans le cadre des processus métiers, la simulation à événements discrets reste la plus utilisée.

4.3.Outils et langage de simulation

4.3.1. BPSIM

Business Process Simulation Interchange Standard (BPSim) est un standard de la WFMC qui permet de simuler les modèles des processus métiers. Il permet de supporter deux types de langage de modélisation : BPMN et XPD. La spécification de BPSim fournit le paramétrage des données possible pour pouvoir simuler et analyser un modèle du processus métier dans un but d'optimisation pré-exécution et post-exécution. La simulation pré-exécution permet de jouer des scénarios de simulation en utilisant des estimations de valeur comme paramètre d'entrée, alors que la simulation post-exécution permet d'utiliser les logs d'exécution des processus comme valeur d'entrée. Un scénario dans BPSIM permet de déterminer :

- Les paramètres d'entrée pour l'analyse, la simulation et l'optimisation
- Les résultats des analyses, des simulations et d'optimisations

- L'historique des données de l'exécution réelle d'un processus métier

BPSim permet l'analyse et la simulation des données liées au temps de traitement, flux de contrôle, coût et ressource.

Les principaux paramètres que BPSIM permet de spécifier aux ressources qui vont exécuter une activité sont :

- Disponibilité « availability » : Un paramètre sous forme de booléen permet de spécifier si la ressource est disponible ou non
- Quantité : Un paramètre qui permet d'indiquer la quantité des ressources
- Sélection : Un paramètre qui permet de sélectionner les ressources à affecter à une activité
- Rôle : Un paramètre qui permet d'indiquer le rôle de la ressource

4.3.2. Bonita Open Solution

Bonita Open Solution est une solution open source complète de gestion de processus fournit par BonitaSoft. Elle contient trois modules :

- Bonita Studio permet la construction des modèles BPMN 2.0 facilement et intuitivement. L'outil implémente les éléments les plus importants du langage et un GUI pour générer des interfaces graphiques automatiquement. Quant à la simulation, l'outil permet également de jouer différents scénarios et analyser les données liées au temps de traitement, flux de contrôle et les ressources
- Bonita Execution Engine est la partie qui se charge de l'exécution des processus modélisés à l'aide de BonitaSoft
- Bonita User Experience fournit un suivi de l'état des processus en cours et des processus finalisés

La simulation avec Bonita Open Solution permet d'évaluer l'exécution du processus avec les différents profils en charge et les différentes disponibilités des ressources. Elle fournit les résultats de la simulation pour chaque élément du processus (le temps d'exécution, le temps d'attente) et également ceux liés aux ressources (nombre de ressources consommés, le coût des ressources et leur taux d'utilisation).

La nouvelle version de Bonita open solution a remplacé l'engin de simulation des processus avec un nouveau modèle appelé Bonita Intelligent Continuous Improvement (BICI). Ce module utilise des algorithmes de process mining pour analyser les données archivées, détecter des tendances pour prédire les problèmes et les inefficiences.

4.3.3. jBPM

jBPM (Java Business Process Management) est un outil de conception et d'exécution des processus open source introduit par JBOSS. L'outil utilise le langage BPMN 2.0 pour la modélisation des processus. L'éditeur graphique de jBPM conçu dans un environnement web permet de modéliser facilement les processus. La simulation dans jBPM se base sur le méta-modèle de BPSIM 1.0 et permet d'afficher les résultats suivants :

- Résultat du processus : Temps d'exécution, nombre d'instances et le coût total
- Résultat des tâches d'utilisateurs : Temps d'exécution, l'utilisation des ressources et le coût des ressources

- Résultats des autres éléments du modèle : temps d'exécution
- Résultats des chemins des processus : Temps d'exécution du chemin

La simulation fournit une analyse statistique du processus durant une durée déterminée en permettant des exécutions multiples d'un même scénario tout en ayant la possibilité de modifier les valeurs de distribution. Il offre également une animation intuitive des résultats obtenus et de bons rapports statistiques.

4.3.4. ARIS Business Simulator

ARIS Business Simulator simulation est un outil de la suite ARIS (Architecture of Integrated Information Systems). La suite ARIS a été développée pour répondre aux besoins d'ingénierie de système d'information dans un premier temps puis dans un deuxième temps répondre aux besoins d'ingénierie d'entreprise. Les méthodes d'architecture et de modélisation proposées sont basées sur les concepts de généricité, de cycle de vie et de vues. Les langages de modélisation utilisés sont EPC et BPMN. ARIS Business Simulator est un outil d'aide à la réingénierie, l'optimisation et l'analyse des processus, l'analyse de différents scénarios et la planification des ressources. Cet outil permet d'évaluer si le processus est correctement exécutable, il permet de fournir son temps de traitement, son temps d'attente et son taux de production. Il permet également d'identifier les besoins en ressources, leur taux d'utilisation, le coût et les goulots d'étranglement d'un processus. En plus de modéliser et simuler les modèles de processus sous forme de flux de contrôle, ARIS prend en considération la modélisation et la simulation de l'aspect structurel et organisationnel d'un processus sous forme d'organigramme. Ainsi, dans le but d'avoir une vision claire au déploiement du personnel, la représentation des ressources humaines peut être décomposée à un niveau détaillé. L'outil contient une interface qui affiche des aperçus des modèles simulés, l'état de leur simulation et les résultats d'évaluations sous forme de tableaux et de diagrammes statistiques.

4.3.5. ARENA

ARENA est un outil de simulation produit par Rockwell Automation utilisant Siman comme langage intégré. Il s'agit d'un outil basé sur un concept orienté objet pour une modélisation complètement graphique avec la possibilité d'écrire du code en cas de nécessité. Ce modèle est créé en plaçant des blocs sur un espace de travail, puis en créant des liaisons entre ces différents blocs. Parmi les points forts d'ARENA, sa capacité à communiquer avec certains programmes externes ce qui permet d'utiliser des données externes en simulation. Cet outil de simulation offre une aide à démontrer, prédire et mesurer la performance du processus, tester plusieurs règles métiers et scénarios avec des conditions variables. Il offre la possibilité de simuler l'assignation, planification et allocation des ressources, le temps de traitement et le temps d'attente d'un processus en ayant la possibilité de choisir la loi de traitement et de modifier les files d'attente. Des animations graphiques 2D et 3D sont possibles

II. Extensions des langages semi-formels

Comme présenté dans les chapitres précédents, la modélisation et la simulation sont des étapes essentielles pour étudier et analyser les processus avant leur automatisation, car ils permettent de donner une vision plus claire des besoins et des anomalies à résoudre. Cependant, la simulation des processus métiers est peu utilisée dans le milieu industriel pour plusieurs raisons.

Parmi celles adressées par (Barth et al., 2012; Uhrmacher, 2012) on trouve : la complexité à trouver une sorte d'équilibre entre simplifier le modèle et l'enrichir avec les informations nécessaires et suffisantes à la simulation. Ajoutant à cela, la difficulté d'implémenter les composants de simulation qui est généralement utilisée par les analystes métiers qui n'ont pas des compétences techniques. (Bocciarelli et al., 2014) a résumé en trois points les principales raisons pour lesquelles les entreprises sont réticentes à l'utilisation de la simulation dans leur projet d'automatisation :

- Les approches de simulation proposées requièrent une maîtrise accrue des compétences mathématiques et/ou informatiques et sont destinées à des experts en simulation plutôt qu'aux analystes métiers. Donc l'entreprise se voit obligé d'embaucher un spécialiste de simulation ce qui est généralement considéré comme une perte d'argent
- On trouve également le problème d'écart entre le modèle conceptuel et le modèle de simulation, l'analyste métier doit modéliser le processus en utilisant un langage de modélisation compréhensible par tout le monde et puis le re-modéliser avec un autre langage de simulation et doit s'assurer de la compatibilité des deux modèles ce qui est considéré comme une perte de temps et d'argent
- Il existe des outils industriels comme présenté dans le paragraphe 4.3 du chapitre 4 consacré à la simulation. Ces outils proposent des solutions aux limitations mentionnées, toutefois la plupart de ces outils sont peu flexible et souvent limités à leur objectif de vente

De ce fait, les entreprises cherchent un outil flexible, avec un langage compréhensible par tout type d'intervenant, qui peut être utilisé à la fois pour la modélisation et la simulation tout en ayant une prise en main facile pour l'analyste métier (M.Law and Kelton, 2000). C'est la raison pour laquelle les chercheurs enrichissent les langages semi-formels qui sont les plus utilisés par le milieu industriel. Comme présenté dans le paragraphe 3.3 du chapitre 3 consacré à la modélisation. Ces langages de modélisation fournissent des éléments de description génériques pour représenter et vérifier la partie fonctionnelle du processus uniquement. Souvent, ces modèles ont besoin d'être enrichis pour décrire et vérifier des spécificités et caractéristiques particulières. L'enrichissement consiste à étendre la syntaxe et/ou la sémantique du langage en réutilisant sa spécification et son implémentation. Nous allons voir dans les prochains paragraphes les différentes extensions proposées dans ce domaine.

(Zarour et al., 2019) ont catégorisé l'extension des langages de modélisation en deux catégories : extension du langage pour représenter les processus d'un domaine d'activité spécifique (domain-specific) et l'extension pour enrichir le langage dans sa globalité. La première catégorie consiste à étendre le modèle en ajoutant de nouveaux éléments pour représenter les caractéristiques spécifiques d'un domaine particulier comme le domaine de la santé, la production, etc. La deuxième catégorie quant à elle consiste à enrichir le langage d'éléments généraux qui peuvent améliorer tout type de processus quel que soit son domaine d'activité. Il s'agit d'enrichir l'expression des besoins en améliorant l'expression de la flexibilité, de la variabilité, etc. Ou encore d'enrichir le langage avec des critères non-fonctionnels comme le coût, la sécurité, qualité, etc. Ainsi nous allons dans les prochains paragraphes représenter les différents travaux en suivant ce classement et en catégorisant la

partie d'enrichissement des langages dans leur globalité en deux parties : extension orientée activité et extension orientée ressource. Plusieurs extensions de langage de modélisation des processus métiers sont proposées dans la littérature ciblant plusieurs objectifs, l'extension du langage BPMN est la plus répandue. Ce chapitre présente, sous forme de tableaux, l'objectif et les méthodes des approches d'extensions des langages de modélisation de processus décrit dans le chapitre de « Modélisation des processus métiers » les plus référencées et les plus récentes.

1. Extension orientée domaine spécifique

L'extension des langages de modélisation pour les domaines spécifiques consiste à ajouter des concepts spécifiques à un domaine technique ou à un domaine métier au langage natif. L'objectif est d'apporter une solution ciblée et efficace aux problèmes liés à un domaine particulier.

<i>Chercheur</i>	<i>Langage</i>	<i>Extension ajoutée</i>	<i>Méthode</i>
<i>(Zhou and Rong, 2010)</i>	IDEF0	-Ajout des paramètres liés à la Supply Chain (approvisionnement, inventaire, etc.) -La possibilité de visualiser et d'évaluer les variations dynamiques d'une Supply Chain	-Construction d'un modèle par décomposition descendante (détailler la fonction principale graduellement)
<i>(Bastos and Ruiz, 2002)</i>	AD-UML	-Intégration des activités liées à la production suivant les concepts C-WF (Bastos and Ruiz, 2001)	-Utilisation des stéréotypes du langage UML pour l'extension -Modification du diagramme graphique en ajoutant de nouvelles propriétés
<i>(Spyrou et al., 2005)</i>	AD-UML	-Ajout des paramètres liés au traitement des documents cliniques	-Utilisation des stéréotypes du langage UML pour l'extension -Ajout de nouveaux éléments graphiques
<i>(Barbalho de Figueroa et al., 2020)</i>	AD-UML	-Intégration des concepts bio-informatique	-Utilisation des stéréotypes du langage UML pour l'extension -Ajout de nouveaux éléments graphiques
<i>(Radloff et al., 2015)</i>	EPC	-Intégration des moyens de contrôles et des spécificités des audits	-Ajout de nouvelles classes au méta-modèle d'EPC -Ajout de nouveaux éléments graphiques en enrichissant un modèle d'interchangeabilité du langage EPML XML (Mendling and Nüttgens, 2006)

<i>(Zor et al., 2011)</i>	BPMN 2.0	-Intégration des concepts et caractéristiques du domaine de production	-Ajout de nouveaux éléments graphiques (tâches de production, etc.) -Ajout de nouvelles passerelles pour les flux de matière
<i>(Braun et al., 2014b)</i>	BPMN 2.0	-Intégration des spécificités liées au parcours de soin d'un patient	-Modélisation des concepts du parcours de soin avec le diagramme de classe -Transformation du modèle en une extension du méta-modèle BPMN -Ajout nouveaux éléments graphiques
<i>(Polderdijk et al., 2018)</i>	BPMN 2.0	-Intégration des propriétés de risques physiques des opérateurs de production	-Modification des types de tâches natives de BPMN -Analyse des risques en intégrant la méthode Checklist Physical Load (TNO, 2012)

Tableau 1: Extension des langages semi-formels pour intégrer les caractéristiques de domaines spécifiques

2. Extension orientée activité, données et flux de contrôle

Cette partie se focalise sur les propositions qui enrichissent les éléments des langages de modélisation des processus métiers avec des propriétés fonctionnelles et non-fonctionnelles au niveau de la tâche, des données et flux de contrôle.

2.1. Extension des propriétés fonctionnelles

L'objectif de ce type d'extension est d'enrichir les langages de modélisation avec de nouveaux éléments permettant d'améliorer son expressivité opérationnelle

<i>Chercheur</i>	<i>Langage</i>	<i>Extension ajoutée</i>	<i>Méthode</i>
<i>(Bazoun, 2015)</i>	IDEF3	-Extended Actigram Star (EA*) : Extension de Grai Extended Actigram (Chen et al., 1997) qui est basée sur IDEF3 -Proposition d'un méta-modèle qui permet d'assurer l'interopérabilité du langage avec d'autres formalismes	-Enrichissement du méta-modèle en se basant sur celui de Grai Extended Actigram -Ajout de nouveaux éléments graphiques
<i>(Mendling et al., 2005)</i>	EPC	-Intégration des 20 patterns introduit par (W.M.P. van der Aalst et al., 2003)	-Ajout de nouvelles classes au méta-modèle EPC -Ajout de nouveaux éléments graphiques

(Amjad et al., 2018)	EPC	-Intégration de la notion d'évènements complexes	- L'ajout des concepts du diagramme d'activité UML au méta-modèle EPC -La vérification de la syntaxe du modèle se fait via la transformation des modèles en utilisant « Timed Computational Tree Logic » (Faugere et al., 2007)
(Narayan Debnath et al., 2003)	AD-UML	-Intégration les concepts et patterns d'un workflow selon le standard WFMC	-Ajout de nouvelles classes au méta-modèle d'UML
(Martinho et al., 2015)	BPMN 2.0	-Intégration des concepts de contrôle d'activité	-Création d'un méta-modèle des concepts liés au contrôle d'activité -Ajout de nouvelles classes au méta-modèle BPMN -Modification de certains éléments graphiques -Utilisation des annotations textuelles pour exprimer les expressions de contrôle
(BenSaid et al., 2016)	BPMN 2.0	-Intégration des concepts intra-organisationnels et inter-organisationnels	-Ajout de nouvelles classes au méta-modèle selon le mécanisme d'extension recommandé par la spécification de l'OMG -Ajout de nouveaux éléments graphiques -Intégration d'algorithmes spécifiques pour générer les versions de modèles

Tableau 2: Extension des propriétés fonctionnelles des langages semi-formelles

2.2.Extension des propriétés non-fonctionnelles

L'objectif de ce type d'extension est d'enrichir les langages de modélisation avec de nouveaux éléments permettant de représenter les propriétés non-fonctionnelles du processus à automatiser. De ce fait, on regroupe dans un seul langage les propriétés fonctionnelles et non-fonctionnelles.

<i>Chercheur</i>	<i>Langage</i>	<i>Extension ajoutée</i>	<i>Méthode</i>
<i>(Jeong et al., 2009)</i>	IDEF0/ IDF3	-Intégration des temps moyen de traitement et le nombre d'unités produites	-Utilisation d'une fonction pour collecter les données de simulation -Proposition d'une approche de simulation basée sur le développement du modèle basée sur les données
<i>(Korherr and List, 2007)</i>	EPC	-Intégration des objectifs à atteindre et des mesures de performance	-Ajout d'une nouvelle vue au méta-modèle d'EPC -Ajout de nouveaux éléments graphiques
<i>(Rosemann and Muehlen, 2005)</i>	EPC	-Intégration des concepts de risque d'exécution	-Identification des risques en utilisant les concepts de probabilité et l'ordre de grandeur des risques
<i>(Gu et al., 2016)</i>	AD-UML	-Intégration des concepts d'analyse temporelle	-Utilisation des descriptions textuelles pour l'ajout des concepts stochastiques -Utilisation d'une extension du langage Petri Net (Vanderperren et al., 2008) pour traduire le comportement stochastique -Transformation de cette traduction en un modèle NPTA (David et al., 2015) pour l'analyse quantitative
<i>(Daw and Cleaveland, 2016)</i>	AD-UML	-Intégration de 37 éléments pour évaluer le comportement du processus	-Utilisation d'UML profile pour l'ajout des nouveaux concepts -Implémentation de la solution dans un outil de simulation implémenté par les mêmes auteurs
<i>(Bartolini et al., 2012)</i>	BPMN 2.0	-Intégration des concepts de performance, de sécurité et de fiabilité	-Utilisation des annotations textuelles pour renseigner les propriétés non-fonctionnelles -Les annotations textuelles sont associées aux instances de classe UML fondée sur un profil UML -Ajouts de nouvelles classes de méta-modèle conformément au méta-modèle PMM (Di Marco et al., 2011) -Transformation du langage étendu en un modèle conforme à la

				notation de Klaper suite (Ciancone et al., 2011) pour l'analyse
(Bocciarelli et al., 2014)	BPMN 2.0	-Intégration des mesures performances et de fiabilité	des de et de	-Ajout de nouvelles classes au méta-modèle -Proposition du langage eBPMN pour générer automatiquement le code de simulation par transformation de modèle -Utilisation des annotations textuelles pour renseigner les données de simulation -Proposition d'une implémentation d'eBPMN en respectant la sémantique d'exécution de BPMN 2.0
(Arevalo et al., 2016)	BPMN 2.0	-Intégration des contraintes de temps	des	-Ajout de nouvelles classes au méta-modèle BPMN 2.0 -Intégration des contraintes OCL (OMG, 2014) par l'utilisation des techniques de l'ingénierie dirigée par les modèles
(Heguy, 2018a)	BPMN 2.0	-Intégration des concepts performance et de fiabilité des données	des de et de	-Ajout d'extension selon le mécanisme décrit par l'OMG -Ajout de nouveaux éléments graphiques -Utilisation de la méthode SWR (Cardoso et al., 2004) et l'agrégation des performances (Ducq * and Vallespir, 2005) pour l'analyse de la performance

Tableau 3: Extension des propriétés non- fonctionnelles des langages semi-formelles

3. Extension orientée ressource

Cette partie présente les approches d'extension des propriétés fonctionnelles et non-fonctionnelles des ressources des langages de modélisation des processus métiers.

<i>Chercheur</i>	<i>Langage</i>	<i>Extension ajoutée</i>	<i>Méthode</i>
(Reid et al., 2019)	IDEF	-Intégration des concepts de qualité de rendement des ressources d'un processus	-Utilisation des méthodes mathématiques pour calculer le rendement

<i>(Bazoun, 2015)</i>	IDEF3	-Distinction des représentations des ressources humaines, matérielles et IT	-Ajout de nouveaux éléments graphiques au méta-modèle conçu par les auteurs
<i>(Russell et al., 2005)</i>		-Distinction de 43 patterns liés à l'exécution des ressources dans un workflow	
<i>(Awad et al., 2009)</i>	BPMN 2.0	-Intégration des concepts d'allocation des ressources tel que défini dans (Russell et al., 2005)	-Ajout de nouvelles classes au méta-modèle -Utilisation d'OCL pour modéliser les contraintes d'allocation des ressources et les analyser -Ajout de nouvelles interfaces graphiques pour renseigner les descriptions d'allocation
<i>(Stroppi et al., 2015)</i>	BPMN 2.0	-Intégration des concepts de structure des ressources, leur droit d'assignation et leur répartition de travail	-Définition des méta-modèles des concepts à intégrer -Ajout de nouvelles classes au méta-modèle de BPMN selon les directives de l'OMG -Utilisation des techniques de l'ingénierie dirigée par les modèles pour l'analyse des contraintes ajoutées et la génération d'un code exécutable des spécifications des workflows -Utilisation des annotations textuelles pour ajouter les informations liées aux ressources
<i>(Vasilecas et al., 2014)</i>	BPMN 2.0	-Intégration des concepts de rôle et disponibilité des ressources, délai et coût	-Ajout de nouvelles classes au méta-modèle -Ajout de nouveaux éléments graphiques
<i>(Domingos et al., 2016)</i>	BPMN 2.0	-Intégration des paramètres de fiabilité des ressources	-Utilisation du langage XML du modèle BPMN pour renseigner les paramètres de fiabilité -Utilisation des probabilités et de la méthode SWR (Cardoso et al., 2004) pour le calcul de la fiabilité

<i>(Bocciarelli et al., 2016)</i>	BPMN 2.0	<ul style="list-style-type: none"> -Intégration des paramètres de performance et de fiabilité des ressources -Proposition d'un mécanisme de présentation de ressource composée -Proposition d'un mécanisme de présentation de ressource alternative 	<ul style="list-style-type: none"> -Ajout de nouvelles classes au méta-modèle -Simulation des propriétés non-fonctionnelles à l'aide des mécanismes de transformation de modèle -Utilisation des annotations textuelles pour renseigner les données de simulation
-----------------------------------	----------	--	--

Tableau 4: Extension des propriétés fonctionnelles et non-fonctionnelles des ressources des langages semi-formelles

D'autres approches ont traité le point des ressources sans étendre les langages de modélisation des processus métiers. (Pika et al., 2014) ont proposé un Framework qui permet d'extraire les informations liées au comportement d'une ressource humaine à partir des logs d'exécution en termes de compétence, d'utilisation, de productivité, etc. Il permet également d'analyser comment leurs performances ont changé à travers le temps. Toujours dans la même approche de « process mining », (Schionig, et al., 2016) ont proposé un Framework qui permet de capturer à partir des logs d'exécution l'implication des ressources, la composition des groupes de travail et comment le travail collaboratif est effectué. (Cabanillas et al., 2015) ont proposé une nouvelle notation permettant de modéliser l'attribution des ressources humaines aux tâches adéquates. Cette proposition est indépendante de tout langage de modélisation de processus. Ils ont également enrichi la notation proposée avec la sémantique du langage RAL ce qui permet de prédire les éventuels problèmes d'allocation des ressources.

III. Ingénierie dirigée par les modèles

1. MDE

1.1.Définition

Afin d'améliorer le développement des applications dans le domaine du génie logiciel, plusieurs approches ont été proposées dans la littérature. On peut en citer la programmation fonctionnelle, la programmation objet, les approches basées sur les composants (patterns), les approches basées sur les aspects, etc. Chaque approche a ses propres avantages et inconvénients selon l'objectif à atteindre. L'ingénierie dirigée par les modèles (IDM) est née du besoin de pouvoir regrouper et intégrer différentes approches et langages adaptés à des objectifs différents (Dolques, 2010)

IDM en anglais Model Driven Engineering (MDE) est une approche qui préconise de se focaliser sur les concepts du système en différenciant ses spécificités conceptuelles séparément de ses spécificités techniques. Elle est basée sur la notion de niveau d'abstraction de modèles et sur leur mécanisme de transformation. Le modèle est le pivot d'IDM, où il est considéré comme le noyau principal du processus de développement d'application et non uniquement un artefact de documentation et de communication. L'idée est d'utiliser les différents langages de modélisation pour décrire graduellement le besoin en partant d'une description abstraite à une

description concrète de la solution à implémenter (Figure 14). Ainsi, IDM propose une démarche itérative pilotée par les modèles où leur cohérence globale est fournie par les mécanismes de transformation.

Les trois piliers de l'IDM sont :

- Le principe de conception par des niveaux d'abstraction
- Le principe de séparation des concepts et des technologies
- Le principe de transformation des modèles

IDM est une approche de développement de logiciel appréciée par le monde industriel et académique du fait des améliorations significatives apportées, et doit sa notoriété aux travaux de l'OMG avec l'architecture dirigée par les modèles (MDA).

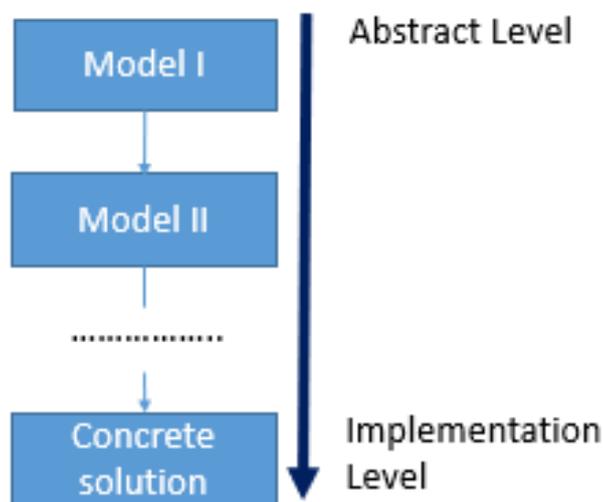


Figure 14: Description graduelle du besoin dans la démarche d'IDM

1.2.Composants d'IDM

Les principaux composants utilisés dans une démarche d'ingénierie dirigée par les modèles sont :

- **Model** : Comme décrit dans le chapitre de l'état de l'art, le modèle est une représentation schématique simplifiée du besoin et représente la base de la démarche IDM
- **Méta-modèle** : Dans le contexte de l'approche IDM, le méta-modèle est défini par « *A meta-model is a model that defines the language for expressing a model* » (OMG, 2002). Ainsi, le méta-modèle définit un langage qui spécifie un ensemble de concepts et leur relation pour construire un modèle. Les modèles écrits dans ce langage sont nommés « modèle conforme au méta-modèle ».
- **Transformation** (Barbier, 2013; Dolques, 2010): Une transformation de modèle permet de transformer un modèle source, conforme à un méta-modèle source, à un modèle cible

conforme à un méta-modèle cible. La transformation se fait par le biais des règles de mappage qui mettent en correspondance les éléments des deux modèles. Cette correspondance se porte sur les concepts partagés qui peuvent être mis en relation. Ainsi, le mécanisme de transformation comprend un ensemble de règles qui représentent des correspondances entre les éléments du modèle source et les éléments du modèle cible. Il existe plusieurs catégories de transformation : les transformations endogènes ou exogènes et les transformations horizontales ou verticales. Une transformation est endogène lorsque le modèle d'entrée (source) est le modèle de sortie (cible) sont conformes au même méta-modèle. Quant à la transformation exogène, chacun des modèles (entrée et de sortie) est conforme à son propre méta modèle. Une transformation est horizontale lorsque le modèle d'entrée et de sortie se trouve au même niveau d'abstraction. Une transformation est verticale lorsque le modèle d'entrée et de sortie se trouvent à des niveaux d'abstraction différents

2. MDA

2.1. Définition

L'approche d'architecture dirigée par les modèles, « Model Driven Architecture » en anglais (MDA) a été présentée publiquement par l'OMG en 2000 et mis à jour en 2003. Elle constitue une déclinaison de l'IDM en réunissant l'ensemble des standards proposés par l'OMG. Ainsi, MDA est une approche qui préconise de concevoir les applications en séparant la logique métier de l'architecture technique. Elle est motivée par le besoin de réduire les tâches de re-conception des applications en partant du principe que le modèle de conception est plus pérenne dans le temps qu'une technologie qui peut changer constamment et peut rapidement devenir obsolète (Elmounadi et al., 2013). De plus, les trois objectifs principaux de MDA sont la portabilité, l'interopérabilité et la réutilisabilité (OMG, 2003). A l'instar d'IDE, MDA se base sur la transformation des modèles métiers de haut niveau vers des modèles d'implémentation. Comme présenté dans la Figure 15, l'approche a défini trois niveaux de modélisation fixe (Combemale, 2008; Garredu et al., 2014) :

- **Computation Independent Model (CIM)** est un niveau indépendant de tout système d'information en représentant les concepts métiers indépendamment de leur implémentation
- **Platform Independent Model (PIM)** est un niveau qui décrit la logique métier du système indépendamment de toute plateforme technique ainsi qu'une représentation partielle d'une vue d'un CIM
- **Platform Specific Model (PSM)** est un niveau qui décrit l'implémentation d'une application sur une plateforme d'exécution

Le Code source représente le résultat final de MDA et est généré automatiquement de manière totale ou partielle

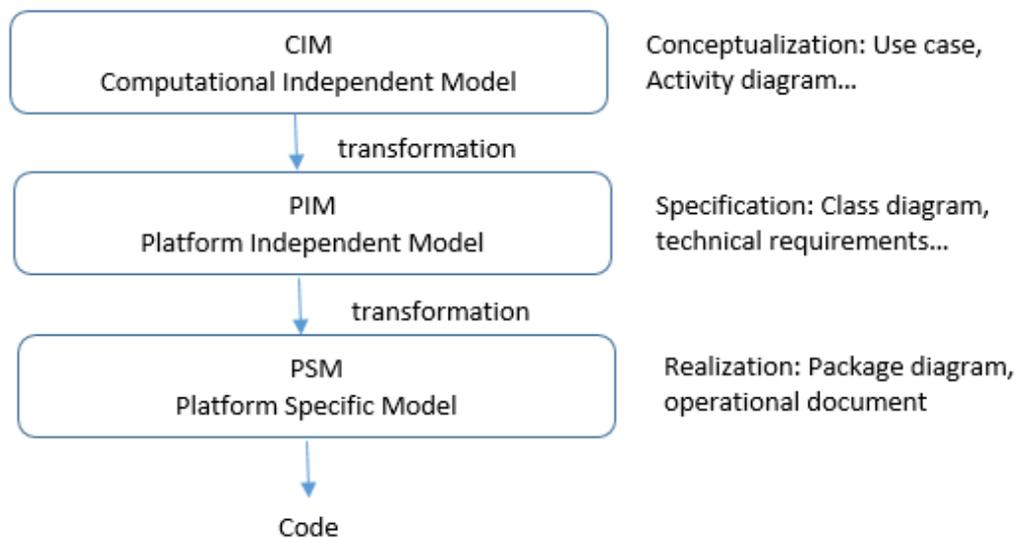


Figure 15: Les niveaux de modélisation de MDA

2.2. Transformations dans MDA

Comme le montre la Figure 16, plusieurs transformations sont possibles dans le cadre de l'approche MDA on trouve parmi elles (Hardebolle, 2008) :

- Transformations horizontales PIM vers PIM et PSM vers PSM. Ce type de transformation de modèle à modèle vise à enrichir et recadrer le modèle. La transformation est automatisable ou partiellement automatisable
- Transformations verticales PIM vers PSM. Ce type de transformation permet de paramétrer le PIM en fonction de la plateforme cible choisie. Cette transformation d'un modèle vers un autre modèle est réalisée en s'appuyant sur les informations fournies par le PDM
- Transformations verticales PSM vers l'implémentation (le code). Il s'agit d'une transformation de type modèle à texte. La transformation peut être entièrement ou partiellement automatisable mais généralement des complétions et modifications manuelles sont requises
- Transformations inverses implémentations vers PSM où PSM vers PIM sont des opérations de rétro-ingénierie (reverse engineering). Elles posent de nombreuses difficultés mais sont essentielles pour la réutilisation de l'existant dans le cadre de l'approche MDA

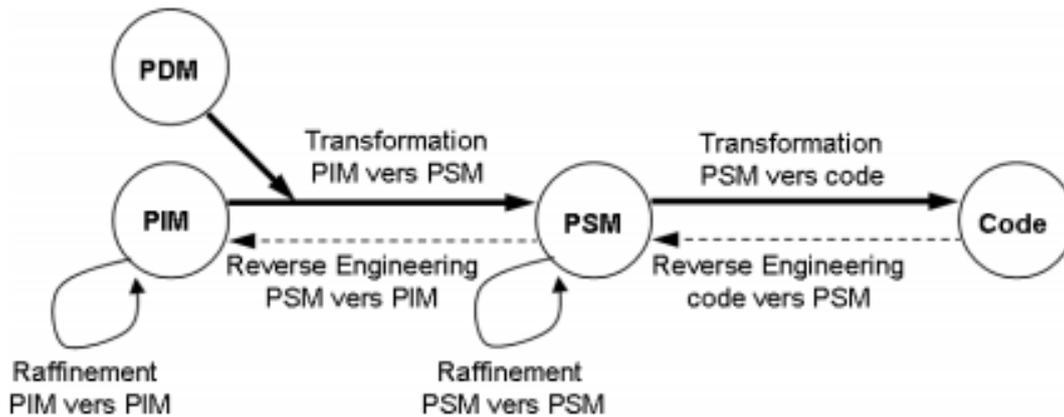


Figure 16: Transformation dans l'approche MDA (Hardebolle, 2008)

3. MDI

3.1.Définition

En comparaison avec les anciennes approches, l'approche MDA offre une meilleure méthodologie qui permet de mieux identifier et résoudre les problèmes d'interopérabilité (Bazoun, 2015). C'est l'une des raisons qui ont motivées (Bourey et al., 2007) à proposer l'approche d'interopérabilité dirigée par les modèles, « Model Driven Interoperability » (MDI) en anglais. L'approche MDI préconise d'intégrer les concepts d'interopérabilité dès le plus haut niveau de modélisation plutôt qu'au niveau technique uniquement. Son objectif est de réduire la complexité d'un système interopérable en la décomposant en plusieurs niveaux de modélisation. Le modèle de référence de l'approche MDI représente les différents types de modèle à utiliser aux différents niveaux d'abstraction et les transformations de modèles qu'il faut réaliser.

Elle enrichit l'approche MDA comme suit :

- Le niveau métier en décomposant le niveau CIM en deux sous-niveaux compte tenu de sa complexité. Top CIM Level se focalise sur la description globale de l'entreprise en prenant en compte le point de vue métier uniquement. Quant au niveau Bottom CIM Level, décrit les règles métiers d'un point de vue technique sans être pour autant dédiée à une technologie ou application spécifique
- Niveau logique (PIM Level) permet de décrire la solution d'architecture répondant au besoin décrit dans le modèle Bottom CIM. Cette description reste indépendante de toute solution technologique
- Niveau technique (PSM Level) permet de décrire l'implémentation de l'architecture représentée dans le niveau logique
- Niveau déploiement (Code Level) est le niveau de la réalisation technique répondant au besoin défini au niveau métier

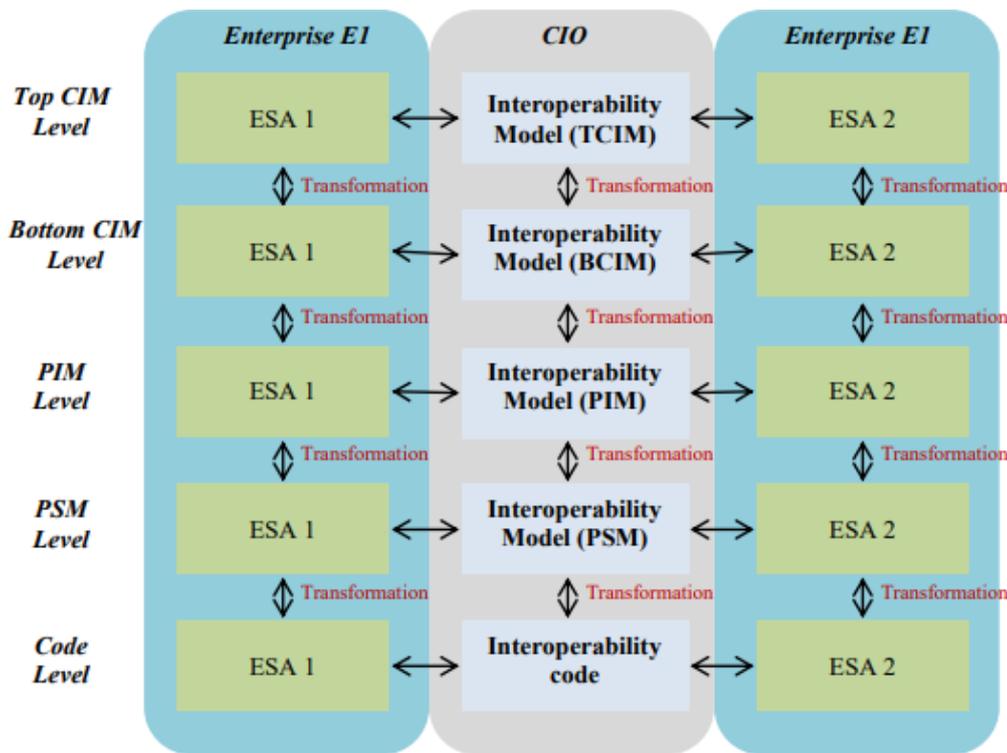


Figure 17:Modèle de référence de MDI

Comme le montre la Figure 17, le modèle d'interopérabilité est intégré à chaque niveau d'abstraction. Une transformation est effectuée d'un modèle d'interopérabilité à un autre en utilisant les annotations sémantiques comme les ontologies pour assurer une compréhension mutuelle entre les différentes entreprises. Dans ce modèle de référence, la transformation d'un modèle supérieur au modèle suivant est considérée comme une transformation verticale. Elle est principalement utilisée pour la génération de code. Quant à la transformation des modèles du même niveau, elle est considérée horizontale et elle est principalement utilisée pour l'échange d'information ou pour vérifier la cohérence entre les différentes entreprises.

4. MDSEA

Les travaux de (Bazoun, 2015) réalisés au sein de notre laboratoire consistent à proposer une méthodologie de transformation de modèle orientée service (MDSEA). Ces travaux ont été menés dans le cadre du projet MSEE, un projet européen de recherche et développement en collaboration avec 18 partenaires et 9 pays européens. L'objectif global de ce projet est de faire évoluer le concept de SSME (Service Science Management and Engineering) vers des systèmes de production et des usines de future.

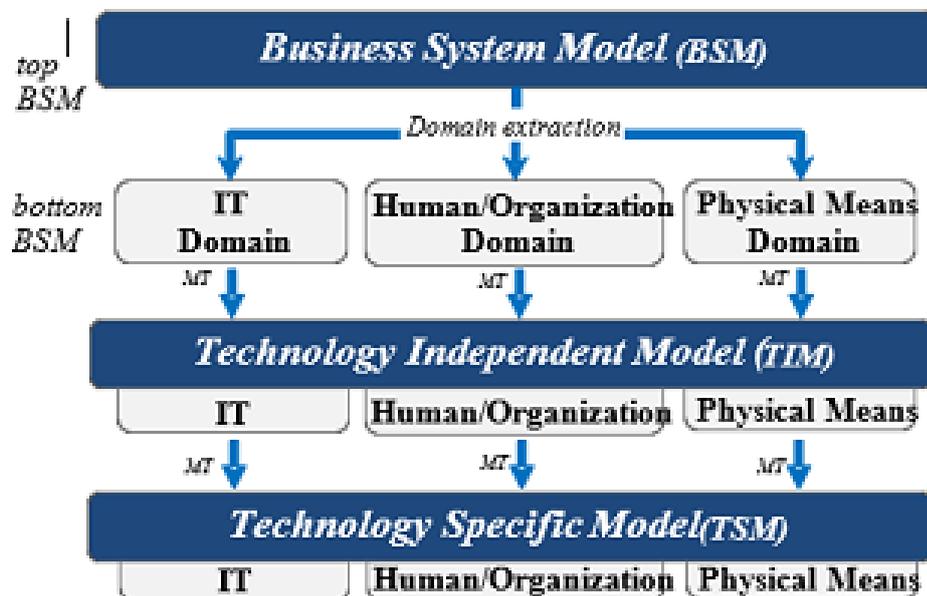
MDSEA « Model Driven System Enterprise Architecture » est une méthodologie basée sur les concepts des méthodes MDA/MDI et vise à pallier leur manque de modéliser et développer des systèmes dans un environnement virtuel. Dans le même esprit de l'ingénierie dirigée par les modèles, MDSEA est une méthodologie qui propose une modélisation sur trois niveaux d'abstraction pour concevoir et mettre en œuvre un système :

- Business Specific Model (BSM),
- Technology independent Model (TIM),
- Technological Specific Model (TSM).

L'un des apports majeurs de cette méthodologie est la définition des trois types de composant d'un système : Ressources humaines/Organisation, Matériel, Technologies de l'information (IT) ainsi que leur intégration au niveau le plus abstrait de modélisation. L'objectif est de s'assurer que les aspects des composants d'un système soient pris en compte en les développant graduellement dans les trois niveaux de modélisation.

Comme représenté dans la Figure 18, MDSEA commence avec le niveau BSM qui permet de spécifier les modèles conceptuels du système en décrivant ses différentes interactions (top BSM). Après une première analyse, ces services sont décrits en intégrant les trois types de ressource qui les composent (Bottom BSM). Le niveau TIM permet de décrire avec plus de détail les opérations des trois ressources sans inclure leur implémentation. Le niveau TSM permet de décrire un modèle détaillant les choix d'implémentation des ressources composés par le system et comment elles seront utilisées. Une explication détaillée des différents niveaux se trouve dans (Ducq 2014).

Model Driven System Engineering Architecture (MDSEA)



MT: Model Transformation

Figure 18: Méthodologie MDSEA

IV. Conclusion

1. Réponse aux questions de recherche

Q1.a Existent-ils d'autres techniques qui permettent de définir un modèle du processus métier applicable à différents domaines d'activité ?

Plusieurs langages de modélisation des processus existent dans la littérature, comme décrit dans le chapitre de l'état de l'art, ils sont soit informels, semi-formels ou formels. Chacun de ces langages possède des points forts et des points faibles selon plusieurs critères comme : la standardisation des notations utilisées, la formalité du langage, le degré d'expression possible pour modéliser les différents composants d'un processus (les acteurs, les données, etc.). Le choix d'utilisation d'un langage dépend du domaine d'application, l'aspect de modélisation prioritaire qu'on souhaite mettre en évidence et l'objectif de cette modélisation. Néanmoins, les langages de modélisation des processus métiers les plus répandues sont semi-formels surtout dans le monde industriel parce qu'ils sont faciles à prendre en main, même si les langages formels proposés sont utiles pour améliorer les processus.

Concernant la généricité des modèles, les langages semi-formels de la modélisation des processus métiers, présentés dans l'état de l'art sont destinés à être utilisés pour décrire les processus de tous les secteurs d'activités confondus. Néanmoins, les langages EPC, le diagramme d'activités d'UML et BPMN 2.0 sont les plus réputés pour avoir un large panel d'éléments pour la modélisation des processus, d'un point de vue informationnel, fonctionnel, comportemental et organisationnel (Mili et al., 2010).

Q1.b Ces techniques sont-elles compréhensibles par les utilisateurs de compétences métiers et techniques différentes ?

Concernant la compréhensibilité, parmi les langages présentés dans l'état de l'art, BPMN 2.0 est le langage de modélisation le plus répandu et utilisé pour la modélisation des processus (Entringer et al., 2019). Il est même considéré comme le de facto standard. D'ailleurs l'objectif premier de ces créateurs est de fournir une notation standardisée lisible et compréhensible par les différents utilisateurs de compétences différentes (du niveau métier au niveau technique).

Q2. Comment les différents types de ressource sont représentés d'un point de vue opérationnel ?

Au niveau des langages natifs, IDEF3 permet la représentation des ressources humaines et machines. Le diagramme d'activité d'UML permet la représentation des ressources humaines uniquement en revanche ses autres diagrammes permettent la représentation des ressources machines également. La modélisation des ressources dans BPMN est implicite, elle se fait via les couloirs ou à travers les différents types de tâche proposés (humaine, script, etc.). Le langage EPC quant à lui n'offre pas des éléments pour modéliser les ressources.

Au niveau des extensions des langages, comme l'indique le Tableau 4, les propositions trouvées sont quasiment orientées vers l'enrichissement du langage BPMN. En termes de modélisation fonctionnelle, les chercheurs ont proposé principalement des mécanismes de description des interactions à un bas niveau de modélisation comme la délégation des tâches entre les utilisateurs, l'allocation des ressources, etc. Ces propositions améliorent principalement la description des comportements des ressources humaines uniquement. A notre connaissance, le

langage EA* est la seule proposition qui offre des éléments de modélisation de trois types de ressource : humaine, matérielle et IT mais il est destiné pour une modélisation de haut niveau uniquement.

Q3.a Est-ce que la représentation et modélisation des propriétés fonctionnelles uniquement dans la phase d'analyse est suffisante pour étudier les performances du système à concevoir ?

De l'état de l'art effectué, nous constatons que la performance du système d'information est liée à l'efficacité fonctionnelle des processus métiers implémentés (ce que le processus doit faire), mais également à leur capacité à répondre aux besoins non-fonctionnels (la manière de le faire). De notre expérience chez Exakis Nelite, on constate également que le choix des techniques d'implémentations, l'environnement de production, etc., sont des éléments cruciaux à l'atteinte des objectifs de performance souhaités. De ce fait, les propriétés fonctionnelles et non fonctionnelles doivent être analysées à la phase de conception et non d'implémentation (Smith et al., 2002).

Q3.b Comment prendre en charge les besoins fonctionnels et non-fonctionnels à la fois ?

Comme indiqué ci-dessus, les langages semi-formels sont les plus utilisés par le secteur industriel. C'est l'une des raisons qui a poussé les chercheurs à étendre ces langages pour les enrichir avec les propriétés manquantes. Comme indiqué dans les tableaux du chapitre 2 : Tableau 1, Tableau 2, Tableau 3 et Tableau 4, il peut s'agir d'une extension pour améliorer le degré d'expressivité du langage, pour ajouter les composants d'un domaine spécifique ou pour ajouter les propriétés non-fonctionnelles. Ainsi, les propriétés non-fonctionnelles sont prises en compte en phase de conception avec les propriétés fonctionnelles.

De manière générale, de notre analyse effectuée, toutes les propositions évoquées dans l'état de l'art sont catégorisées en trois groupes :

- Groupe de chercheur qui propose une extension en ajoutant de nouveaux éléments graphiques tout en respectant le mécanisme d'extension du langage s'il est fourni. Ainsi, ce groupe respecte les avantages des langages surtout des langages standardisés. Cependant, l'inconvénient de l'ajout de nouveaux éléments graphiques est qu'ils ne seront pas pris en compte en cas de changement d'outil de modélisation
- Groupe de chercheur qui propose une extension en modifiant le méta-modèle original, la sémantique des notations originale et en ajoutant de nouveaux éléments graphiques. Ce type de proposition altère les avantages établis par les langages, surtout les langages standardisés. On peut carrément considérer ces propositions comme un nouveau langage proposé et non une simple extension du langage original
- Groupe de chercheur qui propose une extension en respectant le méta-modèle original ou le mécanisme d'extension proposé par le langage. Ils utilisent les annotations textuelles pour renseigner les propriétés non-fonctionnelles. L'avantage de cette approche est la conservation des spécificités des langages et l'interchangeabilité des modèles sans perte de données

En ce qui concerne les ressources, bien que la performance et la fiabilité du workflow dépendent de la performance et la fiabilité de ses ressources, au niveau de la littérature, peu de travaux

traitent l'extension des langages avec les propriétés non fonctionnelles des ressources du workflow. Les approches proposées peuvent être catégorisées en deux groupes :

- Groupe de chercheur qui propose une extension du langage afin d'enrichir son expressivité en termes de comportement technique de la ressource (allocation, distribution, etc.). L'objectif est de fournir une aide de décision pour choisir le WFMS le plus adéquats aux besoins de l'entreprise
- Groupe de chercheur qui propose une extension du langage pour renseigner les propriétés non-fonctionnelles des ressources plutôt que ceux des flux de contrôle. Ce type de proposition reste encore assez rare

A notre connaissance, aucune proposition d'extension de langage ne prend en considération les propriétés non-fonctionnelles des interfaces graphiques qui jouent un rôle important dans l'automatisation des processus. Les travaux proposés sont généralement liés à l'ergonomie de l'interface graphique comme la proposition d'approche de développement d'interface, la proposition d'outil pour les tester, des propositions de démarche de leur évaluation, etc. De plus, dans le cas où la simulation est implémentée, l'affichage de ses résultats se fait sans indiquer le chemin parcouru.

Nous avons également constaté que l'extension des langages standardisés est plus simple du fait de l'uniformisation du méta-modèle. IDEF3 est un langage standardisé mais ne fournit pas un mécanisme d'extension, ainsi chaque chercheur propose son propre dispositif d'extension du langage. UML est un langage standardisé par l'OMG et qui fournit un ensemble de méta-modèle pour ces différents diagrammes, tout en les reliant. Il propose également un mécanisme d'extension des méta-modèles afin de permettre aux utilisateurs de personnaliser et étendre le langage en fonction de leurs besoins particuliers. L'extension se fait à travers les stéréotypes, les « tagged value » et les contraintes. EPC n'est pas un langage standardisé, les mécanismes d'extensions ne sont pas clairement identifiés. De ce fait chaque utilisateur souhaitant étendre le langage, propose son propre méta-modèle et son propre mécanisme d'extension. BPMN 2.0 a un méta-modèle standardisé avec un dispositif sous format d'XML pour l'interchangeabilité des modèles. Il fournit également un mécanisme d'extensibilité qui permet l'intégration de nouveaux concepts tout en garantissant la conformité de ses éléments.

Q4. Dans le cas d'analyse d'un processus, le modèle et la représentation statiques sont-ils suffisants ?

Comme énoncé dans la question Q2.a, la prise en compte des propriétés non-fonctionnelles du processus à automatiser pendant la phase de conception est essentielle à l'étude de sa performance. De ce fait, une modélisation productive plutôt que contemplative est recommandée pour automatiser le plus possible leur analyse et ainsi faciliter leur étude. Dans ce sens, on peut catégoriser l'approche productive adoptée par les propositions d'extension de langage en trois catégories :

- Utilisation de la simulation pour ajouter un composant dynamique au modèle identique à une instance d'exécution. Le but est de mettre en exergue les différentes contraintes fonctionnelles et non-fonctionnelles et de permettre également la comparaison des différents scénarios possibles avant la mise en production. Le type de simulation le plus utilisé et la simulation à événements discrets

- Utilisation des algorithmes, il s'agit de développer des algorithmes pour évaluer la performance du processus à automatiser comme celui proposé par SWR, puis les associés au modèle à évaluer. On trouve des approches qui associent leur algorithme à la simulation
- Utilisation des méthodes analytiques. Les méthodes analytiques permettent d'évaluer la performance d'un processus en se basent sur les équations des théories choisies. Elles peuvent être associées à la simulation ou être implémenter sous forme d'algorithme

Nous constatons que les approches, qui ont pour objectif de s'aligner le plus possible à l'implémentation de la solution, utilisent la simulation à évènements discrets. Les solutions analytiques donnent suffisamment d'information sur la performance du processus si les équations analytiques sont simples à implémenter et si la méthode utilisée couvre la majorité des cas d'utilisations car tous les modèles ne peuvent pas être résolus de manière analytique.

Q5. Comment lier les différents résultats des différents points de vue d'analyse ?

La description d'un processus est effectuée par les utilisateurs métiers. Alors que la description de son implémentation technique (workflow) est décrite par les développeurs. Comme énoncé au niveau de la problématique, le problème qui peut surgir est l'écart entre les attendus des utilisateurs métiers et de la solution implémentée.

Comme présenté dans l'état de l'art, concernant les outils et techniques proposés par les chercheurs pour pallier cette problématique, les premières propositions étaient (et le sont toujours) d'améliorer le domaine de modélisation des processus en proposant de nouveaux langages et de nouveaux outils. D'autres chercheurs ont démontré l'utilité et l'intérêt de prendre en compte les propriétés non-fonctionnelles dès les premières phases d'analyse des besoins en utilisant la simulation pour aider les différents utilisateurs à réduire l'écart entre la phase de conception et la phase d'implémentation. Par conséquent, plusieurs outils commerciaux ont été proposés pour modéliser et simuler en même temps les processus à automatiser. L'inconvénient de ces outils est leur manque de flexibilité et leur limitation à leur objectif de vente. De ce fait des chercheurs ont proposé des extensions des langages les plus utilisés pour modéliser et simuler en même temps les processus métiers. Les extensions proposées sont axées sur les activités, les flux de contrôle et les flux de données alors que moins d'attention est portée sur les ressources.

Concernant les méthodologies qui permettent de pallier cette problématique, les approches d'ingénierie des modèles sont les plus couramment utilisées dans la littérature scientifique. Les modèles y sont considérés comme le socle du processus d'automatisation où le besoin est décrit graduellement, en séparant la partie conception de la partie technique. L'approche MDSEA contrairement aux autres approches n'est pas orientée vers l'implémentation de l'application logiciel uniquement mais plutôt sur l'implémentation de tous les types de ressource (humaine, matérielles et IT). C'est la raison pour laquelle elle recommande de les intégrer dès la première phase de modélisation.

2. Synthèse

Pour synthétiser, les processus automatisés fonctionnent toujours dans le cadre d'un système, généralement composé de ressources humaines, matérielles et IT. Par conséquent, la performance du workflow dépend de l'ensemble des performances des ressources qui lui sont liées. D'après la présentation de l'état de l'art et d'après les réponses aux questions posées, nous constatons que peu d'approche prennent en considération la modélisation des différents types de ressource d'un point de vue fonctionnel et non-fonctionnel. Tandis que les ressources sont le socle du processus pour produire des biens ou des services. De plus, chaque type de ressource a ses propres caractéristiques qui influent sur la performance du processus. Par exemple, les caractéristiques à prendre en considération pour calculer le temps de traitement d'une tâche par une ressource humaine ne sont pas identiques à celles d'une ressource matérielle. Le peu de travaux qui traitent ce point de vue, le font soit à un haut niveau de modélisation, soit au niveau de la conception. On ne trouve pas une méthodologie d'automatisation des processus orientée ressource, qui :

- Regroupe les différentes techniques énoncées dans l'état de l'art de la phase du recueil du besoin jusqu'à la phase des choix d'implémentation
- Prend en considération les différentes caractéristiques de performance de chaque type de ressource.
- Agrège les performances des différents types de ressource

De plus, à notre connaissance aucune approche ou technique ne prend en considération l'analyse des propriétés non-fonctionnelles des interfaces graphiques utilisées dans l'automatisation des processus.

Pour l'ensemble des points de questionnements non résolus par l'état de l'art et rappelés ici en synthèse nous proposons des contributions conceptuelles puis opérationnelles dans les chapitres suivants.

Chapitre III. CONTRIBUTION CONCEPTUELLE

I. Méthodologie proposée	48
1. Choix des langages de modélisation et simulation	48
2. Méthodologie	52
II. Distinction des ressources	54
1. Grille des performances des ressources	55
2. Ressources humaines	57
3. Ressources matérielles	59
4. Ressources IT	63
III. Agrégation des performances des ressources	64
1. Méthodes utilisées	64
2. Application sur l'agrégation des performances des ressources	68
IV. Interface graphique	72
V. Analyse des propriétés non-fonctionnelles en se basant sur un chemin	75
VI. Conclusion	78

Dans ce chapitre nous allons prendre en considération les limitations identifiées dans le chapitre de l'état de l'art. Nous allons proposer une méthodologie d'automatisation des processus par caractérisation des ressources du recueil du besoin jusqu'à la phase des choix d'implémentation. Notre approche est basée sur les principes de l'approche MDSEA qui intègre dans sa démarche tous les types de ressources d'un processus : humain, matériel et IT. Nous allons également présenter nos choix de langages de modélisation et de simulation utilisés. Ces langages permettent de représenter à la fois les propriétés fonctionnelles et non-fonctionnelles pour pouvoir les évaluer. Par la suite nous allons définir les méthodes utilisées pour agréger les performances des ressources et introduire notre proposition sur la simulation des propriétés non-fonctionnelles des interfaces graphiques. Pour finir, nous allons définir notre proposition d'analyse basée sur un chemin du processus.

I. Méthodologie proposée

1. Choix des langages de modélisation et simulation

1.1. Langage de modélisation

Comme nous l’avons présenté dans l’état de l’art, plusieurs langages de modélisation des processus existent dans la littérature. Puisque notre objectif est de fournir une méthodologie compréhensible par toutes les parties prenantes et qui couvre tous les secteurs d’activités le langage de modélisation à choisir doit répondre aux critères suivants :

- Haut niveau : Avoir la possibilité de modéliser graduellement le processus
- Fonctionnalité générique : Avoir la possibilité de modéliser tout type de processus
- Représentation des ressources
- Modèle standardisé

Langage	Haut niveau	Fonctionnalité générique	Représentation des ressources	Modèle standardisé
BPMN 2.0	X	X	-	X
EPC (Aris)	X	X	-	-
UML AD	X	X	-	X
IDEF 3	X	X	X	-

Tableau 5: Comparaison des langages de modélisation des processus

Comme le montre le Tableau 5, les langages BPMN 2.0 et UML AD sont les langages qui répondent à nos critères. Cependant, nous avons choisi le langage BPMN 2.0 car il est considéré comme le de facto standard pour la modélisation des processus. En outre, pour mieux exploiter la représentation des ressources avec UML, il faut utiliser ses différents diagrammes (diagrammes d’activités, diagramme de classe, etc.), or, nous souhaitons regrouper les différentes informations en un seul modèle.

Les raisons de notre choix se résument ainsi :

- Il fournit une notation graphique standardisée et largement compréhensible par les différents utilisateurs impliqués dans le processus de développement du workflow qu’ils soient fonctionnels ou techniques
- Il est à la fois facile à prendre en main et suffisamment riche pour modéliser les différents aspects d’un processus métier
- Il garantit la portabilité du modèle, ce qui rend possible l’importation du modèle d’un outil à un autre
- Il est associé à un ensemble de langages d’exécution comme « Business Process Execution Language » (BPEL)
- Il fournit un mécanisme d’extension qui permet de l’enrichir sans affecter ses spécificités

Les points abordés sont importants dans le contexte d’une SSII qui travaille avec plusieurs clients, de secteurs différents et avec des outils différents.

1.2.Choix de l’outil de simulation

Comme évoqué dans l’analyse de l’état de l’art, la prise en considération des propriétés non-fonctionnelles pendant la phase de conception est très importante pour anticiper la performance du workflow à développer. En outre, dans le cadre d’une SSII, il est important de garantir la compatibilité du modèle BPMN 2.0 pour que les clients puissent l’importer dans leur outil existant. Il est également important de garder les informations des propriétés non-fonctionnelles renseignées. De ce fait, l’outil à choisir doit faire partie de la catégorie 3 des propositions d’extension comme évoqué dans le chapitre de conclusion de l’état de l’art.

Avant de choisir un outil de simulation adéquat à notre proposition de méthodologie par caractérisation des ressources, nous avons analysé comment les ressources sont prises en compte dans BPMN 2.0.

Le méta-modèle de BPMN fournit des éléments pour décrire l’allocation des ressources pour traiter les activités d’un processus (Figure 19). Nous allons décrire de manière non exhaustive les éléments de ce diagramme : La classe « Resource » représente une ressource humaine ou matérielle qui est liée à une activité. Plusieurs activités peuvent utiliser la même ressource. L’affectation des ressources se fait via les éléments de la classe « ResourceRole ». La classe « Performer » représente la ressource responsable de l’exécution de l’activité, elle peut être un individu, un groupe, une organisation, etc. La spécification et le paramétrage liés aux ressources et leur affectation se font à l’aide des éléments des classes « ResourceAssignmentExpression » ou « ResourceParameterBinding ». Les éléments fournis par le langage sont utiles pour assurer l’exécution d’un processus. La modélisation des ressources quant à elle est hors de portée de la norme comme le souligne clairement l’OMG dans sa spécification. De ce fait, les partitions « Pools » et « Lanes » sont souvent utilisées dans le diagramme BPMN pour représenter visuellement les ressources en regroupant les activités d’une même ressource dans une seule « Lane ».

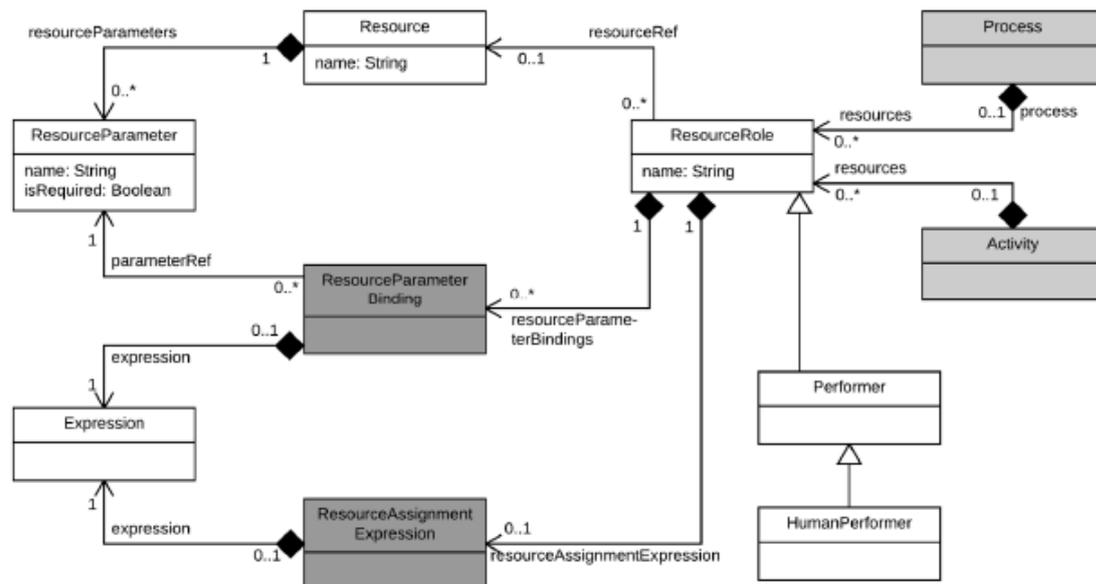


Figure 19: Diagramme de classe de la représentation des ressources dans BPMN (Domingos et al., 2016)

Une autre technique pour représenter les ressources visuellement dans BPMN est d'utiliser les différents types de tâches que le langage propose. Une activité ou une tâche est un élément de BPMN qui représente une action effectuée par une ressource du processus. L'accomplissement successif des tâches permet d'atteindre l'objectif final du processus. Dans BPMN, plusieurs types de tâche sont proposés. Les plus utilisés sont : Tâche abstraite, tâche humaine, tâche d'utilisateur, tâche de service et tâche de script. Chaque tâche représente un type de ressource particulier qui la traite. Le Tableau 6 reprend la description de chaque type.

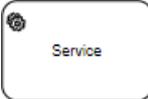
Tâche	Description
 <p>Tâche Abstraite</p>	Une tâche sans type de comportement spécifique. Elle est souvent utilisée lors d'une modélisation à haut niveau
 <p>Tâche Service</p>	Une tâche de service est une tâche automatisée qui représente l'utilisation d'un service externe à l'environnement d'exécution du processus (Web service, base de données...)
 <p>Tâche Script</p>	Une tâche de script est une tâche automatisée sans intervention humaine qui permet d'intégrer du code interprétable par l'application utilisée dans le processus
 <p>Tâche Manuelle</p>	Une tâche manuelle est une tâche effectuée exclusivement par une ressource humaine sans interaction avec une application informatique.
 <p>Tâche d'Utilisateur</p>	Une tâche utilisateur est une tâche effectuée par une ressource humaine à l'aide d'une application informatique (BPM, ECM, etc.)

Tableau 6: Description des tâches du langage BPMN

De ce fait, plusieurs types de tâche sont proposés par le langage BPMN 2.0 pour différencier visuellement les spécificités de chaque type. D'autre part une seule tâche d'utilisateur ou manuelle peut être exécutées par plusieurs ressources. Il s'agit alors d'une tâche exécutée par une ressource composée. Par conséquent, la performance de ce type de tâche est composée de la performance de toutes les ressources qui la traitent. Ainsi, quand ce cas se présente, il est important d'indiquer que la tâche est effectuée par une ressource composée et de distinguer chacune d'elle avec ses propriétés non-fonctionnelles pendant la phase de simulation. Dans le paragraphe suivant, nous allons décrire le choix de l'outil de simulation utilisé dans la méthodologie que nous proposons.

L'outil de simulation à choisir doit répondre aux critères suivants :

- Le langage de simulation utilisé : l'outil doit être compatible avec BPMN 2.0 et sa propre notation et la portabilité du modèle doivent être garanties

- La licence du produit : nous avons besoin d'un outil open source pour pouvoir ajouter les aspects manquants
- La prise en considération de la simulation des ressources
- La prise en considération de la simulation des ressources composées

La première sélection s'est faite sur les outils industriels les plus cités dans les travaux de recherche et les outils proposés par des chercheurs qui se rapprochent de notre besoin. Le Tableau 7 synthétise l'évaluation de ces outils en fonction des critères choisis.

Outil	BPMN	Open Source	Ressource	Ressource composée
BPSIM	X	X	X	-
Bonita	-	X	X	-
JBPM	-	X	X	-
ARIS	-	-	X	-
ARENA	-	-	X	-
(Vasilecas et al., 2014)	-	-	-	-
(Domingos et al., 2016)	X	X	X	-
(Bocciarelli et al., 2016)	X	X	X	X

Tableau 7: Comparaison des outils de simulation industriels

Comme le montre le Tableau 7 les outils ARIS et ARENA ne sont pas open source et ne sont pas compatibles avec les spécificités de BPMN 2.0. Les outils JBPM et BONITA n'implémentent pas tous les éléments du langage BPMN 2.0 suivant les recommandations de l'OMG (Arnaud, 2014; Geiger et al., 2015). Pour finir, BPSIM ne prend pas en considération le fait qu'une tâche peut être effectuée par une ressource composée. De ce fait, nous avons comparé les extensions proposées des propriétés non-fonctionnelles des ressources du langage BPMN 2.0 avec BPSIM. Nous avons constaté que l'outil eBPMN (Bocciarelli et al., 2016) répond à tous nos critères présentés ci-dessus et à notre connaissance, il est le seul outil à prendre en considération la composition d'une ressource. En outre, comme le montre la thèse de (Paglia, 2017), le langage proposé contourne également les limitations de BPSIM.

Les fonctionnalités générales de l'outil eBPMN se résument comme suit :

- La possibilité de modéliser et simuler à la même étape en utilisant l'ingénierie dirigée par les modèles pour générer automatiquement le code de simulation à partir du modèle BPMN 2.0,
- La garantie de la compatibilité du modèle BPMN 2.0 en proposant une extension qui garde le corps essentiel de son méta-modèle et en implémentant le langage eBPMN,
- L'implémentation du langage eBPMN se fait en respectant la sémantique du langage BPMN 2.0 et en utilisant le concept du token,
- La possibilité de simulation de la performance et la fiabilité des ressources qui exécutent la tâche. Ces informations sont renseignées dans les annotations textuelles,
- La prise en considération qu'une ressource peut être composée,

- La gestion des paramètres non-prévisibles en utilisant une distribution exponentielle et en générant des nombres pseudo-aléatoires

2. Méthodologie

La méthodologie que nous proposons est fondée sur l'approche MDSEA avec la combinaison des langages de modélisation BPMN 2.0 et de simulation eBPMN comme le montre la Figure 20. Le F représente les besoins fonctionnels et le NF les besoins non-fonctionnels.

Cette approche peut être adoptée dans le cas d'optimisation des processus avant leur automatisation par les clients finaux ou les SSII qui traitent les problématiques des clients de secteurs différents et avec des objectifs différents. L'analyste métier ou le consultant MOA représentent les pilotes de cette approche car ils sont les premiers responsables du recueil et de l'analyse des besoins auprès des parties prenantes du processus à automatiser. La Figure 21 représente le modèle BPMN de notre approche.

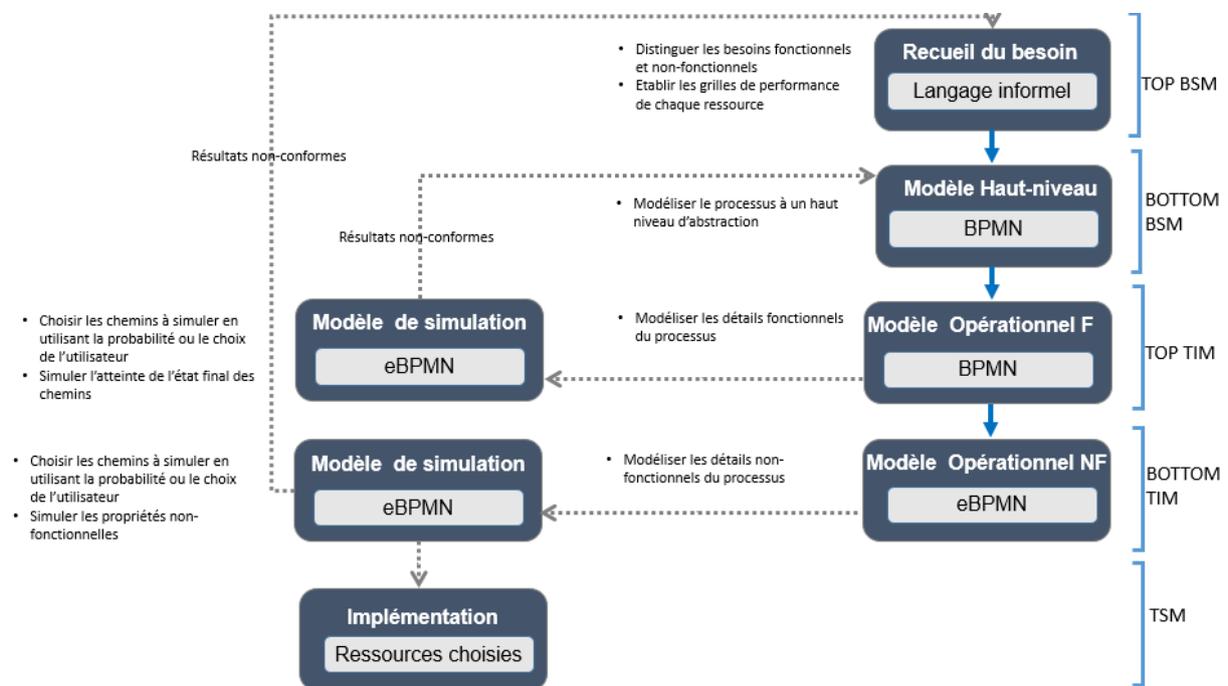


Figure 20: Représentation générale de l'approche d'automatisation des processus par caractérisation des ressources

Chaque étape de la méthodologie est décrite comme suit :

- **TOP BSM :** L'objectif de cette étape est de clarifier conceptuellement les exigences fonctionnelles et non-fonctionnelles du cahier de charge pour fournir une meilleure compréhension des objectifs à atteindre. Ensuite, vient la phase d'identification des besoins en ressource, tout type inclus, en utilisant la grille d'évaluation de leur caractéristique de performance que nous présentons dans la partie 2 de ce chapitre. Cette grille nous permettra dans les phases suivantes d'évaluer l'alignement des performances des ressources aux exigences non-fonctionnelles à atteindre. Puisque la collecte d'informations et l'aboutissement à un accord sont des tâches collaboratives, l'analyste métier, les utilisateurs métiers et les analystes IT impliqués dans le processus à automatiser sont inclus dans cette étape.

- **BOTTOM BSM** : L'objectif de cette étape est de modéliser de manière plus formelle le processus à automatiser à un haut niveau d'abstraction en utilisant le langage BPMN 2.0. Pour cette étape, la distinction des ressources est faite en utilisant uniquement les différents types de tâche proposés par BPMN (tâche manuelle, tâche utilisateur, tâche de service, etc.). La présence de l'analyste IT n'est pas requise pour cette étape puisque les solutions d'implémentation techniques ne sont pas discutées.

Pour résumer, les résultats de la phase BOTTOM BSM sont :

- Une identification claire des objectifs fonctionnels à atteindre
- Une identification claire des objectifs non-fonctionnels à atteindre
- Une identification de tous les types de ressource impliqués
- Des grilles d'évaluation des caractéristiques de chaque ressource impliquée remplies
- Un modèle BPMN à un haut niveau d'abstraction du processus

Nous avons décomposé l'étape « TIM » de l'approche MDSEA en « TOP TIM » et « BOTTOM TIM » pour clarifier les étapes d'évaluation des propriétés fonctionnelles et non-fonctionnelles.

- **TOP TIM** : L'objectif de cette étape consiste à modéliser en détail les fonctionnalités opératoires du processus à l'aide du langage BPMN. La description des détails opérationnels n'inclut pas celle des solutions d'implémentation. Ensuite, vient l'étape de la vérification des chemins en choisissant explicitement ceux à évaluer ou en assignant à chacun d'eux la probabilité de 50% pour les évaluer tous. Ce mécanisme d'analyse par chemin est décrit en détail dans la partie 5 de ce chapitre. Si tous les chemins à évaluer atteignent la fin du processus et sont conformes aux exigences fonctionnelles, la prochaine étape peut démarrer. Comme pour l'étape « BOTTOM BSM », la présence d'un analyste informatique n'est pas requise. Il est important de vérifier que tous les chemins à étudier atteignent leur fin avant de démarrer l'étape de l'analyse non-fonctionnelle
- **BOTTOM TIM** : L'objectif de cette étape est d'étudier et analyser les propriétés non-fonctionnelles pour chaque type de ressource. Nous utiliserons la grille d'évaluation remplie durant la phase « TOP BSM » que nous expliquons dans la partie 2 de ce chapitre. Les ressources sont assignées aux tâches adéquates avec leur mesure des propriétés non-fonctionnelles à l'aide du langage eBPMN. Ensuite, la simulation est lancée pour évaluer l'efficacité des mesures non-fonctionnelles des ressources pour le chemin choisi. Si les résultats sont conformes aux objectifs à atteindre, la phase d'implémentation peut démarrer, dans le cas contraire, la simulation est jouée après chaque changement apporté jusqu'à l'atteinte des objectifs de performance désirés. Il est important de revenir à la phase TOP TIM en cas de changement structurel dans le modèle du processus pour s'assurer que le chemin à analyser est atteignable. Pour cette étape, il est recommandé que toutes les parties prenantes soient présentes, surtout quand les différents scénarios d'implémentation sont joués.

A l'issue de cette phase, les résultats suivants sont obtenus :

- Un modèle BPMN détaillé des fonctionnalités du processus
 - Une validation des chemins à analyser au niveau fonctionnel et au niveau de leur « atteignabilité », c'est-à-dire, s'assurer que les différents chemins du modèle arrivent à l'état final du processus
 - Des résultats de simulation conformes aux objectifs désirés des propriétés non-fonctionnelles des ressources en utilisant la grille d'évaluation
 - Une sélection des ressources retenues
- **TSM** : L'objectif de cette étape est de décrire en détail l'implémentation de la solution en utilisant les ressources choisies lors de la phase « BOTTOM TIM » et ceci d'un point de vue technique, matériel et humain. Cette étape permet également de trouver des solutions pour contourner les faiblesses identifiées de la sélection des ressources pendant la phase TIM. Par exemple réfléchir à un mécanisme de maintenance préventive pour assurer le plus possible la fiabilité du processus automatisé, mécanisme de formation aux ressources humaines, etc.

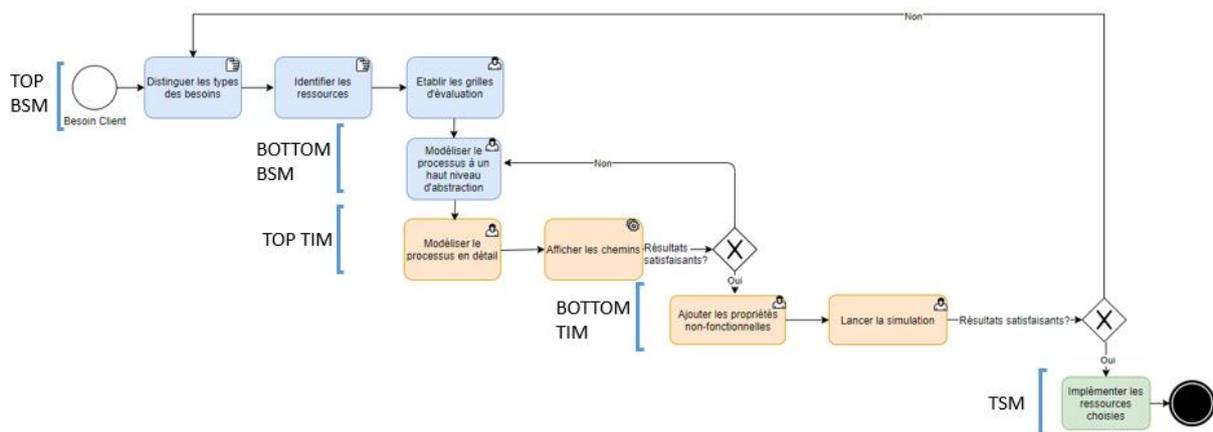


Figure 21: Représentation de la méthodologie d'automatisation des processus par caractérisation des ressources

Les parties suivantes détailleront les enrichissements apportés à l'outil eBPMN pour l'adopter à la méthodologie que nous proposons.

II. Distinction des ressources

Les ressources sont des composants importants pour une entreprise, car elles permettent la transformation et la valeur ajoutée au produit / service fourni au client (Figure 22). C'est la raison pour laquelle il est recommandé par l'approche MDSEA de les intégrer dès les premières phases de modélisation. Dans la même perspective, notre approche vise à incorporer les types de ressource dans toute les phases d'un projet d'automatisation des processus pour aider les différents types d'intervenants (analyste métier, consultant informatique et utilisateurs) à prendre la meilleure décision d'affectation des ressources pour chaque tâche, qu'elles soient humaines, matérielles ou IT.

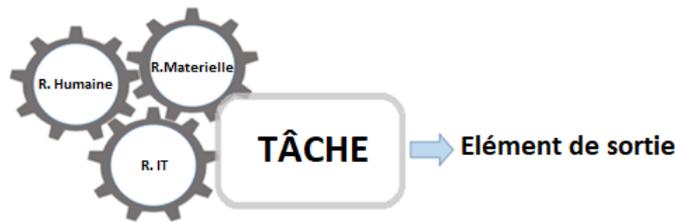


Figure 22:Types des ressources

Comme nous l'avons indiqué dans le chapitre de l'état de l'art et dans le chapitre des choix de langages de modélisation et de simulation, l'outil eBPMN utilise les annotations textuelles pour renseigner les informations non-fonctionnelles des ressources. Dans eBPMN, la ressource peut être humaine ou matérielle, mais aucune distinction syntaxique n'est proposée pour les représenter. La même syntaxe textuelle représente les deux types de ressources. De ce fait, dans nos travaux, nous avons différencié la syntaxe des trois types de ressources comme recommandé par l'approche MDSEA. Nous allons les présenter dans la partie de la contribution opérationnelle c'est-à-dire de l'application de notre contribution conceptuelle à un cas réel.

Après la distinction des types de ressource, il nous faut compléter l'apport de cette approche par la prise en considération de la différenciation de leurs caractéristiques. Pour ce faire, nous proposons d'utiliser une grille des performances qui nous permettra d'évaluer chaque profil de ressource impliquée dans le processus selon les caractéristiques de sa nature. Cette grille sera remplie lors de l'étape « TOP BSM » et sera exploitée pendant la phase de simulation dans l'étape « BOTTOM TIM ». Nous allons dans les parties suivantes présenter les caractéristiques que nous proposons dans la grille d'évaluation de chaque type. Compte tenu de la diversité des clients, cette grille doit pouvoir être modifiable selon leurs objectifs et leur secteur d'activité.

1. Grille des performances des ressources

Le travail effectué dans un processus est indéniablement accompli par les ressources qui exécutent les tâches qui composent le processus. L'un des objectifs de notre approche est de prendre en considération les caractéristiques de chaque type de ressource pour calculer leur degré d'efficacité dans un processus. Pour ce faire, nous proposons d'utiliser une grille d'évaluation contenant les critères à prendre en compte pour chaque type de ressource. La grille de performance permet de situer la qualité et la performance d'une ressource qui ne peut être évaluée tout simplement comme efficace ou inefficace. La grille est composée de plusieurs colonnes :

- Un **critère** est un élément de référence à partir duquel une prestation, un produit ou une performance sont estimés ou évalués. Un critère peut être divisé en sous-critères
- **L'échelle d'appréciation** permet d'indiquer la qualité ou la quantité de l'entité décrite
- Le **poids du critère** vise à déterminer le degré d'importance de prise en compte du critère

Etape 1 - Choisir les critères d'évaluation

Il est important de choisir des critères pertinents compte tenu de la nature de l'entité qu'on souhaite évaluer. Ils doivent permettre de se prononcer sur les principales dimensions qui caractérisent ce que l'on évalue en tenant en compte de l'objectif à atteindre. Il est recommandé que le choix des critères se fasse par les personnes qui maîtrisent l'entité à évaluer et qui ont une vision clarifiée de l'objectif souhaité.

Ainsi, la première étape consiste à choisir les bons critères d'efficacité de chaque ressource. Dans notre cas un critère représente la caractéristique d'évaluation de la ressource. Comme vu dans les chapitres précédents, la littérature scientifique a proposé plusieurs caractéristiques définissant le degré d'efficacité des ressources humaines, matérielles et IT. Pour ce travail, nous avons classé ces caractéristiques en prenant en considération deux critères :

- Caractéristiques communes à différents domaines : Une SSII travaille avec plusieurs secteurs d'activité, et de ce fait, nous devons choisir des caractéristiques communes aux différentes activités commerciales, avec la possibilité d'en ajouter de nouvelles à la demande du client,
- Caractéristiques objectives : Il faut également choisir un critère clairement défini et facilement calculable. Ainsi, les acteurs impliqués dans le remplissage de la grille atteindront tous les mêmes résultats parce qu'ils sont quantifiables.

Etape 2 - Choisir l'échelle d'appréciation

L'échelle d'appréciation est une succession d'éléments gradués qui correspond à plusieurs degrés de qualité visée par le critère. Ces éléments sont appelés échelons et leur nombre dépend du but souhaité. Les nombres d'échelons les plus utilisés sont :

- L'échelle à quatre échelons, son avantage est la rapidité d'application
- L'échelle à cinq échelons, son avantage et la possibilité d'utiliser une cote moyenne
- L'échelle à 6 échelons ou plus permet d'avoir un degré d'évaluation des critères plus fin mais il est plus difficile à utiliser car il nécessite d'explicitier les différentes valeurs de l'échelle.

Les types d'échelles les plus répandus sont numériques, alphabétiques ou qualitatifs. Ils peuvent également être descriptifs dans lesquels on décrit en détail le comportement recherché pour chaque échelon.

Dans notre approche nous avons choisi le type d'échelle numérique pour avoir une évaluation quantitative des performances des ressources et pouvoir les agréger. Nous avons également choisi un type d'échelle à trois échelons de 0 à 2, pour faciliter l'évaluation des caractéristiques d'une ressource. Deux étant le meilleur score.

Etape 3 – Choisir la pondération des critères

On peut considérer les critères avec des unités équivalentes ou leur accorder une pondération. La pondération à choisir pour chaque critère doit refléter l'importance qui lui est accordée en fonction de l'objectif à atteindre. Généralement, elle est exprimée sous forme d'un coefficient chiffré affecté à chacun des critères. Il est recommandé de classer les critères par ordre

d'importance pour faciliter leur pondération. Concernant la somme des poids, des approches préconisent de la contraindre à 1, d'autres approches proposent de calculer les poids en utilisant des méthodes classiques comme AHP. Dans notre approche, le calcul des poids est libre de toute contrainte, et nous pensons qu'il est plus judicieux et plus logique de laisser le client choisir la méthode de pondération de chaque critère en fonction des objectifs qu'il souhaite prioriser.

Nous allons présenter dans les parties suivantes les différentes caractéristiques de chaque type de ressource.

2. Ressources humaines

2.1. Caractéristiques des ressources humaines

Les ressources humaines représentent la principale richesse d'une entreprise de production ou de prestation de service. Chaque entreprise doit principalement sa réussite à son capital humain qui peut la différencier et la démarquer de ses concurrents. Malgré l'automatisation des processus, la présence des ressources humaines est indispensable. D'où l'importance de prendre en considération leurs spécificités pour choisir les ressources humaines adéquates aux différentes tâches afin de satisfaire les objectifs, la qualité et les délais de réalisation. L'efficacité des ressources humaines n'est pas statique et varie d'une période à une autre en fonction de l'environnement de travail et de leur niveau d'expérience, etc. En outre, les caractéristiques de performance d'une ressource humaine peuvent être perçues comme des contraintes pour la réussite dans l'accomplissement des tâches assignées. Nous allons présenter les caractéristiques et les mesures de performance des ressources humaines proposées dans la littérature scientifique.

Dans sa thèse, (Bouzidi, 2017) souligne l'importance de prendre en compte les contraintes relatives aux ressources humaines de manière explicite dans la planification des ordonnancements de production et de maintenance. Il considère alors les contraintes suivantes :

- **Disponibilité** : L'exécution des activités des processus est généralement conditionnée par la présence des ressources humaines adéquates. Or, les ressources humaines n'ont pas nécessairement le même rythme de travail. Il convient donc de tenir compte de leur disponibilité variable
- **Compétence** : La compétence est une caractéristique représentant l'aptitude d'une ressource humaine à réaliser ou non une activité donnée. Elle se compose en savoir de connaissance, en savoir-faire (pratique), en savoir-être (comportement relationnel) et en des aptitudes physiques. Elle est souvent mesurée selon le degré de maîtrise d'une activité par la ressource
- **Performance** : La performance est liée à la qualité et le temps opératoire effectué par un employé pour exécuter la tâche. Le temps de traitement d'un processus ne peut pas être constant car il dépend en partie des intervenants l'exécutant. Ainsi, la performance d'un processus dépend en partie de la performance de ses intervenants. La performance d'un opérateur humain est liée à ses compétences et principalement à son état physique.

Par exemple un employé peut être compétent pour réaliser une activité mais peut avoir une performance faible à la suite d'une charge de travail élevée.

(Jin, 2019) énonce que la réussite du processus de conception dépend de l'efficacité de la communication entre ses acteurs. De ce fait, il propose une allocation des ressources humaines en prenant en considération les caractéristiques suivantes :

- **Compétence** : reprend la définition donnée par (Bouzidi, 2017)
- **Créativité** : consiste à mesurer l'aptitude de la ressource humaine à trouver des idées et solutions nouvelles et pertinentes au contexte.
- **Disponibilité** : représente la difficulté de trouver des ressources pour exécuter les différentes tâches d'un processus
- **Taux d'occupation** : représente le rapport entre le temps de travail requis par une ressource humaine et le temps total alloué à l'exécution du processus
- **Satisfaction personnelle** : représente la satisfaction qu'éprouve un employé à exécuter la tâche assignée. Elle dépend du degré d'expérience vis-à-vis de l'exécution de la tâche, de la motivation, etc.
- **Education** : représente le niveau scolaire de la ressource humaine (sans diplôme, niveau licence, master, etc.)
- **Expérience** : représente le nombre d'années que l'employé a exercé une fonction
- **Age** : représente l'âge de la ressource humaine

Dans leur revue de littérature, (Xu et al., 2018) ont présenté les principales mesures à prendre en considération pour évaluer l'efficacité d'une ressources humaines pour évaluer les interfaces des systèmes dans la simulation:

- **Performance** : représente l'efficacité et la qualité du travail rendu par l'employé
- **Charge de travail** : représente la quantité de travail qu'un opérateur humain doit exercer durant la journée
- **Conscience de la situation « Situation awareness »** : représente l'aptitude d'une ressource humaine à identifier, comprendre et prédire les éléments importants et pertinents à l'accomplissement de la tâche assignée
- **Travail de groupe/collaboration** : représente l'aptitude de collaboration des opérateurs humains entre eux et des opérateurs humains avec les machines

En plus des dimensions décrites ci-dessus, d'autres chercheurs ont proposé des mesures et caractéristiques plus fines liées à l'aspect physique et psychique d'une ressource humaine comme le stress, la motivation, l'état général de santé, capacité physiologique, etc. La prise en considération de ces mesures permet de mieux évaluer leur influence sur la rapidité d'exécution et la qualité d'exécution des tâches assignées.

2.2.Grille des performances des ressources humaines

Les caractéristiques de performance des ressources humaines sont multiples et variées, nous proposons dans notre démarche de prendre en considération les caractéristiques objectives de sorte que les évaluateurs puissent facilement trouver les réponses à chacun des critères. Ce point

est le plus important dans le cas des ressources humaines où l'évaluation de leur performance est difficile et délicate à appréhender. Pour faire les choix des critères, nous nous sommes basés sur les caractéristiques énumérées ci-dessus et notre expérience avec les utilisateurs métiers.

Disponibilité

- Charge de travail et Capacité : Pour évaluer la disponibilité, nous proposons de nous baser sur la charge et la capacité de travail de l'utilisateur. Quand la charge de travail d'une ressource est supérieure à sa capacité, nous sommes dans le cas d'une surcharge de la ressource. Dans le cas contraire, où la charge de travail est inférieure à la capacité, il lui reste de la disponibilité pour le processus à automatiser.

Compétence

Pour évaluer la compétence d'un utilisateur nous proposons de la décomposer en savoir-faire métier et savoir-faire informatique.

- Savoir-faire métier représente l'ensemble des compétences pratiques et techniques liées au domaine de la tâche à exécuter. Il est généralement basé sur le fruit des retours des tâches similaires menées
- Savoir-faire informatique représente l'ensemble des compétences liées à l'utilisation des outils informatiques. Il est généralement jugé en fonction de la facilité qu'a un utilisateur à manier les outils informatiques et l'application mis à sa disposition

Performance

Pour évaluer la performance d'un utilisateur nous proposons de la décomposer en rapidité d'exécution et adéquation de la tâche.

- Rapidité d'exécution représente la capacité d'exécution qu'a un utilisateur pour traiter la tâche. Elle peut être liée à son niveau d'expérience.
- Adéquation de la tâche représente l'adéquation du domaine métier de la tâche au profil de l'utilisateur l'exécutant

Le Tableau 8 représente la grille d'efficacité des ressources humaines proposées.

	Disponibilité		Compétence				Performance				Résultats
	Charge de travail/Capacité		Savoir-faire métier		Savoir-faire informatique		Rapidité d'exécution		Adéquation de la tâche		
	Score	Poids	Score	Poids	Score	Poids	Score	Poids	Score	Poids	
Ressource humaine 1											
Ressource humaine 2											

Tableau 8: Grille de performance des ressources humaines

3. Ressources matérielles

3.1. Caractéristiques des ressources matérielles

Les ressources matérielles représentent les biens physiques (tangibles) utilisés par une entreprise pour mener à bien ses activités. Elles sont classées en plusieurs catégories : les locaux, les équipements et outillage de production, les équipements bureautiques, les équipements informatiques, etc. D'après (Edoze, 2010) la gestion et la performance des ressources matérielles contribuent significativement à l'amélioration des performances et la réduction des coûts. D'où l'importance de prendre en considération leurs caractéristiques pour choisir les ressources matérielles adéquates aux différentes tâches d'un processus. Le choix de leurs caractéristiques d'évaluation de leur efficacité dépend du secteur d'activité de l'entreprise et des objectifs à atteindre. Dans ce qui suit nous allons décrire les caractéristiques d'efficacité liées aux machines de production et aux machines informatiques.

Les indicateurs d'efficacité ou d'efficience des machines de production traduisent différentes notions, telles que les notions de rendement, de disponibilité, de fiabilité et de maintenabilité (Cauffriez et al., 1997). Dans notre étude, nous allons nous concentrer sur les caractéristiques qui jouent un rôle sur la disponibilité et la fiabilité des ressources matérielles. Il existe également d'autres méthodes comme l'AMDEC qui permettent d'étudier en profondeur la fiabilité des ressources, ces méthodes ne sont pas prises en compte dans notre étude.

Dans la littérature, plusieurs indicateurs de performance d'un système de production ont été proposés. Nous allons présenter les indicateurs d'évaluation de l'efficacité des machines les plus communs pour mesurer la non-performance des lignes de productions (Cauffriez et al., 1997; Taillard, 2008; Ouazene, 2013; Karolina et al., 2020). Parmi ceux liés à la disponibilité et la fiabilité :

- **La disponibilité propre (DP)** prend en considération uniquement les pertes occasionnées de production par arrêts propres. Elle est définie par le rapport suivant :

$$DP = \frac{\text{Temps de fonctionnement}}{\text{Temps de fonctionnement} + \text{Temps d'arrêt propre}}$$

- **La disponibilité opérationnelle (DO)** traduit la proportion de temps durant lequel la machine assure sa fonction par rapport au temps requis :

$$DO = \frac{\text{Temps de fonctionnement}}{\text{Temps requis}}$$

- **Le temps de fonctionnement moyen (TFM)** représente le temps moyen entre pannes (MTTF) :

$$TFM = \frac{\text{Temps de fonctionnement}}{\text{Nombre d'arrêts propres}}$$

- **Le temps d'arrêt propre moyen (TAPM)** représente la notion de maintenabilité et traduit le temps moyen de réparation (MTTR) :

$$TAPM = \frac{\text{Temps d'arrêt propre}}{\text{Nombre d'arrêts propres}}$$

- **Le taux de qualité (TQ)** représente la qualité de production et est le rapport entre la production bonne et la production totale réalisée

$$TQ = \frac{\text{Nombre de pièces bonnes}}{\text{Nombre total de pièces}}$$

- **Le taux de rendement synthétique (TRS)** représente le taux d'utilisation des machines :

$$TRS = \text{Taux de Disponibilité} \times \text{Taux de Qualité} \times \text{Taux de Performance}$$

D'autres caractéristiques sont impliquées dans la mesure de l'efficacité des systèmes de production telle que l'implantation de l'atelier : l'organisation des machines, déplacement et manutention, défauts logistique, etc. (Taillard, 2008)

La considération des ressources matérielles informatiques « Hardware » est aussi importante car elles représentent le socle des environnements d'exécution des applications. Un système d'application a besoin d'une machine pour fonctionner, et plus cette machine est performante, plus l'utilisation du système est fluide. Nous allons présenter les principales caractéristiques de mesure d'efficacité du matériel informatique (Paschke and Schnappinger-Gerull, 2006; Ravindran, 2002).

- **Disponibilité** « Availability » désigne la probabilité qu'un système soit opérationnel pour l'utilisateur
- **Temps d'arrêt maximal** « Maximum down-time » la durée maximale pendant laquelle des machines peuvent être interrompues sans causer de préjudice irréparable
- **Fréquence de défaillance** « Failure frequency » représente la fréquence de défaillance qui empêche le système d'accomplir sa tâche
- **Temps de réponse** représente la durée entre la soumission d'une instruction et sa réception par la machine
- **L'accessibilité des machines** « Accessibility in case of problems » évalue l'accessibilité aux machines en cas de problème
- **Temps de service** représente la durée du traitement d'une tâche effectuée par la machine
- **Temps processeur** représente le temps de traitement d'une tâche par processeur
- **Instruction par seconde** est une unité de mesure de la vitesse du processus de la machine
- **Taux d'utilisation du matériel** « Hardware utilization » est une unité de mesure qui permet d'évaluer le taux d'utilisation des machines d'une infrastructure

Ces indicateurs permettent d’avoir une vision très précise des performances des ressources matérielles mais ne sont pas tous adaptés à notre cas, car nécessitent beaucoup de données non disponibles lors de la conception d’un système de workflow.

3.2. Grilles des performances des ressources matérielles

Comme présenté, les caractéristiques de performance des ressources matérielles dépendent principalement de la nature du bien tangible utilisé. Les clients d’Exakis Nelite ont plus recours à l’automatisation des processus de service ou des processus administratifs. Pour ce type de processus les machines de production ne sont pas utilisées. De ce fait nous avons choisi de considérer la partie infrastructure d’une application comme étant une ressource matérielle. Pour faire les choix des critères, nous nous sommes basés sur les caractéristiques énumérées ci-dessus et sur notre expérience avec les utilisateurs.

Disponibilité

- Charge de travail d’un serveur représente le nombre d’applications utilisant le même serveur ainsi que le nombre d’utilisateurs ayant recours au serveur
- Dimensionnement d’un serveur/infrastructure représente la conformité d’installation aux exigences requises

Performance

- Performance du processeur représente la fréquence du processeur donc sa vitesse d’exécution
- Taille de la mémoire représente sa vitesse et sa réactivité à recevoir et traiter les données
- Fréquence de lecture/écriture du disque dur représente la rapidité d’exécution en lecture et écriture des données

Accessibilité des machines

- Rapidité de résolution représente la vitesse de traitement des défaillances du matériel

Concernant les applications hébergées dans le cloud les critères de disponibilité et de performance seront évalués de manière générale sans prendre en considération les sous-critères qui sont en la possession de l’hébergeur.

Le Tableau 9 représente la grille d’évaluation d’efficacité des ressources matérielles proposées.

	Disponibilité				Performance						Accessibilité		Résultats	
	Charge de travail		Dimensionnement		Performance Processeur		Taille de la mémoire		Fréquence de lecture/écriture		Rapidité de résolution			
	Score	Poids	Score	Poids	Score	Poids	Score	Poids	Score	Poids	Score	Poids		
Ressource matérielle1														
Ressource matérielle2														

Tableau 9: Grille des performances des ressources matérielles

4. Ressources IT

4.1. Caractéristiques des ressources IT

La technologie de l'information est omniprésente dans les entreprises, elle est considérée comme un catalyseur important à l'innovation et à la réussite des entreprises. Les ressources informatiques sont considérées comme le fondement technologique d'une entreprise pour fournir en temps réel, des informations précises et complètes pour la communication. Il s'agit de tous les matériels, logiciels et infrastructures sur lesquels les outils informatiques sont construits. La définition des indicateurs d'efficacité des ressources informatiques dépend de l'objectif à atteindre. Par exemple, certains se concentrent sur les performances du matériel comme les processeurs, tandis que d'autres se concentrent sur les performances des logiciels comme la rapidité de chargement et d'autres sur les performances de l'infrastructure. Les métriques généralement utilisées sont : débit binaire, délai de transmission, disponibilité. Dans le cadre de ce travail de thèse, les ressources IT représentent les applications informatiques développées.

(Kopyltsov, 2020) a proposé plus de 23 indicateurs pour évaluer la qualité d'un logiciel. Nous présentons ici celles qui reviennent le plus souvent dans la littérature scientifique :

- **La vitesse d'entrée/sortie** représente la satisfaction du temps de chargement des informations par l'application
- **L'efficacité d'utilisation de la mémoire** représente si l'application consomme beaucoup de ressources
- **La complétude** représente si le produit logiciel contient tous les composants nécessaires au besoin
- **La robustesse** représente la capacité d'une application à se remettre des situations erronées ou de conditions stressantes
- **Tolérance aux fautes** représente l'aptitude d'une application à accomplir sa fonction malgré la présence ou l'occurrence de fautes
- **Simplicité** représente le degré de prise en main de l'application
- **Ressemblance** représente le degré de similarité des composants de l'application avec les produits logiciels
- **Interopérabilité** représente le degré de communication possible de l'application avec d'autres environnements/applications hétérogènes
- **Sécurité** représente la possibilité d'authentification et possibilité de gestion des droits d'accessibilité des informations pour chaque utilisateur
- **Maintenabilité** représente le degré de flexibilité du produit pour l'ajout de nouveaux besoins
- **Assistance** représente le degré de facilité d'identification des sources d'erreurs/bugs

4.2. Grille de performance des ressources IT

Comme on peut le constater, les produits logiciels ont leurs propres caractéristiques qui ne sont pas similaires à celles des produits matériels. Pour faire les choix des critères, nous

nous sommes basés sur les caractéristiques énumérées ci-dessus et sur notre expérience avec les demandes des utilisateurs.

Performance logicielle

- Consommation des ressources représente le taux d’utilisation de l’application des ressources matérielles (processeur, mémoire, etc.)
- Stabilité représente la capacité d’une application à préserver son temps de réponse aux exécutions demandées (ne se met pas en erreur, ne se rafraîchit pas subitement, etc.)
- Tolérance aux fautes on garde la même définition que celle présentée dans le paragraphe ci-dessus
- Simplicité on garde la même définition que celle présentée dans le paragraphe ci-dessus

Assistance

- Maintenance représente le degré de facilité fourni par la technologie de développement pour identifier les sources des bugs
- Rapidité de résolution représente la vitesse de traitement des bugs par le support technique de la technologie choisie

Le Tableau 10 représente la grille d’évaluation d’efficacité des ressources IT proposées.

	Performance Logicielle								Assistance				Résultats
	Consommation des ressources		Stabilité		Tolérance aux fautes		Simplicité		Maintenance		Rapidité de résolution		
	Score	Poids	Score	Poids	Score	Poids	Score	Poids	Score	Poids	Score	Poids	
Ressource IT 1													
Ressource IT 2													

Tableau 10: Grille de performance des ressources IT

III. Agrégation des performances des ressources

Comme énoncé dans les chapitres précédents, une seule tâche peut être exécutée par une ressource composée. De ce fait, la performance de cette tâche est liée à la performance de toutes ses ressources. Par conséquent, pour pouvoir calculer la performance d’une tâche il faut agréger les performances des ressources la composant. A cet effet, nous proposons dans cette section notre démarche d’agrégation des performances des ressources en nous appuyant sur la décomposition et les travaux utilisés dans notre laboratoire par (Heguy, 2018b). Ces travaux s’appuient sur les propositions de réduction et d’agrégation des performances des processus de (Ducq, 1999) et (Cardoso et al., 2004).

1. Méthodes utilisées

1.1. Décomposition

Comme vu dans les chapitres précédents, la modélisation et la simulation des besoins permettent de mettre au clair les anomalies qui peuvent émerger dans un environnement de production. Les méthodes de décomposition font parties des techniques qui permettent d'aider à lever les ambiguïtés concernant ces problématiques soulevées. Chacune des méthodes proposées a sa propre démarche et ses propres spécificités, mais elles se basent toutes sur le même principe : l'objectif à traiter est décomposé en un ou plusieurs sous objectifs faciles à étudier. Ainsi, le but principal de la décomposition est de décomposer le problème ou objectif original, de manière récursive, en plusieurs sous-objectifs dont la finalité est connue. Les approches par décomposition peuvent être résumées en trois étapes (Ouazene, 2013) :

- L'identification et caractérisation des sous-objectifs
- La définition d'équations reliant les paramètres des sous-objectifs décomposés
- Le développement d'un algorithme de résolution de ces équations

A l'instar des trois étapes, dans notre approche, nous souhaitons décomposer les ressources assignées à une tâche pour faciliter l'étude de leur performance. De ce fait, nous allons dans un premier temps identifier les ressources de chaque tâche atomique avec leur performance, ensuite identifier la structure de leur agencement et pour finir déterminer un mécanisme reliant leur performance. Nous allons présenter ces mécanismes dans les parties suivantes.

Concernant l'identification de la structure d'agencement, nous allons prendre en considération trois décompositions génériques adaptées à tous les domaines et couvrent les principaux besoins à modéliser, et qui sont simples à implémenter :

Décomposition séquentielle : consiste à découper une activité en plusieurs sous-activités individuelles se déroulant de façon séquentielle (Figure 23).

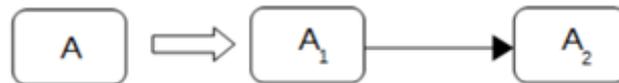


Figure 23: Décomposition séquentielle

Décomposition alternative (en OU) consiste à décomposer une activité en sous-activités individuelles travaillant indifféremment, soit l'activité A1 est exécutée soit l'activité A2 (Figure 24)

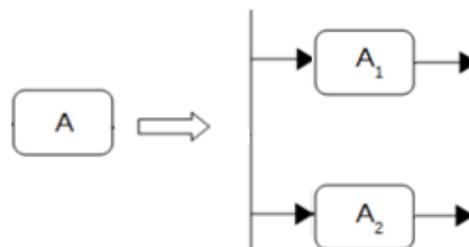


Figure 24: Décomposition alternative

Décomposition parallèle (en ET) consiste à décomposer une activité en sous-activités individuelles travaillant en parallèle (Figure 25)

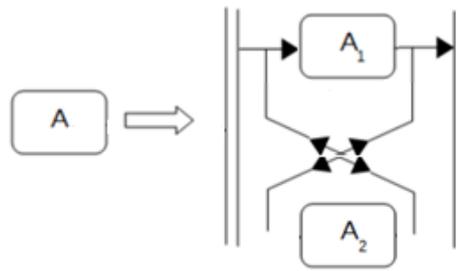


Figure 25: Décomposition en parallèle

1.2. La méthode de réduction

La réduction des processus consiste à appliquer un ensemble de règles de réduction à un ensemble de tâches afin d'en agréger la performance. A chaque application de règle de réduction, la structure du processus change. Plusieurs itérations de réduction sont appliquées jusqu'à ce qu'il ne reste qu'une tâche atomique dont la performance sera la performance globale du processus. Les mesures de la performance de cette tâche correspondent donc à celles de l'ensemble des tâches qui ont été réduites. (Cardoso et al., 2004) a développé six règles de réduction qui couvrent la majorité des implémentations possibles d'un processus : séquentielle, parallèle, conditionnelle, tolérance aux fautes, boucle et réseau.

Pour notre approche nous appliquons le même principe de réduction sur les ressources décomposées pour obtenir la performance globale de la tâche. Dans le même cas que (Heguy, 2018b), nous allons retenir uniquement trois règles de réduction : séquentielle, parallèle et conditionnelle. La règle de réduction des réseaux correspond à la réduction des sous-processus. Les règles de réduction boucles et tolérance aux fautes sont plus adéquate aux modèles correspondant à l'implémentation des processus. Dans la présente étude, il s'agira de réduire la structure d'agencement des ressources d'une seule tâche. De ce fait, la notion de sous-processus, tolérance aux fautes et boucles sont inapplicables.

La réduction dans un agencement séquentiel, comme le montre la Figure 26, chaque activité se déroule intégralement avant de passer à l'activité suivante.

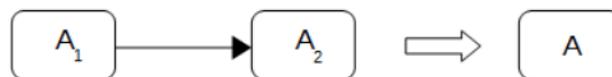


Figure 26: Réduction séquentielle

La réduction dans un agencement alternatif (en OU), comme le montre la Figure 27, l'exécution de l'activité se fait indifféremment, soit l'activité 2 soit l'activité 3.

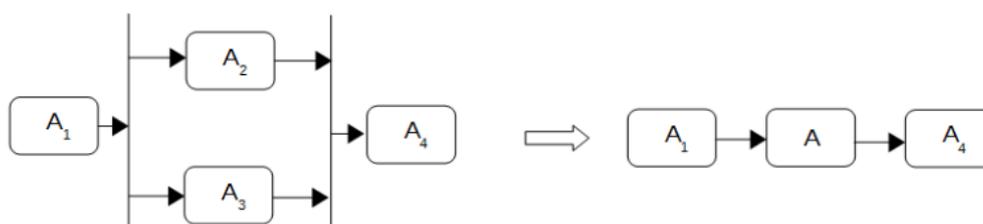


Figure 27: Réduction en OU

La réduction dans un agencement en parallèle (en ET), comme le montre la Figure 28 les activités 2 et 3 s'effectuent en parallèle

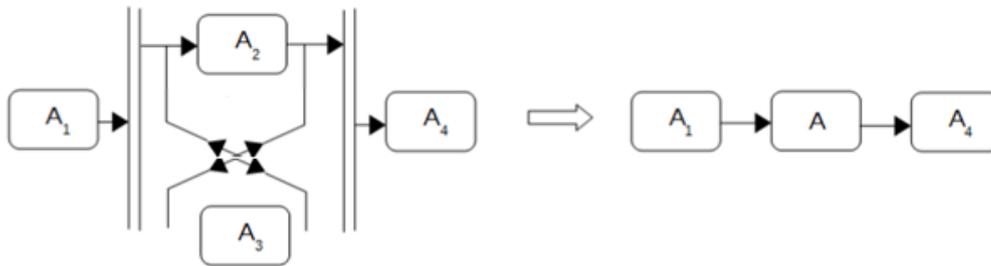


Figure 28: Réduction en ET

1.3.Agrégation des performances

Les méthodes d'agrégation sont utilisées dans plusieurs domaines tels que le traitement d'images, les systèmes multicritères pour la prise de décision, la biologie moléculaire, les systèmes réseaux, etc. L'objectif commun est de pouvoir obtenir une interprétation globale à partir de plusieurs données fournies par de nombreuses sources. Les formes d'agrégation sont variées. Selon (Dosimont, 2006), n peut agréger par un produit similaire, par ensemble de machines, par type de performance, etc. De manière générale, l'agrégation est une opération qui combine un ensemble d'éléments indépendants mais de même nature pour produire un seul élément qui les synthétise (Kromm, 2002). Nous nous intéressons ici à l'agrégation des performances. Cette opération consiste à synthétiser les expressions de performance élémentaires en une expression de performance globale (Sahraoui et al., 2007).

L'évaluation des performances d'un processus est étroitement liée à l'agencement des tâches le composant. Chaque type de réduction a sa propre formule de calcul d'agrégation des performances. (Ducq, 1999; Cardoso et al., 2004) ont défini les formules d'agrégation de plusieurs types de performance. Nous nous focalisons sur le temps de service et la fiabilité :

L'agrégation séquentielle

La valeur du temps de traitement est égale à la somme des valeurs des durées (D) de traitement des activités détaillées

$$\text{Temps de traitement} = \sum_{i=1}^d D_i$$

- La valeur de la fiabilité est égale au produit des valeurs de fiabilité des activités détaillées

$$\text{Fiabilité} = \prod_{i=1}^r R_{ei}$$

L'agrégation en OU

- La valeur du temps de traitement est égale au maximum des valeurs des durées des activités détaillées

$$\text{Temps de traitement} = \mathbf{Max}(D1, \dots, Di)$$

- La valeur de fiabilité est égale au minimum des valeurs de fiabilité des activités détaillées

$$\text{Fiabilité} = \mathbf{Min}(R1, \dots, Ri)$$

L'agrégation en ET

- La valeur du temps de traitement est égale au maximum des valeurs des durées des activités détaillées

$$\text{Temps de traitement} = \mathbf{Max}(D1, \dots, Di)$$

- La valeur de fiabilité est égale au produit des valeurs de fiabilité des activités détaillées

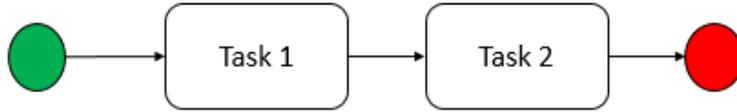
$$\text{Fiabilité} = \prod_{i=1}^r Rei$$

2. Application sur l'agrégation des performances des ressources

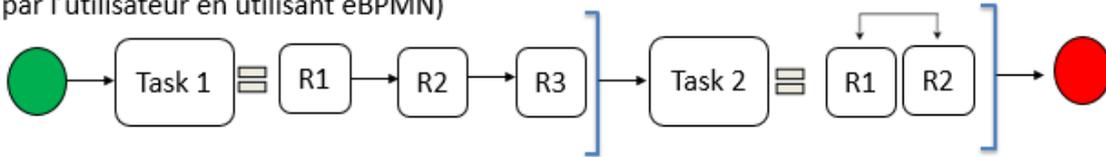
Comme expliqué dans le chapitre des choix de langage de modélisation et de simulation, une tâche atomique peut être exécutée par plusieurs ressources, et de ce fait, ses performances sont liées aux performances de ses ressources. Par conséquent, l'agrégation des performances des ressources associées à la nature et description de la tâche définit la performance globale de la tâche.

L'agrégation des ressources consiste à décrire comment les ressources fonctionnent ensemble et représentent une seule entité cohérente et structurée. Ainsi, dans notre cas, l'agrégation des performances des ressources consiste à combiner toutes les mesures associées aux ressources affectées à chaque tâche. La Figure 29 illustre le fonctionnement global de la démarche. Pour chaque tâche d'un processus nous allons décomposer la tâche en ressources atomiques, les réduire et agréger leur performance.

1. Modèle opérationnel du processus (par l'utilisateur en utilisant BPMN)



2. Décomposer chaque tâche en sous-tâche atomique exécutée par une seule ressource (par l'utilisateur en utilisant eBPMN)



3. Réduction et agrégation des performances des ressources (implémenté dans le code de simulation)

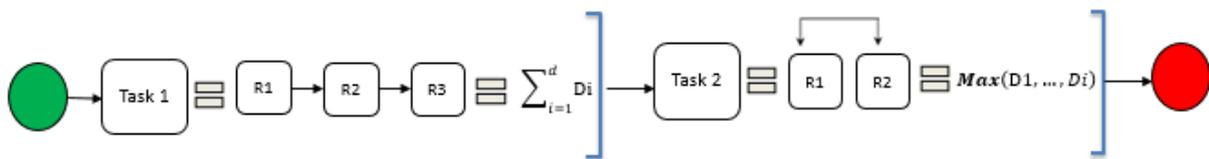


Figure 29: Etape de décomposition/agrégation des ressources

La décomposition de la tâche consiste à la diviser en un ensemble structuré de sous-tâches atomiques. Ainsi, chaque sous-tâche de cette décomposition doit être exécutée par une seule ressource comme le montre la Figure 30.

Pour ce faire :

- On identifie les ressources d'une tâche
- On identifie la structure d'agencement des ressources identifiées qui permet de réaliser la valeur de sortie escomptée par la tâche
- On identifie les mesures de performance de chaque ressource

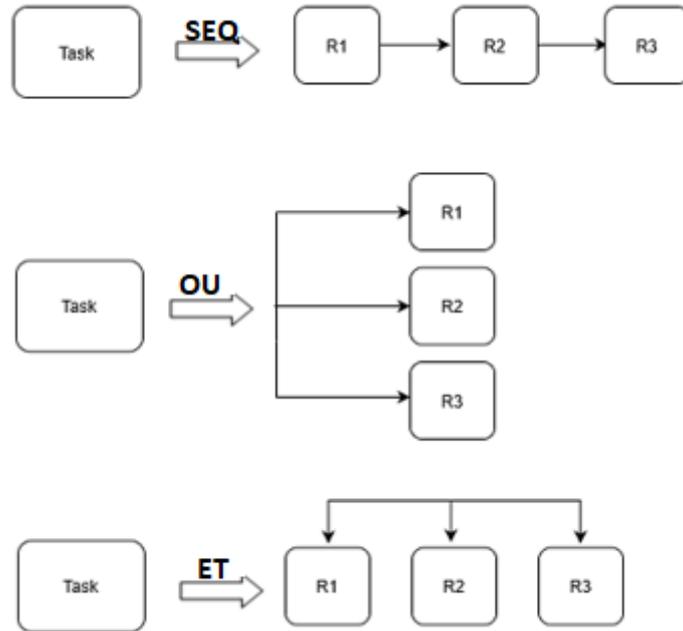


Figure 30: Décomposition d'une tâche en sous-tâches atomique

Après l'étape de décomposition d'une tâche en tâche atomique faisant appel à une ressource unique, vient l'étape de réduction et d'agrégation des performances des ressources selon les règles présentées dans le paragraphe précédent et synthétisées dans le Tableau 11.

Pour illustrer cette démarche, nous allons prendre l'exemple d'une tâche de validation d'un document.

- 1) Cette tâche peut être décomposée en deux sous-tâches atomiques ou plus selon le degré de granularité d'analyse souhaité. Pour valider un document on a besoin de :
 - Un approbateur qui est une ressource humaine
 - Un formulaire de validation qui est une ressource IT
 - On peut décomposer encore avec d'autres ressources matérielles comme, la plateforme d'exécution, le pc portable utilisé, etc.

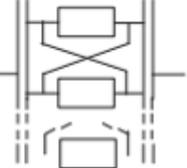
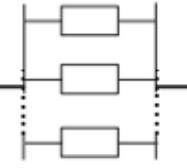
	Durée de traitement	Fiabilité
<p>SEQUENTIAL</p> 	$\sum_{i=1}^D Di$	$\prod_{i=1}^R Ri$
<p>AND</p> 	$Max(D1, \dots, Di)$	$\prod_{i=1}^R Ri$
<p>OR</p> 	$Max(D1, \dots, Di)$	$Min(R1, \dots, Ri)$

Tableau 11: Les règles d'agrégation de performance

On identifie par la suite les mesures de performances de ces ressources atomiques, et dans ce cas on prendra la durée de traitement de chaque ressource :

- Durée de traitement Approbateur = 10 min
- Durée de traitement Formulaire= 13 min
- Durée de traitement de la plateforme d'exécution = 14 min, (le traitement de la plateforme d'exécution consiste à effectuer les tâches nécessaires à l'utilisation du formulaire)

La structure d'agencement de ces tâches est une structure en parallèle.

2) Après l'étape de décomposition, on utilise la réduction des tâches et l'agrégation des performances. Puisqu'il s'agit d'une structure en parallèle, la durée de traitement est égale au maximum des valeurs déterminées. La Figure 31 représente le résultat final.

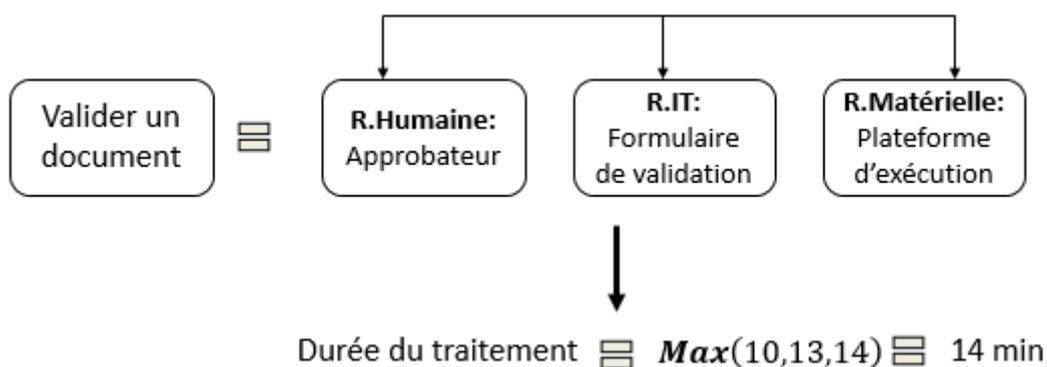


Figure 31: Décomposition en trois tâches atomiques

Le but de l'agrégation des performances des ressources est d'obtenir une vision synthétique et globale des performances et de l'efficacité d'un processus. De manière globale, elle aide l'analyste métier à identifier plus facilement le chemin le plus critique à analyser et à améliorer. De plus, prendre en compte la configuration d'agencement des ressources qui exécutent une tâche permet d'obtenir des mesures de performances et d'efficacité plus pertinentes et plus précises car comme le montre le Tableau 11, chaque structure a sa propre règle d'agrégation.

IV. Interface graphique

Dans cette partie nous allons décrire la ressource composée que nous proposons pour simuler la performance des interfaces graphiques.

Comme évoqué dans la partie des choix de langage, plusieurs tâches sont identifiées dans BPMN pour distinguer visuellement les spécificités de chacune d'elle. Dans ce chapitre, nous allons nous intéresser aux aspects de la tâche utilisateur qui est fréquemment utilisée dans le langage BPMN 2.0 et illustre la représentation typique d'une tâche de workflow « Workflow Task ». La tâche utilisateur est une tâche effectuée par une ressource humaine à l'aide d'une application informatique (Gestionnaire de processus, gestion de contenu d'entreprise, etc.). Ces applications offrent différents moyens pour interagir avec le système informatique ; l'interface graphique est le moyen d'échange le plus utilisé entre la ressource humaine et la ressource IT. De la même manière, la tâche utilisateur est généralement associée à l'interface graphique d'une application et est communément appelée « formulaire de tâche ». Elles sont également utilisées à grande échelle dans les workflows administratifs. De ce fait, la tâche utilisateur est composée d'une ressource humaine (utilisateur), d'une ressource matérielle (plateforme d'exécution) et d'une ressource IT (formulaire de tâche). Nous allons nous focaliser dans notre travail sur la relation entre l'utilisateur et le formulaire de tâche (Figure 32).

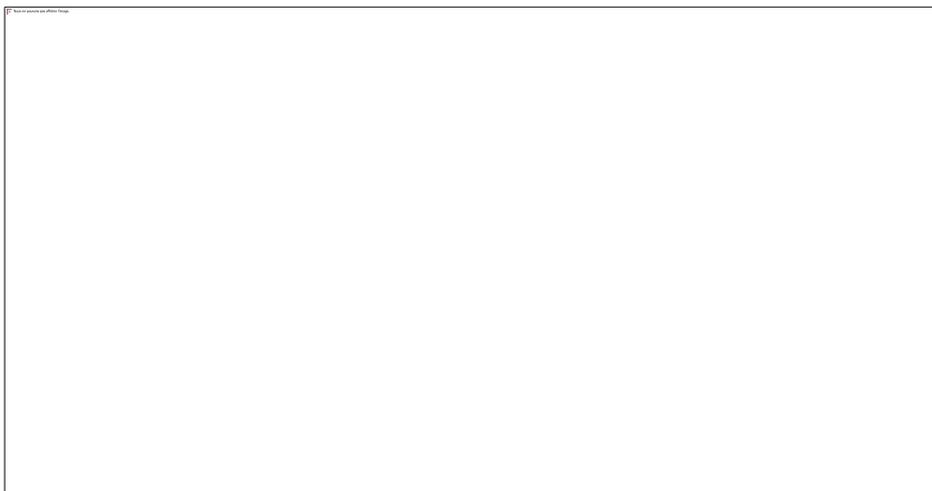


Figure 32: Composition de la tâche utilisateur

Un formulaire de tâche est une interface graphique composée d'un ensemble d'éléments graphiques affichés sur un écran qui permet de faciliter la communication entre l'utilisateur et le processus automatisé. Son utilisation est similaire à celle d'un formulaire en papier,

l'utilisateur devant remplir des champs et ensuite valider le formulaire. Cette validation va déclencher la poursuite de l'exécution du workflow.

Plusieurs travaux de recherche se sont intéressés à l'étude des interfaces graphiques qui représentent une partie importante du domaine de l'Interaction Homme Machine. Le but de ces recherches et de proposer des méthodologies et des techniques pour développer un formulaire qui permet aux utilisateurs une interaction conviviale et intuitive. Comme évoqué dans l'analyse de l'état de l'art, les recherches effectuées se sont plus concentrées sur l'importance de l'ergonomie d'un point de vue visuel et psychologique pour développer une interface graphique la plus adéquate au besoin et la plus intuitive possible (proposition d'approche de développement d'interface, proposition d'outil pour tester les interfaces graphiques, proposition de démarche d'évaluation des interfaces, etc.). Ces démarches sont généralement adoptées après la phase de modélisation et de simulation.

Les propositions d'amélioration de l'ergonomie des interfaces graphiques sont utiles et importantes car elles prennent en considération le facteur humain dans leur approche. Néanmoins, bien souvent, l'analyse de ses composants se fait à la dernière étape après la finalisation de l'analyse des fonctionnalités du processus à automatiser. De ce fait, cette ergonomie est rarement considérée comme un composant à part entière et important du workflow, et ses propriétés non-fonctionnelles sont brièvement étudiées. A notre connaissance, aucun travail de recherche n'a proposé une méthodologie ou technique pour mesurer l'efficacité de ses performances non-fonctionnelles pendant la phase de modélisation et de simulation. Or, comme évoqué dans l'analyse de l'état de l'art, l'étude des propriétés non-fonctionnelles pendant la phase de conception est importante pour aligner l'outil à développer au besoin à automatiser. De ce fait, dans cette étude, nous proposons le concept d'une ressource composée qui représente l'interface graphique.

Puisque les ressources IT et humaines sont impliquées pour exécuter la tâche utilisateur, le degré d'efficacité de la tâche dépend du degré d'efficacité de la combinaison des deux ressources. Au niveau de la ressource humaine, étudier la durée de remplissage de chaque champ du formulaire par l'utilisateur peut aider à se poser les bonnes questions et à apporter des réflexions sur le contenu du formulaire, la quantité des informations requises et les aides possibles qui peuvent faciliter son remplissage. Au niveau de la ressource IT, ce travail permet de se poser les bonnes questions sur la possibilité d'ajout des techniques facilitant le remplissage du formulaire et leur impact sur sa performance. Pour certaines technologies, plus on ajoute des composants élaborés dans le formulaire, plus sa performance se dégrade considérablement. De ce fait, ce travail peut également faciliter le choix de la technologie à utiliser. L'objectif de ce travail est de proposer une simulation de granularité fine, et de ce fait, l'obtention des informations sur les différents composants utilisés dans une interface graphique est nécessaire (ex : les champs comme champ de texte, checkbox, etc.). L'idée est d'attribuer à chaque composant (champ) les attributs de performance liés à l'utilisateur qui remplit le formulaire de tâche. Par exemple, comme le montre la Figure 33 pour remplir le champ « Description » du formulaire de tâche « Purchase Request », l'utilisateur a besoin de cinq minutes, et pour remplir le champ « Name » il a besoin de trois secondes. Nous pensons que cela aidera les parties prenantes (analyste métier, consultant IT, designer d'interface, etc.) à :

- Choisir les meilleures techniques de développement pour implémenter l'interface graphique (remplir automatiquement des champs, fournir des aides de remplissage, etc.)
- Prendre en considération des différents niveaux d'expérience des utilisateurs
- Attribuer l'utilisateur adéquat à la tâche en prenant en considération son expérience, ses forces et faiblesses lors de l'exécution des tâches similaires

The image shows a 'Purchase Request' form. On the left, a box labeled 'Task' with a person icon and an arrow points to the form. The form has the following fields: 'Name' with a value of '3' and a red asterisk; 'Department' with a value of '2' and a dropdown menu showing 'Please select a value...'; 'Request Date' with a value of '2' and a date/time picker showing '00' and '00'; and 'Description' with a value of '5' and a large text area. At the bottom, there are two buttons: 'SAVE AND SUBMIT' and 'CANCEL'.

Figure 33: Exemple d'une interface graphique

La Figure 34 présente la représentation conceptuelle de l'interface graphique qui est une ressource composée d'un opérateur humain et d'un moyen IT. Pour faciliter son implémentation, nous avons choisi de la considérer comme une ressource spécialisée d'une ressource IT caractérisée par le temps de chargement « Loading time ». L'interface graphique est composée d'un ensemble de champs et d'éléments caractérisés par le temps de remplissage par la ressource humaine. Quant au temps de chargement, il est alimenté en fonction de l'impact des éléments utilisés dans l'interface graphique.

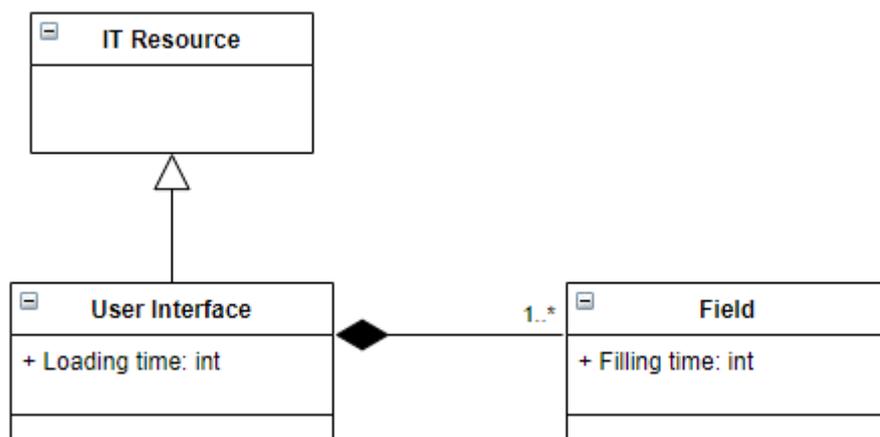


Figure 34: Représentation conceptuelle de la ressource Interface Graphique

V. Analyse des propriétés non-fonctionnelles en se basant sur un chemin

Un processus métier est composé d'un ensemble de flux traversant les activités qui utilisent des passerelles pour diviser et fusionner les chemins possibles. Les passerelles divisent une tâche en différents flux possibles et produisent à cet effet plusieurs alternatives de chemins d'exécution. Ainsi, les chemins d'un processus représentent tous les cas possibles de son utilisation. La représentation des détails fonctionnels et non-fonctionnels des chemins d'un processus aide les parties prenantes à mieux l'analyser en identifiant plus facilement les premières pistes d'amélioration. De plus, l'analyse de l'impact des différents scénarios de changement d'un modèle est mieux illustrée en affichant les mesures des propriétés non-fonctionnelles d'un chemin d'un processus, plutôt que ceux de chaque tâche isolée. Cela rend l'analyse plus globale et plus efficace en facilitant l'identification des chemins critiques, ce qui permet de mieux se focaliser sur les solutions potentielles pour améliorer le flux. De ce fait, on peut catégoriser les objectifs d'une analyse orientée chemin comme suit :

- Au niveau fonctionnel : Identifier et énumérer les différents chemins d'un processus est un moyen stratégique pour s'assurer que le modèle de processus est conforme aux exigences fonctionnelles. Cela permet d'identifier plus facilement les erreurs commises pour les corriger (tâche manquante, erreur de flux de séquence, etc.). Il s'agit d'une première étape essentielle pour s'assurer que le chemin à étudier atteint les objectifs opérationnels et qu'il est atteignable
- Au niveau non-fonctionnel : Pour améliorer l'efficacité d'un processus, il est plus intéressant de concentrer les efforts sur l'investigation des cas les plus probables à être exécutés en production. Par exemple pour un processus donné, quatre chemins peuvent être parcourus : chemin A, B, C et D avec respectivement, une probabilité d'exécution de 10%, 75%, 50% et 20%. Les chemins B et C ont le plus de chance d'être accomplis en production et ainsi représentent les premières pistes d'investigation en cas d'anomalie. L'état de performance de ces chemins représente l'état fréquent de performance du processus
- Au niveau de l'implémentation : Comme dans le domaine du développement logiciel, les chemins les plus exécutés en production sont considérés comme des chemins critiques, ils sont nommés en anglais « hot path » (Buse and Weimer, 2009). « Hot path » représente la partie du code la plus critique au niveau des performances puisqu'il s'agit de la partie la plus fréquemment exécutée. C'est la raison pour laquelle, les efforts d'optimisation de code sont concentrés sur cette partie. Dans la même perspective, l'analyse orientée chemin dans le cadre d'automatisation des processus, va aider les consultants IT à mieux centraliser leurs efforts sur les moyens les plus efficaces pour mettre en œuvre les chemins les plus fréquemment exécutés.

La méthode d'affichage des résultats de simulation en fonction du chemin parcouru est déjà proposée par les outils industriels. Cependant, comme évoqué dans l'état de l'art, ces outils ne sont pas flexibles et ne respectent pas entièrement les spécifications de l'OMG pour implémenter le langage BPMN 2.0. A notre connaissance, les propositions des outils de

simulation open source qui pallient cette problématique, affichent les résultats de simulation pour chaque tâche et de manière globale mais sans indiquer les chemins parcourus. Nous estimons qu'une investigation en se basant sur le chemin parcouru est d'une grande aide voire essentielle pour faire émerger les différents scénarios possibles de changement. Cela est encore plus vrai lorsqu'on se focalise sur les chemins les plus fréquentés en production.

L'objectif de ce travail est d'afficher les résultats de simulation obtenus des chemins choisis par l'analyste métier et les parties prenantes. Le choix du chemin (le scénario à simuler) peut être fait en utilisant les probabilités comme le permet l'outil eBPMN (Figure 35), dans notre approche nous proposons de spécifier explicitement le choix effectué par l'utilisateur (Figure 36). Par exemple, un processus peut conduire au chemin A si l'utilisateur approuve la demande ou au chemin B si l'utilisateur la refuse. Ainsi, l'analyste métier et les parties prenantes peuvent focaliser leur effort sur la recherche des solutions des chemins qui sont exécutés de manière fréquente.

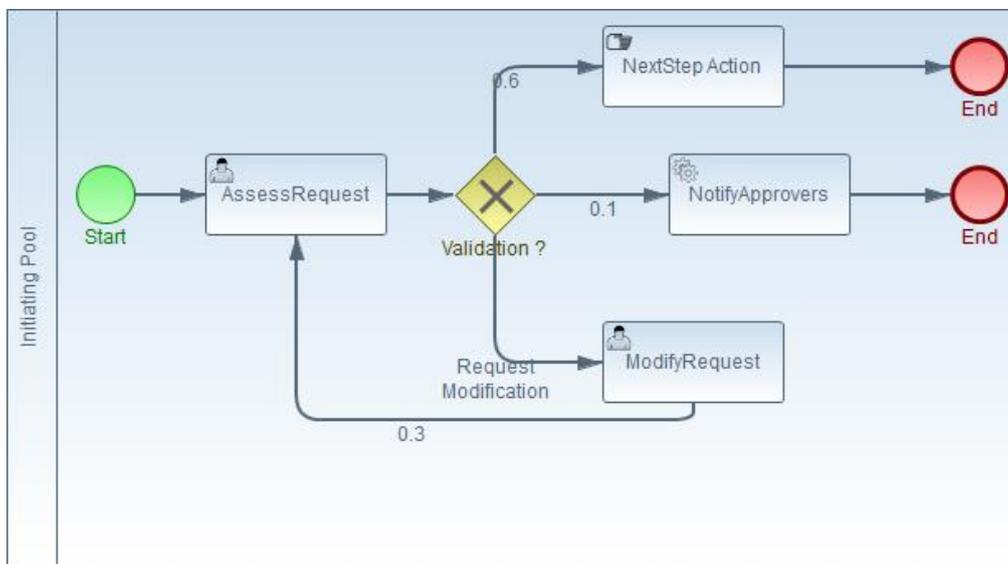


Figure 35: Choix des chemins de simulation en utilisant la probabilité

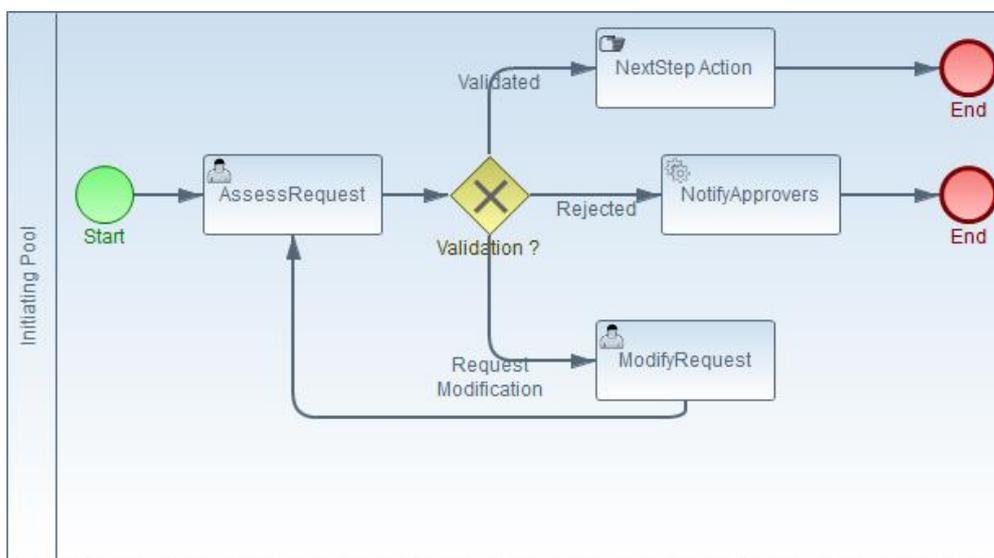


Figure 36: Choix des chemins de simulation en explicitant la décision de l'utilisateur

Nous avons également modifié les options d’affichage des résultats de simulation de l’outil eBPMN. eBPMN permet d’afficher les résultats globaux du processus et ceux de chaque tâche sans indiquer les chemins parcourus par la simulation. Dans notre travail, nous proposons deux options d’affichage en plus à l’outil (Figure 37):

- Afficher le résultat global de simulation de chaque chemin parcouru en utilisant l’agrégation des performances des ressources. Cela permet d’avoir une vue globale sur les différentes performances du processus, ce qui peut être une aide considérable pour identifier les chemins les moins efficaces et commencer par leur amélioration
- Afficher le résultat global de simulation de chaque type de ressource. Cette option permet d’avoir la performance globale de chaque type de ressource en fonction du chemin parcouru : on affiche la performance des ressources humaines, ressources matérielles et ressources IT. Ce type d’affichage permet d’identifier le type de ressource qui a le plus besoin d’investigation et d’amélioration
- Afficher le résultat de chaque ressource comme le propose nativement l’outil eBPMN, On y ajoute alors uniquement le chemin parcouru pour faciliter l’investigation

L’idée est de proposer des options d’affichage inspirées du principe d’IDE, du plus global au plus détaillé pour fournir plus d’indications concernant les pistes à améliorer. En outre, d’un point de vue cognitif, l’être humain a plus de facilité à analyser des données en partant du global vers le détail car dans le cas contraire il est difficile de restituer le détail au sein du modèle plus global.

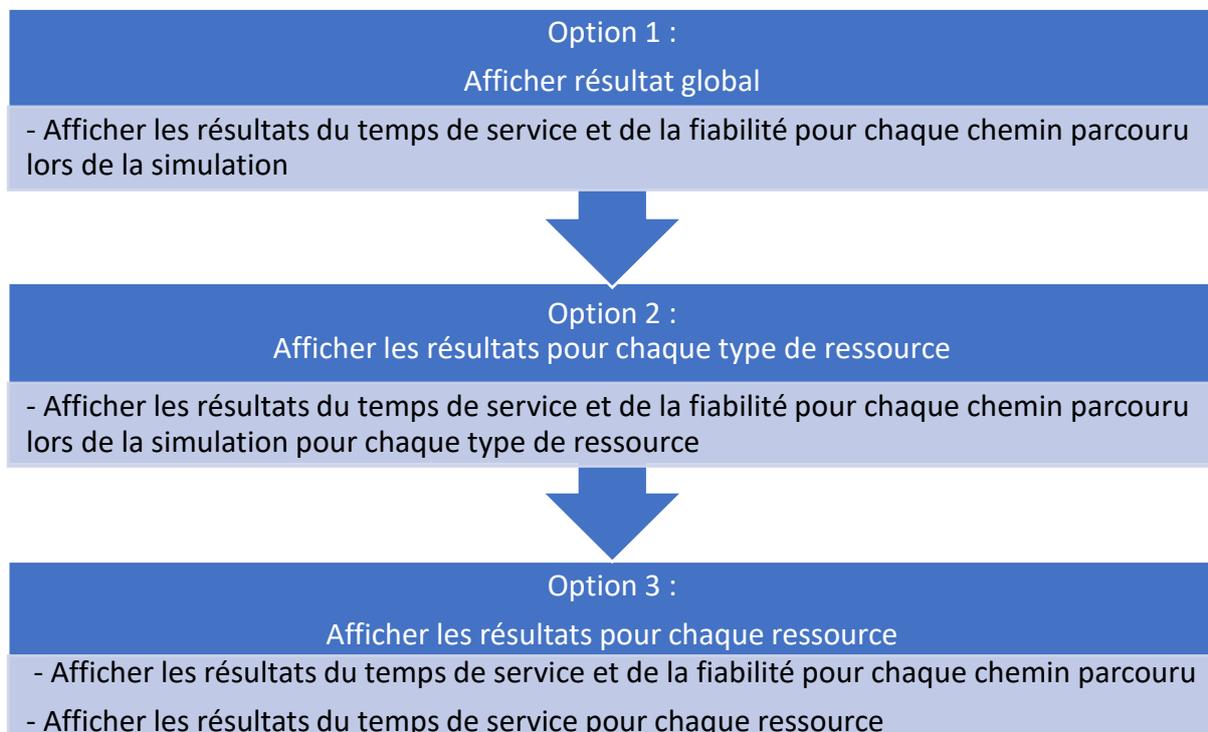


Figure 37: Les options d’affichage des résultats

VI. Conclusion

Dans ce chapitre nous avons présenté notre approche d'étude des processus à automatiser par caractérisation des ressources en se basant sur la méthodologie MDSEA et en utilisant les langages BPMN et eBPMN pour la modélisation et la simulation.

Dans un premier lieu, nous avons énuméré les raisons du choix du langage BPMN 2.0 comme langage de modélisation. À la suite de l'analyse de sa capacité à représenter les ressources, nous avons constaté qu'une seule tâche pouvait être exécutée par plusieurs ressources. De ce fait, ce critère faisait partie de la raison de notre sélection du langage eBPMN comme outil de simulation. Par la suite, nous avons détaillé les étapes de notre méthodologie basée sur l'approche MDSEA en introduisant les deux types d'exigences fonctionnelles et non-fonctionnelles.

Dans un deuxième temps, nous avons présenté les différents enrichissements ajoutés à l'outil eBPMN pour l'adapter à la méthodologie que nous proposons. Nous avons décrit les différentes caractéristiques des différents types de ressource en proposant pour chacun d'eux une grille d'évaluation de leur performance. Par la suite, nous avons présenté notre approche d'agrégation des performances des ressources en y transposant les méthodes de décomposition, de réduction et d'agrégation des processus métiers. Nous avons également avancé notre proposition de représentation de la ressource interface graphique en la considérant comme une ressource composée d'un opérateur humain et IT. Nous pensons que cette technique permet de mieux étudier ses propriétés non-fonctionnelles. Pour finir, nous avons mis l'accent sur l'importance de la prise en considération des chemins lors de l'amélioration des processus. A cet effet nous avons proposé un affichage de résultat de simulation graduel, en les affichant du plus global (comme l'indique l'option 1 de la Figure 37) au plus détaillé (comme l'indique l'option 2 et 3 de la Figure 37).

Chapitre IV. CONTRIBUTION OPERATIONNELLE

I. Implémentation des contributions dans eBPMN	80
1. Outils d'implémentation d'eBPMN	80
2. Implémentation de la distinction des ressources	81
3. Implémentation de la ressource interface graphique.....	83
4. Implémentation des options d'affichage des résultats de simulation.....	84
5. Agrégation des performances des ressources	86
II. Cas d'étude	88
III. Conclusion	98

Ce chapitre fait suite à la contribution conceptuelle, présentée dans la troisième partie de ce manuscrit, en proposant l'implémentation des approches proposées dans l'outil eBPMN pour l'adapter à la méthodologie proposée. Par la suite un cas d'étude est présenté en se basant sur un workflow développé chez Exakis Nelite, il illustre la méthodologie caractérisée par les ressources et les améliorations apportées à l'outil eBPMN.

I. Implémentation des contributions dans eBPMN

1. Outils d'implémentation d'eBPMN

Pour pouvoir modéliser et simuler les processus avec le même langage, l'implémentation d'eBPMN se base sur les techniques de transformation de l'ingénierie dirigée par les modèles. Ceci permet de générer automatiquement le code de simulation à partir du modèle BPMN. La figure représente le déroulement de la simulation avec l'outil eBPMN (Figure 38).



Figure 38: Déroulement de la simulation de l'outil eBPMN

- L'utilisateur modélise la partie fonctionnelle du processus en utilisant le langage BPMN, ce qui permet d'obtenir un modèle BPMN
- L'utilisateur enrichi par la suite le modèle BPMN avec les propriétés non-fonctionnelles en utilisant les annotations textuelles, ce qui permet d'obtenir un modèle BPMN annoté avec les besoins non-fonctionnels
- Le modèle BPMN annoté est transformé automatiquement en modèle PyBPMN (transformation d'un modèle à un autre modèle)
- Le modèle PyBPMN est transformé en code de simulation eBPMN
- L'utilisateur exécute le code de simulation pour obtenir les résultats du scénario joué

« Eclipse Modeling Framework » (EMF) est utilisé pour l'implémentation technique de l'outil eBPMN en se basant sur les principes de l'approche MDA. Nous décrivons ci-dessous les outils utilisés de la plateforme.

EMF est un Framework open source de la plateforme Eclipse, il permet la modélisation et la génération de code des applications basées sur des modèles de données structurées. Le modèle peut être défini sous forme de diagramme de classe, ou sous forme d'un modèle Ecore (un méta-modèle d'EMF pour définir des modèles), ou décrit en utilisant le langage XMI. EMF fournit les outils nécessaires pour transformer le modèle en classes JAVA qui peuvent être utilisées par la suite pour implémenter l'application souhaitée.

BPMN2 Modeler est un plugin Eclipse soutenu par RedHat, un des leaders de logiciel open source les plus reconnues. Il est sous forme d'un outil graphique qui permet de créer et d'éditer des modèles conformes à la norme OMG BPMN 2.0. L'outil fournit également un mécanisme d'extensibilité pour étendre le langage BPMN avec des attributs supplémentaires en conservant son noyau original.

Query View Transform (QVT) est un standard défini par l'OMG pour l'exécution des transformations d'un modèle à un autre. Il est représenté par deux langages de transformation. QVT Relation (QVT-R) un langage impératif et QVT Operational (QVTo) un langage déclaratif. Le langage QVTo est celui utilisé dans l'outil eBPMN et utilise un engin de transformation des modèles Ecore. Il nécessite la spécification des méta-modèles Ecore source et cible. La première partie du standard est nommée « Query » car des requêtes peuvent être appliquées à un modèle source, la deuxième partie est nommée « View » car elle offre la

possibilité de décrire à quoi devrait ressembler le modèle cible et la troisième partie est nommée « Transform » car c'est la partie où les résultats des requêtes sont projetés sur la vue en créant ainsi le modèle cible. Le langage QVTo est basé sur l'Object Constraint Language (OCL).

Acceleo est un générateur de code source du projet Eclipse qui permet aux utilisateurs de créer des applications basées sur un modèle. Il implémente les standards de transformation d'un modèle à un texte de l'OMG. Il fournit des outils pour la génération de code à partir de modèles Ecore, tout en donnant au développeur la possibilité de le modifier. La génération peut se faire en tout langage de programmation textuel, dans notre cas, il s'agit du langage JAVA.

2. Implémentation de la distinction des ressources

Comme nous l'avons mentionné dans le chapitre précédent, dans eBPMN, la ressource peut être sémantiquement humaine ou matérielle, mais aucune distinction syntaxique n'est proposée pour les représenter. La même syntaxe textuelle représente les deux types de ressource. Nous proposons dans notre travail de les distinguer. Dans cette section nous allons présenter brièvement la partie du modèle PyBPMN utilisée dans le langage eBPMN. Les références (Paglia, 2017; Bocciarelli et al., 2016; Bocciarelli and D'Ambrogio, 2011) détaillent plus explicitement les composants du méta-modèle PyBPMN.

2.1. Modification au niveau du méta-modèle PyBPMN

PyBPMN propose trois composants pour représenter les ressources présentés Figure 39 :

- La classe « PyBaseResource » représente la ressource de manière générale et est spécialisé en trois classes « PyPerformer », « PyBroker », « PySubsystem »
- La classe « Pyperformer » représente une ressource atomique quelle que soit sa nature
- La classe « PySubSystem » représente une ressource composée c'est-à-dire le groupe de ressource nécessaire pour mener à bien une tâche
- La classe « PyBroker » représente un groupe de ressources alternatives qui peuvent mener à bien une tâche

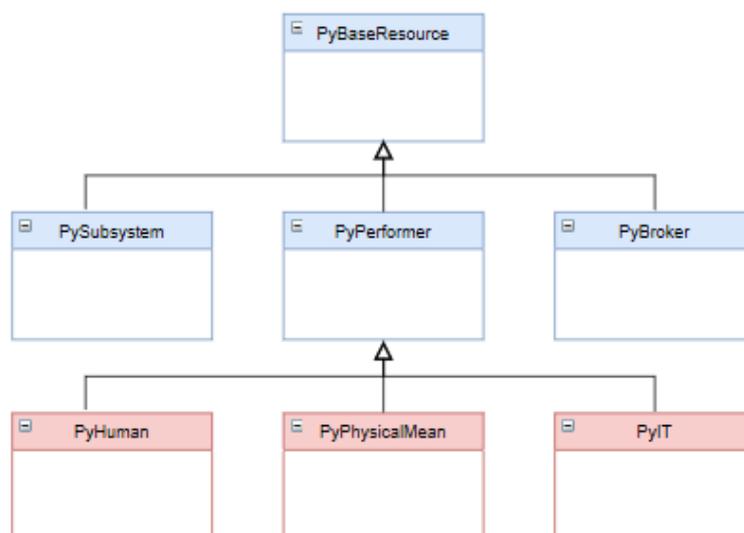


Figure 39: Ajout des classes de distinction des types de ressources dans PyBPMN

Pour pouvoir implémenter notre approche, nous avons codé avec le langage JAVA et nous avons ajouté des nouvelles classes au méta-modèle pour pouvoir distinguer les différents types de ressource comme le recommande l'approche MDSEA. Trois classes qui héritent de la classe « PyPerformer » ont été ajoutées : « PyHuman », « PyPhysicalMeans » et « PyIT ».

2.2.Modification au niveau du langage eBPMN

Comme nous l'avons présenté dans la partie précédente, eBPMN utilise les annotations textuelles du langage BPMN 2.0 pour renseigner les propriétés non-fonctionnelles, selon une syntaxe formelle conforme à EBNF (Scowen, 1998). Nous nous sommes servis du même schéma et de la même construction de la syntaxe proposée pour distinguer les types de ressource. Dans l'exemple ci-dessous, la description de la syntaxe utilisée est détaillée et illustrée par la suite dans la Figure 40.

Ci-dessous, la syntaxe d'une ressource humaine atomique nommée « ResponsableOfMethods » avec un temps de service de 100 secondes renseigné dans la composition des ressources « PySubsystem » que nous décrivons dans le point suivant. La même syntaxe est utilisée pour les autres types de ressource.

```
<<PyHuman>> {name = ResponsableOfMethods serviceTime= (value=1, unit=s)}
```

Pour définir une ressource nommée « TaskResources » composée d'une ressource IT « TechnologyA », d'une ressource matérielle « MachineA » et d'une ressource humaine « ResponsableOfMethods » avec leur temps de service, on utilise la syntaxe suivante :

```
<<PySubSystem>>{name= TaskResources components= ResponsableOfMethods [100], TechnologyA[10], MachineA[100]}
```

L'assignation de la tâche avec la ressource composée « TaskResources » se fait en l'associant à l'élément de l'annotation textuelle en utilisant la syntaxe suivante :

```
<<PyDescriptor>> {resources = (TaskResources)}
```

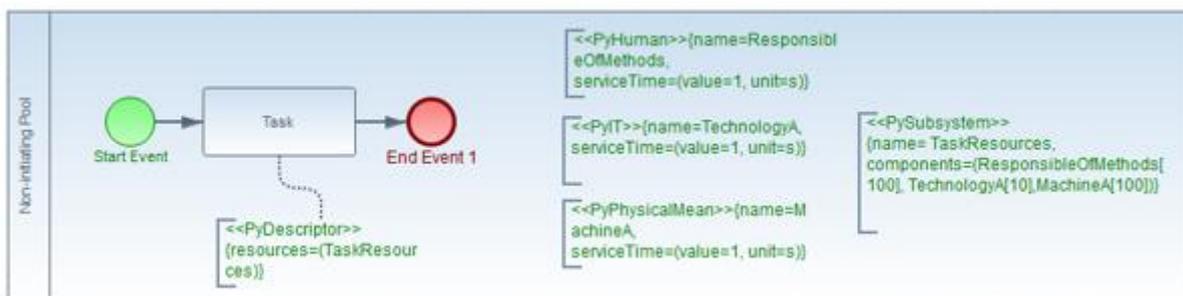


Figure 40: Exemple d'utilisation des syntaxes de distinction des ressources

3. Implémentation de la ressource interface graphique

3.1. Modification au niveau du méta-modèle PyBPMN

Pour prendre en considération la simulation des propriétés non-fonctionnelles des différents composants d'une interface graphique, nous proposons d'ajouter une relation d'héritage entre la classe « PyInterface » et la classe « PyIT ». Puisqu'une interface graphique est composée de plusieurs champs, une relation de composition lie la classe « PyInterface » et la classe « PyField » (Figure 41).

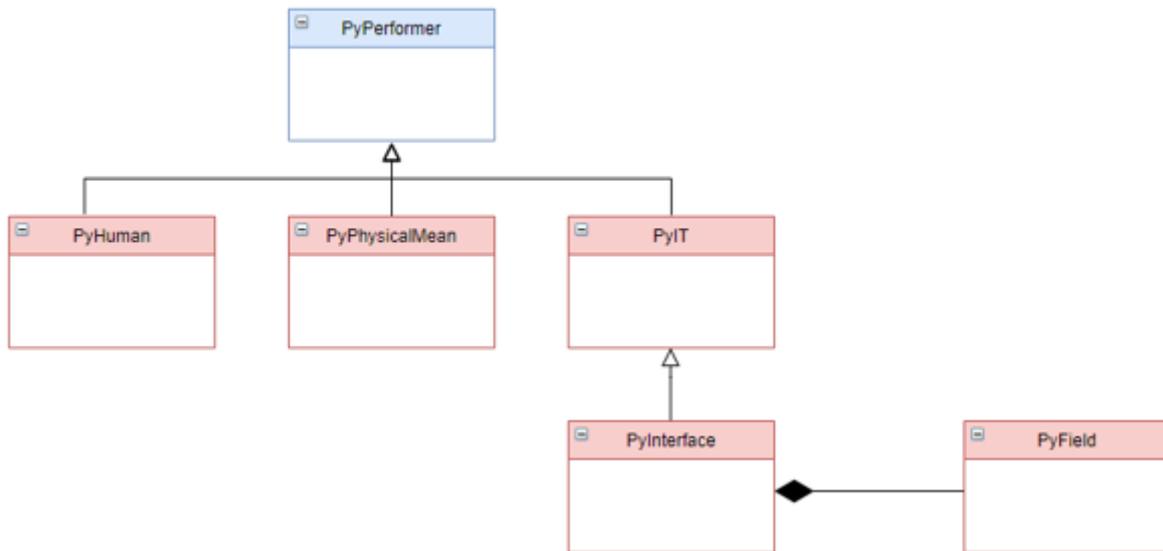


Figure 41: Extension du méta-modèle PyBPMN avec la classe PyInterface

3.2. Modification au niveau du langage eBPMN

Ci-dessous un exemple utilisant la syntaxe proposée pour la ressource interface graphique, en détaillant sa structure et ses composantes. On reprend l'exemple du formulaire de tâche de la Figure 33:

- Nous utilisons la syntaxe ci-dessous pour représenter le champ « Description » où le temps de remplissage de l'utilisateur est de 5 minutes. Le temps de remplissage est indiqué au niveau de la composition de la ressource comme nous le mentionnons dans le point suivant

```
<<PyField>>{name=Description serviceTime=(value=1, unit=min)}
```

- Nous utilisons la syntaxe ci-dessous pour définir la ressource composée représentant l'interface graphique avec le temps de remplissage de chaque champ

```
<<PyInterface>>{name=UserInterface,fields=(Name[3],Department[2],RequestData[2],Description[5])}
```

- L'assignation de la tâche du langage BPMN avec la ressource composée « UserInterface » se fait en l'associant à l'élément de l'annotation textuelle en utilisant la syntaxe ci-dessous

```

<<PyDescriptor>>{resources=(User_Interface)}
```

La Figure 42 illustre la syntaxe que nous proposons pour définir une interface graphique

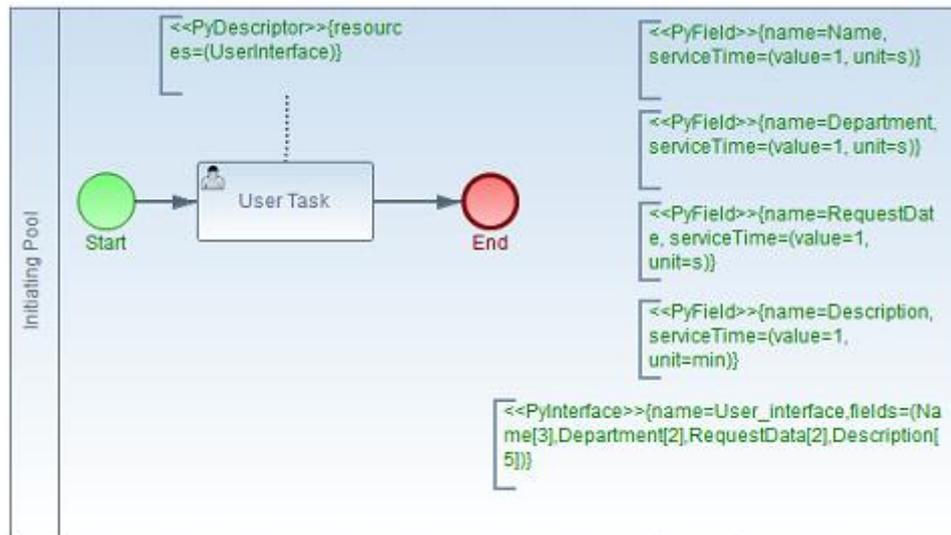


Figure 42: Exemple d'utilisation de PyInterface

4. Implémentation des options d'affichage des résultats de simulation

L'outil eBPMN affiche les résultats du processus globalement sans indiquer les chemins parcourus pendant la simulation et se base uniquement sur l'assignation des probabilités des chemins. A cet effet, pour les choix des chemins à simuler nous avons ajouté la possibilité de choisir explicitement celui que l'utilisateur souhaite analyser en indiquant la décision prise par la ressource. Pour ce faire, nous avons ajouté l'attribut « userchoise » à la classe « PyDescriptor » du méta-modèle PyBPMN qui permet, comme indiqué dans les illustrations précédentes, de lier l'élément BPMN aux ressources qui lui sont assignées. La syntaxe est décrite ci-dessous et la Figure 43 représente son illustration.

```

<<PyDescriptor>> {resources=(Validator[720]) userChoice=Validated}
```

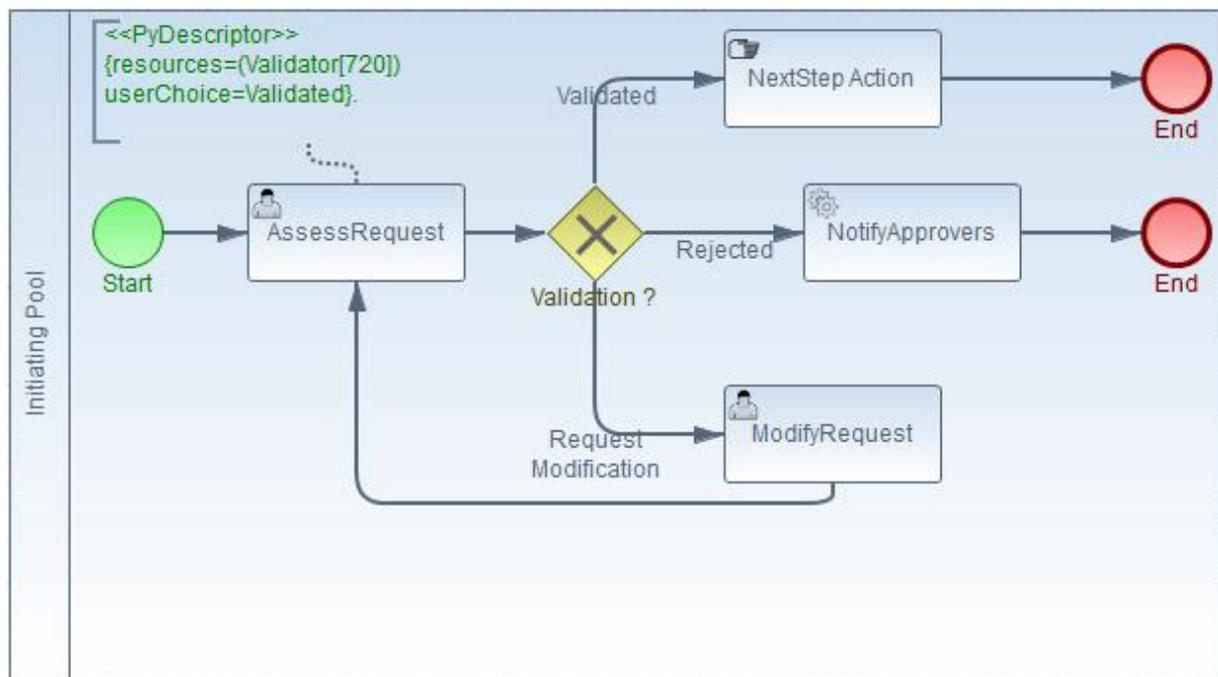


Figure 43: Exemple de choix du chemin à simuler en indiquant la décision de l'utilisateur

Concernant l'ajout des options d'affichage du plus global au plus détaillé, eBPMN utilise le concept des jetons « token » pour la simulation du modèle BPMN. Le jeton illustre le comportement d'exécution de chaque élément du langage. Il parcourt le modèle du point de départ jusqu'au point final, donc, un jeton peut être considéré comme un cas d'utilisation du processus. De ce fait, nous utilisons le jeton pour pouvoir afficher les chemins parcourus pendant la simulation.

Les entités utilisées pour la simulation du langage eBPMN sont catégorisées en deux types :

- Les objets eBPMN correspondant aux éléments originaux de BPMN
- Les objets eBPMN correspondant aux éléments du méta-modèle PyBPMN

Dans notre travail, nous utilisons le premier type pour afficher les différents chemins par lesquels le jeton est passé pour évaluer leur atteignabilité, puis, nous utilisons le second type pour afficher les mesures des propriétés non-fonctionnelles du chemin parcouru pendant la simulation. Nous avons également ajouté la classe « Path » au diagramme de classe du langage eBPMN. Nous décrivons brièvement le diagramme dans cette section, la description détaillée se retrouve dans les références suivantes (Paglia, 2017; Bocciarelli et al., 2014) :

- La classe « EBPMNELEMENT » représente la classe générale représentant les entités de eBPMN. Elle est spécialisée en deux autres classes « FlowNode » et « EBPMNResource »
- La classe « EBPMNResource » représente les entités correspondant au comportement d'exécution des ressources selon le méta-modèle de PyBPMN comme illustré dans la Figure 41
- La classe « FlowNode » représente les entités représentant le comportement d'exécution des éléments BPMN

Pour l'implémentation de notre approche, puisqu'un chemin est composé principalement des éléments de BPMN, nous avons ajouté une relation de composition entre la classe « FlowNode » et « Path » comme le montre la Figure 44.

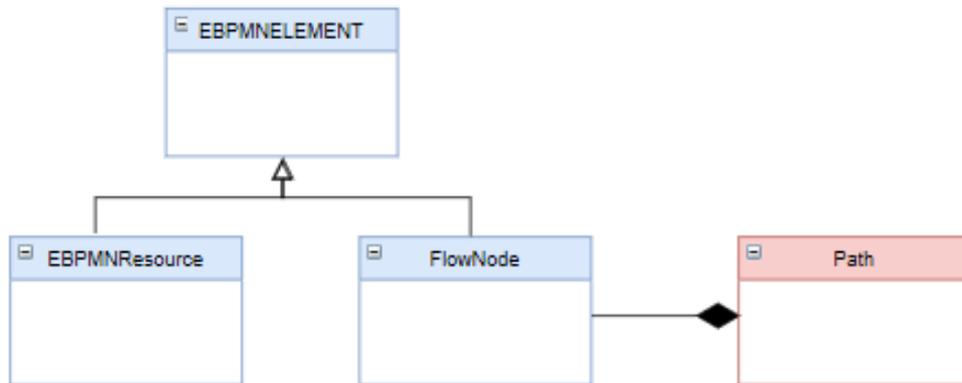


Figure 44: Extension d'eBPMN avec la classe « Path »

5. Agrégation des performances des ressources

5.1. Modification au niveau du méta-modèle PyBPMN

Pour implémenter l'agrégation des ressources nous avons ajouté un attribut « composition » à la classe « PySubsystem » du méta-modèle PyBPMN qui peut avoir une des valeurs suivantes : « seq » pour la structure séquentielle, « and » pour la structure parallèle et « or » pour la structure alternative. Si l'utilisateur n'utilise pas cet attribut la structure est considérée séquentielle comme le propose nativement le logiciel eBPMN.

5.2. Modification au niveau du langage eBPMN

Concernant la syntaxe d'agrégation, on construit la ressource composée comme nous l'avons expliquée dans les parties précédentes et nous ajoutant l'attribut « composition » comme le montre l'illustration de la Figure 45.

```

<<PySubsystem>>{name=RAssessStore, components=(Responsible,
UserInterface), composition=and}
    
```

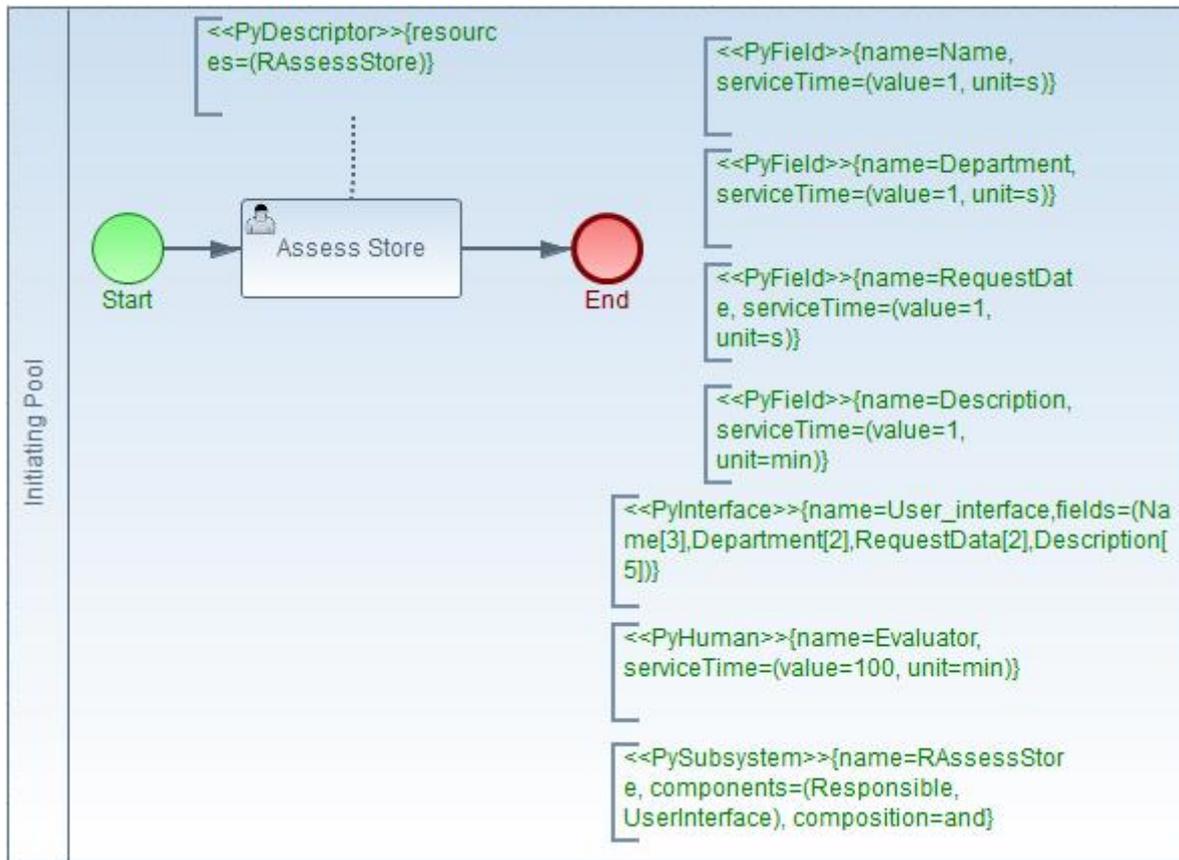


Figure 45: Exemple d'utilisation de syntaxe d'agrégation de ressource

Comme nous l'avons expliqué dans la partie précédente, dans le diagramme de classe d'eBPMN, la classe « EBPMNResource » représente le comportement non-fonctionnel d'une ressource. Puisque nous souhaitons agréger les performances des ressources, nous avons ajouté une relation d'agrégation entre la classe « Performer » et « PerformanceAgregation ».

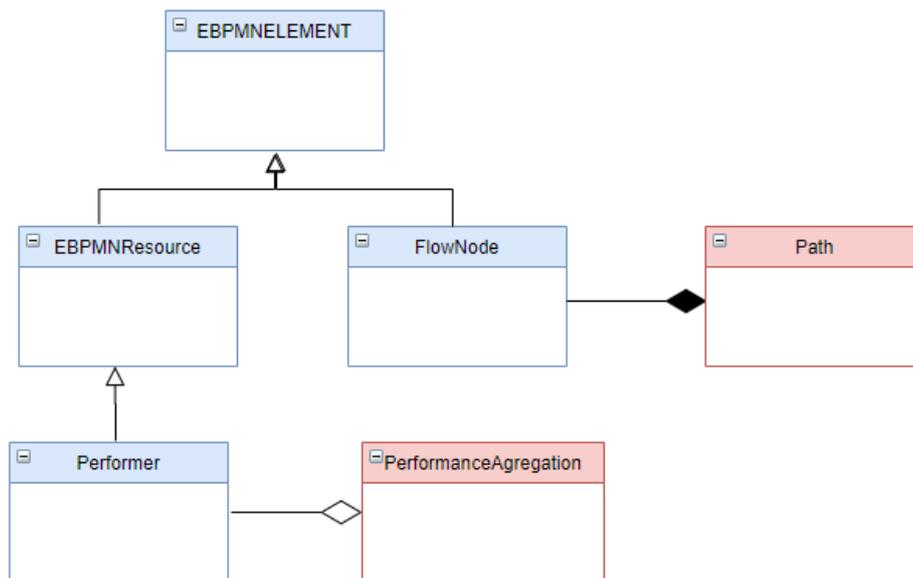


Figure 46: Extension d'eBPMN avec la classe « PerformanceAgregation »

Quand le choix d’affichage est global, donc dans le cas de l’agrégation des performances des ressources, au niveau du code pré-exécution, nous avons implémenté les instructions suivantes :

- Agréger les performances en appliquant les règles du Tableau 11 adéquates à l’attribut composition renseigné
- Créer une nouvelle ressource en lui attribuant l’agrégation des propriétés non-fonctionnelles
- Attribuer à la ressource agrégée les ressources la composant
- Au niveau du code post-exécution, nous avons implémenté les instructions suivantes dans le cas où la ressource doit accomplir la tâche assignée :
 - Activer l’état d’exécution de la ressource agrégée
 - Mettre les ressources la composant en état actif
 - Envoyer la ressource agrégée en exécution
 - Mettre les ressources la composant en état inactif après la fin d’exécution de la ressource agrégée

Dans le cas où l’affichage des résultats choisis n’est pas global, nous avons modifié le fonctionnement d’attribution des états d’exécution en fonction de la structure d’agencement choisie, puisque dans eBPMN, les ressources composées sont simulées de façon séquentielle uniquement.

Composition séquentielle :

- Activer l’état d’exécution de la ressource composée
- Mettre les ressources la composant en état actif
- Envoyer une ressource en exécution
- Mettre la ressource en état inactif après la fin d’exécution
- Répéter l’étape 3 et 4 jusqu’à la finalisation d’exécution de toute les ressources
- Appliquer les règles du Tableau 11 pour le calcul des propriétés non-fonctionnelles post-exécution

Composition en parallèle :

- Activer l’état d’exécution de la ressource composée
- Mettre les ressources la composant en état actif
- Envoyer toutes les ressources en exécution
- Mettre la ressource en état inactif après la fin d’exécution
- Répéter l’étape 4 jusqu’à la finalisation d’exécution de toutes les ressources
- Appliquer les règles du Tableau 11 pour le calcul des propriétés non-fonctionnelles post-exécution

II. Cas d’étude

Cette section vise à illustrer la méthodologie proposée et le rendu des enrichissements que nous avons apportés à l’outil eBPMN : distinction des ressources, agrégation des performances, ressource composée représentant l’interface graphique et l’affichage des résultats de simulation

avec le chemin parcouru. Par souci de simplicité, nous avons choisis d'utiliser une partie du processus d'un des clients d'EXAKIS NELITE et nous présenterons deux scénarios de deux profils de ressources humaines : junior et expérimenté.

Afin de préserver la confidentialité du client d'EXAKIS NELITE, nous allons dans cet exemple changer la nature fonctionnelle du processus. Le client est une chaîne de restauration rapide qui vend principalement des sandwichs et des salades et se développe constamment en France. C'est la raison pour laquelle ils veulent automatiser leur processus d'ouverture et d'évaluation de franchise.

Après la réception du cahier de charge, la méthodologie que nous proposons s'illustre comme suit :

Phase TOP BSM

Cette phase doit être effectuée par toutes les parties prenantes liées au processus pour clarifier les besoins fonctionnels des besoins non-fonctionnels.

- **Besoins fonctionnels** : Afin de gérer leurs franchises (ouverture, maintenance et fermeture), le client souhaite utiliser un workflow pour automatiser l'évaluation des méthodes et pratiques utilisées dans ses restaurants. L'évaluation est effectuée par un évaluateur et est basée sur des critères qualitatifs. La demande est ensuite soumise pour validation à un circuit de validation. L'outil à mettre en place doit répondre aux besoins suivants :
 - ✓ Offrir une interface graphique qui regroupe les critères qualitatifs de l'évaluation
 - ✓ Soumettre la demande d'évaluation au circuit de validation adéquat à la nature de la demande en prenant en considération : la nature de l'opération (ouverture, maintenance, fermeture) et la ville où se trouve la franchise
- **Besoins non-fonctionnels** : leurs principaux objectifs sont les suivants :
 - ✓ Augmenter l'efficacité interne de la gestion des demandes d'ouverture et de fermeture
 - ✓ Pouvoir traiter plus de demandes tout en améliorant les délais de réponse et de décision car : la fermeture tardive d'un restaurant non conforme à leurs critères peut nuire à leur image et l'ouverture tardive d'une potentielle franchise pourrait diminuer leur chiffre d'affaire

Ils reçoivent en moyenne cinq demandes par semaine, afin d'atteindre leurs objectifs, leur but est de pouvoir traiter deux demandes par semaines.

Les ressources impliquées dans ce processus sont :

- Ressources humaines : Evalueur, responsable méthode, responsable qualité, assistante juridique, responsable juridique, manager des franchises
- Ressources matérielles : PC portables, tablettes et téléphone, environnement cloud
- Ressources IT : Editeur A, Editeur B

La grille de performance des évaluateurs :

	Disponibilité		Compétence				Performance				Résultats
	Charge de travail		Savoir-faire métier		Savoir-faire informatique		Rapidité d'exécution		Adéquation de la tâche		
	Score	Poids	Score	Poids	Score	Poids	Score	Poids	Score	Poids	
Evaluateur expérimenté	1	2	2	2	2	1	2	4	2	2	1.82
Evaluateur junior	1	2	0	2	1	1	0	4	2	2	0.64

Tableau 12: Grille de performance des évaluateurs

Après la clarification des différents types de besoin et des ressources impliquées, nous passons à l'étape suivante.

Phase BOTTOM BSM

Cette étape permet de modéliser formellement le processus à automatiser à un haut niveau d'abstraction en utilisant le langage BPMN 2.0. Comme le montre la Figure 47, les ressources sont distinguées en utilisant les différents types de tâche fournis par le langage.

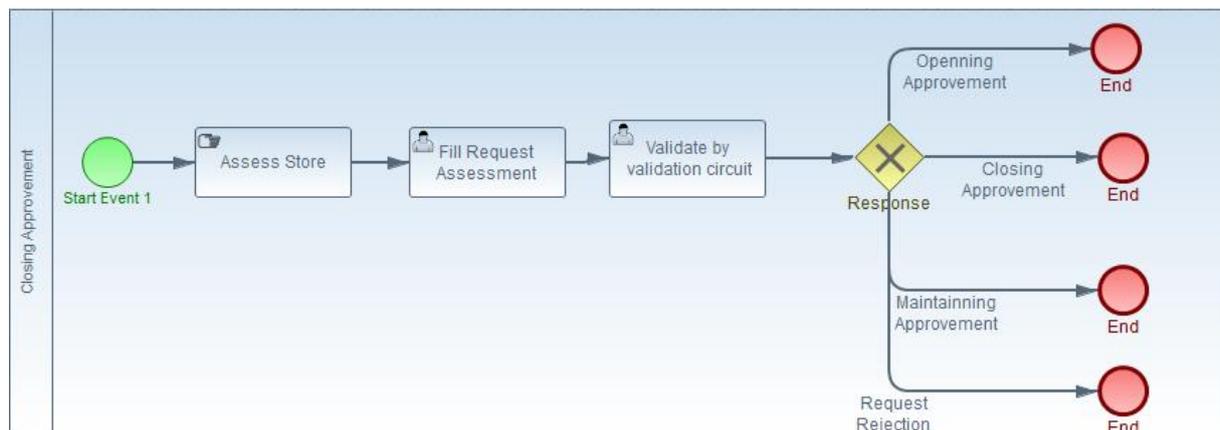


Figure 47: Modélisation du processus à un haut niveau d'abstraction

Phase TOP TIM

La première étape de cette phase est de modéliser en détail les besoins fonctionnels du processus en utilisant le langage BPMN pour décrire les activités et leur flux de séquence (Figure 49). La seconde étape consiste à vérifier les chemins à analyser pour s'assurer qu'ils sont atteignables et correspondent aux besoins fonctionnels. Comme le montre la Figure 48, le premier chemin arrive à la fin du processus, le jeton a pu atteindre l'état final contrairement au deuxième chemin. Avant de passer à la phase suivante, il est important de s'assurer que tous chemins à analyser sont accessibles.



Figure 48: Atteignabilité des chemins

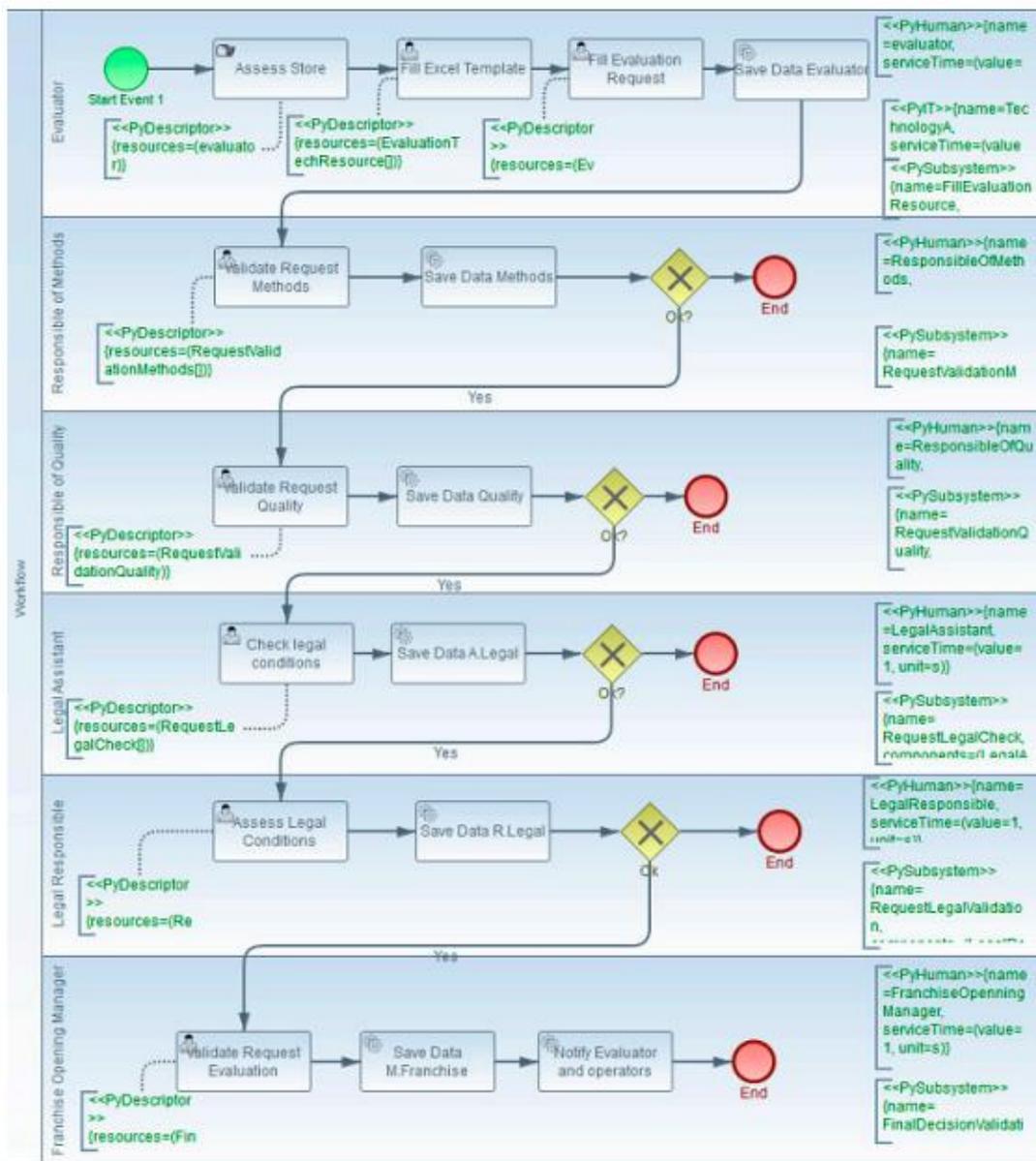


Figure 49: Modélisation détaillée du processus

Phase BOTTOM TIM

La première étape de cette phase est de déterminer les valeurs des propriétés non-fonctionnelles pour chaque tâche, en utilisant la grille de performance remplie pendant la phase « TOP BSM ». Le Tableau 63 illustre l'exemple du temps de traitement des deux profils d'évaluateur.

Tâche	Temps moyen d'exécution	E. Expérimenté	E. Junior
Evaluation du restaurant	720	396	1125
Remplir fichier Excel	61	34	95
Remplir la demande d'évaluation	31	17	48

Tableau 63: Détermination des mesures non-fonctionnelles des tâches de l'évaluateur

La seconde étape de cette phase est d'associer pour chaque tâche, ses ressources, leur structure d'agencement et leurs mesures non-fonctionnelles. Pour ce workflow la structure de toutes les ressources composées est en parallèle (Figure 50).

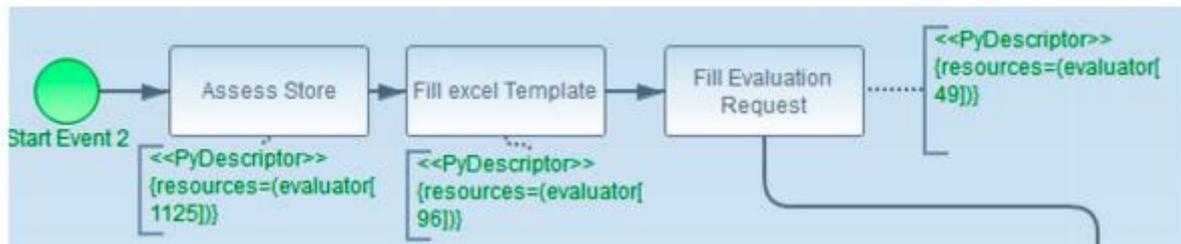


Figure 50: Tâches associées aux mesures non-fonctionnelles

Comme énoncé précédemment, on a trois possibilités d'affichage des résultats :

- Obtenir un résultat de simulation global pour chaque chemin parcouru par le jeton en utilisant l'agrégation des performances des ressources
- Obtenir un résultat de simulation global pour chaque type de ressource et pour chaque chemin parcouru
- Obtenir un résultat de simulation pour chaque ressource et chaque tâche

Nous allons montrer dans cet exemple les résultats des deux premières options puisqu'elles représentent notre proposition d'amélioration d'affichage de l'outil eBPMN. Le chemin choisi est le chemin de validation de l'évaluation. Les scénarios joués sont ceux des ressources juniors et expérimentées. La durée de simulation est de trente jours avec cinq demandes lancées par semaine.

➤ Résultat global de la simulation

Nous avons choisi pour la première étape de la simulation non-fonctionnelle de jouer le scénario des ressources juniors. Cette démarche va permettre de pousser la réflexion sur les tâches inutiles qui peuvent être supprimées.

Comme le montre la Figure 51, dans ce scénario, la demande est traitée en 10 jours ce qui est loin du but à atteindre. Ainsi, après investigation, il s'est avéré que les tâches « Vérifier les

conditions légales » et « Evaluer les conditions légales » sont presque similaires. De plus, le service juridique ne contient que trois personnes. Puisque l’option d’augmenter le nombre de ressources n’est pas envisageable, il a été décidé de supprimer la tâche « Vérifier les conditions légales ». Ainsi, les résultats de simulation de la nouvelle configuration montrent qu’une demande est traitée en 8 jours (Figure 52).

```

=====
                                PATH STATISTICS: RESOURCE AGGREGATION
=====
### Path..... :

Start Event 1==>Assess Store==>Fill Excel Template==>Fill Evaluation
Request==>Save Data Evaluation==>Validate Request Methods==>Save Data
Methods==>Validate Request Quality==>Save Data Quality==>Check legal
conditions==>Save DataA Legal==>Assess Legal Conditions==>Save DataR
Legal==>Validate Request Evaluation==>Save DataM Franchise==>Notify Evaluator and
operators==>End
Mean Service Time .....: 8780.753 min
Mean Waiting Time .....: 5639.485 min
=====

```

Figure 51: Résultat global en utilisant l’agrégation des performances des ressources

Plusieurs scénarios de différentes configurations peuvent être joués jusqu’à l’atteinte de la non-possibilité de supprimer d’autres activités. Il est important de vérifier l’accessibilité de la nouvelle configuration à étudier avant de lancer la simulation de ses propriétés non-fonctionnelles.

```

=====
                                PATH STATISTICS: RESOURCE AGGREGATION
=====
### Path..... :

Start Event 1==>Assess Store==>Fill Excel Template==>Fill Evaluation
Request==>Save Data Evaluation==>Validate Request Methods==>Save Data
Methods==>Validate Request Quality==>Save Data Quality==>Assess Legal
Conditions==>Save DataR Legal==>Validate Request Evaluation==>Save DataM
Franchise==>Notify Evaluator and operators==>End
Mean Service Time .....: 4130.492 min
Mean Waiting Time .....: 7471.699 min
=====

```

Figure 52: Résultat global de la simulation des ressources juniors après la suppression des tâches inutiles

Après la suppression des étapes inutiles, la simulation du meilleur scénario en termes de ressource humaine (ressources expérimentées) a été effectué. Comme le montre la Figure 53, les résultats montrent que la demande est traitée en 6 jours.

Il est important de mentionner que le temps de traitement est le maximum des durées des ressources en parallèle comme mentionné précédemment. Ce qui prouve l’importance de la prise en considération de la structure d’agencement des ressources dans le cas d’une ressource composée. Pour cet exemple, si on suppose que la simulation ne prend que le cas de la structure séquentielle, comme le propose l’outil eBPMN avant nos modifications, le double du résultat sera obtenu, et cela s’applique pour les autres configurations aussi. Ceci démontre l’importance de prendre en considération la configuration des ressources pendant la phase de simulation.

```

=====
                                PATH STATISTICS: RESOURCE AGGREGATION
=====
### Path..... :

Start Event 1=>Assess Store=>Fill Excel Template=>Fill Evaluation
Request=>Save Data Evaluation=>Validate Request Methods=>Save Data
Methods=>Validate Request Quality=>Save Data Quality=>Assess Legal
Conditions=>Save DataR Legal=>Validate Request Evaluation=>Save DataM
Franchise=>Notify Evaluator and operators=>End
Mean Service Time .....: 2742.330 min
Mean Waiting Time .....: 5635.648 min
=====

```

Figure 53: Résultat de simulation global pour les ressources humaines expérimentées

Pour mieux orienter l’investigation après avoir joué le meilleur scénario des ressources humaines, l’option de l’affichage des résultats en fonction de chaque type de ressource a été lancée.

➤ Résultat global pour chaque type de ressource

Pour cette étape, la simulation a été lancée en utilisant la configuration où les tâches inutiles sont supprimées. Les deux scénarios des ressources humaines (juniores et expérimentées) ont été lancés pour pouvoir analyser quel type de ressource a besoin d’amélioration.

```

=====
                                PATH STATISTICS: RESOURCE TYPE DETAILS
=====

### Path..... :

Start Event 1=>Assess Store=>Fill Excel Template=>Fill Evaluation
Request=>Save Data Evaluation=>Validate Request Methods=>Save Data
Methods=>Validate Request Quality=>Save Data Quality=>Assess Legal
Conditions=>Save DataR Legal=>Validate Request Evaluation=>Save DataM
Franchise=>Notify Evaluator and operators=>End
Mean Service Time .....: 8800.168 min
Mean Waiting Time .....: 5700.890 min
=====

### Resources Path..... :

Type Resources..... :

Human Resources:
*****
Mean Service Time .....: 7900.133 min
Mean Waiting Time .....: 4000.226 min
IT Resources:
*****
Mean Service Time .....: 8800.168 min
Mean Waiting Time .....: 5700.890 min
=====

```

Figure 54: Résultat de chaque type de ressource pour le cas des ressources humaines juniores

```

=====
                                PATH STATISTICS: RESOURCE TYPE DETAILS
=====

##### Path..... :
Start Event 1==>Assess Store==>Fill Excel Template==>Fill Evaluation
Request==>Save Data Evaluation==>Validate Request Methods==>Save Data
Methods==>Validate Request Quality==>Save Data Quality==>Assess Legal
Conditions==>Save DataR Legal==>Validate Request Evaluation==>Save DataM
Franchise==>Notify Evaluator and operators==>End
Mean Service Time .....: 2680.168 min
Mean Waiting Time .....: 5300.890 min
=====
##### Resources Path..... :

Type Resources..... :

Human Resources:
*****
Mean Service Time .....: 2000.48 min
Mean Waiting Time .....: 5200.226 min
IT Resources:
*****
Mean Service Time .....: 2680.168 min
Mean Waiting Time .....: 5300.890 min
=====

```

Figure 55: Résultat de chaque type de ressource pour le cas des ressources humaines expérimentées

Comme le montrent la Figure 52 et la Figure 53, les ressources IT sont celles qui ont un temps de service élevé. De ce fait, il est plus judicieux d’investiguer sur l’amélioration des ressources IT en premier, surtout dans ce cas où les ressources fonctionnent en parallèle.

Distinguer les résultats de chaque type de ressource permet d’obtenir plus d’informations sur leur implication et leur performance vis-à-vis du processus, ce qui facilite la prise de décision.

➤ Simulation des propriétés non-fonctionnelles des interfaces graphiques

Puisqu’il s’agit d’un processus de service, les interfaces graphiques sont les ressources IT dominantes. De ce fait, une simulation de granularité plus fine peut être effectuée pour évaluer l’efficacité des formulaires de tâche. Après la première tentative d’exécution de la simulation en utilisant l’exemple de l’interface comme le montre la Figure 56, la durée de traitement de service de la ressource composée du responsable juridique et de l’interface graphique qu’il utilise est de 3544 minutes. Les résultats obtenus ont permis d’investiguer sur les champs qui peuvent être améliorés par leur automatisation ou en fournissant des aides de remplissage. Ces actions ont permis d’optimiser 1714 minutes comme le montre la Figure 57.

Le Tableau 74 présente les quelques améliorations effectuées sur l’interface graphique.

Champ	Temps de remplissage	Description	Amélioration	Temps de remplissage
Responsable juridique (Champ de texte)	15 sec	Initialement, le champ est alimenté manuellement, le remplir automatiquement fait gagner du temps	Remplissage automatique	0 sec
Type d'évaluation (Champ de texte)	15 sec	Le champ est rempli manuellement, l'utilisation des choix est un gain de temps et évite les erreurs de saisies	Champ choix	6 sec
Type de point de vente (Champ de texte)	20 sec	Le champ est rempli manuellement, l'utilisation des choix est un gain de temps et évite les erreurs de saisies	Champ choix	6 sec
Région (Champ choix)	7 sec			
Ville (Champ choix)	20 sec	Filtrer le champ de la ville en fonction de la région choisie permettrait de gagner du temps	Champ choix filtré par région	10 sec
Evaluation de l'analyse	3520 min	<p>Pour évaluer la demande, l'utilisateur doit comparer cette demande avec les anciennes similaires demandes, ainsi que la vérification des règles juridiques ayant les mêmes critères. Pour ce faire, il doit chercher ces informations manuellement dans le système, ce qui peut prendre du temps, surtout quand il s'agit d'une demande complexe.</p> <p>Pour améliorer ce cas, il est possible de faire faciliter l'accès à ces informations préalablement filtrées en fonction des critères de la demande dans le formulaire de tâche. Ce qui permet à l'utilisateur de se concentrer sur la partie de l'analyse</p>	Information à porter de main	1800min

Tableau 74: Description et amélioration d'une interface graphique



Figure 56: Résultat de la simulation de la ressource composée interface graphique



Figure 57: Résultat de la simulation de la ressource composée interface graphique après les améliorations

Après l’investigation sur chaque interface utilisée dans le processus, une simulation globale a été effectuée pour évaluer les nouvelles configurations mises en place. Les résultats de la simulation du cas des ressources humaines expérimentées montrent que la demande est traitée en 3 jours ce qui est plus proche de l’objectif à atteindre (Figure 58). Plusieurs configurations peuvent être jouées et évaluées jusqu’à l’obtention de la configuration qui satisfait les exigences requises.

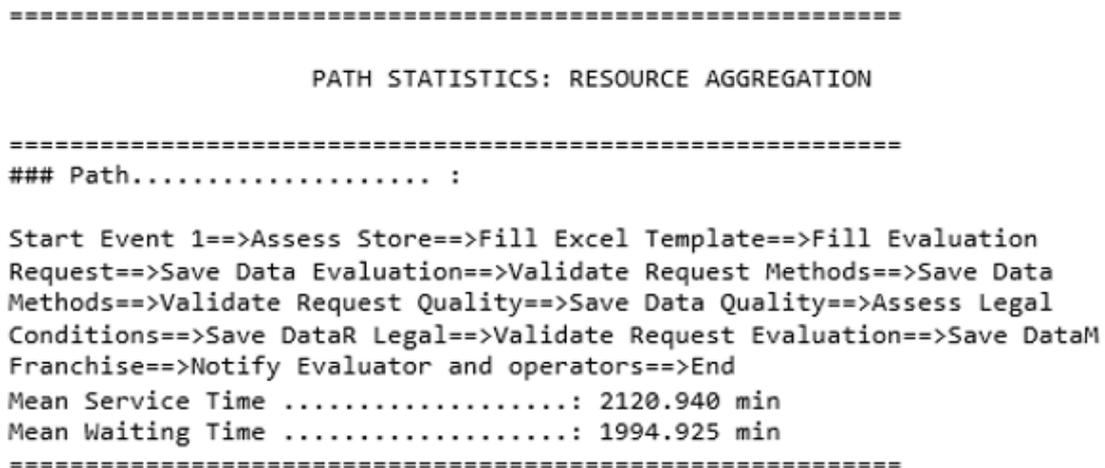


Figure 58: Résultat global du chemin analysé après les améliorations des interfaces graphiques

III. Conclusion

Dans ce chapitre nous avons décrit l'implémentation des différentes approches proposées dans la partie contribution conceptuelle. Les améliorations apportées à l'outil eBPMN peuvent se résumer comme suit :

- Distinction syntaxique des types de ressource
- Indication de la structure d'agencement des ressources composées
- Prise en considération de l'agencement des ressources pendant la simulation
- Affichage des chemins parcourus pendant la simulation pour chaque type d'affichage
- Affichage des résultats de simulation de manière globale en utilisant l'agrégation des performances des ressources
- Affichage des résultats de simulation de manière globale pour chaque type de ressource

Nous avons par la suite présenté un cas d'étude à partir d'un des workflows d'un des clients d'Exakis Nelite. Dans cette partie nous avons illustré un exemple de fonctionnement de la méthodologie d'automatisation des processus par caractérisation des ressources. Nous avons exemplifié comment les différents scénarios orientés ressources permettent d'aider à faire émerger les solutions nécessaires à l'atteinte de l'objectif fixé.

Nous constatons et confirmons dans ce chapitre, par l'application de l'approche sur un cas issu d'un problème métier réel, que plus le projet progresse dans la conception de la solution plus il est nécessaire de modéliser et simuler en détail un nombre important d'informations comme le révèle la représentation de la ressource interface graphique. Notre méthode contribue bien à intégrer dans les modèles et les simulations un maximum de détails descriptifs des ressources contrairement à la majorité des méthodes et outils qui restent à un niveau macro de paramétrage.

CONCLUSION ET PERSPECTIVES

De nos jours, l'automatisation des processus métiers représente un facteur de réussite des entreprises en donnant d'avantage de maîtrise des activités internes de l'entreprise, ceci permettant de mieux réagir aux demandes extérieures et ainsi subsister dans un monde économique compétitif. Par conséquent, assurer l'alignement des besoins fonctionnels et non-fonctionnels avec l'implémentation mise en place est importante, d'autant plus pour une époque où le digital est omniprésent et doit être pleinement exploité. Néanmoins, cette ère de digitalisation n'est pas une solution en soi, car les entreprises font souvent appel aux différents éditeurs et aux différentes SSII en pensant qu'une technologie, seule, résout leurs problèmes et répond à leurs différents objectifs. Cette thèse est donc née du besoin de lier l'innovation technique à une démarche permettant de mieux prendre en considération les acteurs du processus à automatiser, pour proposer des modèles et simulations de processus plus pertinents.

Comme nous l'avons présenté dans le chapitre présentant la problématique, les SSII adoptent différentes méthodes de développement des projets pour mieux maîtriser le périmètre des besoins clients. Cependant, ces méthodologies ont pour fonction première de gérer les différentes étapes d'un projet de développement d'une application dont la conduite est trop souvent dictée par une orientation technique. Cette vision est peut-être suffisante pour des projets de développement d'un intranet ou d'un site d'e-commerce simples dont la performance des ressources n'impacte que peu la conception. Mais pour l'automatisation des processus métiers plus complexes, il est important de s'assurer du bon fonctionnement du processus et de prendre en considération ses différents composants et acteurs avant d'étudier la phase technique de son implémentation.

L'état de l'art que nous avons réalisé démontre l'importance de l'analyse fonctionnelle et non-fonctionnelle avant l'automatisation d'un processus. Plusieurs propositions ont été faites à ce sujet : méthode de gestion des processus BPM, propositions de nouveaux langages et techniques de modélisation, propositions de démarche de simulation, proposition d'enrichissement des langages les plus adoptés pour prendre en considération les différents types de besoin, etc. Or, la majorité de ces propositions adopte un paradigme guidé par les activités pour analyser les processus à automatiser. Alors qu'une application informatique de ces processus, en l'occurrence un workflow, fonctionne toujours dans un système appelant différents types de ressource. De plus, ces mêmes ressources représentent les entités qui accomplissent et exécutent les différentes activités et tâches d'une entreprise. D'où l'importance d'adopter une analyse orientée ressource. Comme nous l'avons présenté dans la conclusion de l'état de l'art, les études proposées pour représenter les ressources peuvent être catégorisées en deux catégories. La première catégorie est une analyse des techniques orientées ressources quant à la deuxième catégorie est une analyse de leurs propriétés non- fonctionnelles. Malgré la pertinence des études proposées pour modéliser et représenter les ressources ainsi que leur gestion, elles ne prennent pas en considération leurs différents types et la différence de leurs caractéristiques.

De ce fait, l'enjeu principal de notre travail de recherche était de proposer une méthodologie d'automatisation des processus par caractérisation des ressources. Le but est d'avoir une approche méthodologique globale, de la définition des besoins jusqu'à la phase des choix d'implémentation, qui permet de prendre en considération l'ensemble des éléments liés au workflow en intégrant les différents types de ressource et leurs différentes spécificités. Pour ce faire, nous nous sommes appuyés sur les notions liées aux processus métiers et des travaux effectués au sein de notre laboratoire. A cet effet, nous avons proposé une méthodologie basée sur l'approche MDSEA qui est à notre connaissance la seule approche qui préconise la prise en considération des trois types de ressource : humain, matériel et IT. Nous avons également utilisé les langages BPMN 2.0 et eBPMN pour modéliser et simuler les processus à automatiser. Le choix du langage BPMN a été motivé par sa standardisation mais ne permet pas de décrire les ressources avec précision, cependant BPMN offre la possibilité d'être étendu. Nous avons toutefois choisi de ne pas le modifier mais plutôt d'utiliser ses annotations de texte pour ajouter un langage de description des ressources et de leurs performances.

Ensuite, le choix du langage eBPMN pour la simulation a été motivé principalement par sa capacité à conserver les avantages du langage BPMN et sa capacité à simuler une ressource composée. Nous avons enrichi le langage eBPMN pour l'adopter à notre méthodologie. A cet effet, nous avons distingué syntaxiquement les différents types de ressource qu'elle soit atomique ou composée et nous avons proposé des grilles de performance pour chaque type. Comme nous l'avons décrit dans le chapitre de la contribution, chaque type de ressource a ses propres caractéristiques pour évaluer son temps de service et sa fiabilité. Par exemple, pour calculer le temps de service d'une ressource humaine on peut prendre en considération son expérience, sa charge de travail, etc. Alors que pour une ressource IT, le débit et le temps de chargement ont plus de sens dans l'évaluation de leur performance. De plus, ces paramètres peuvent avoir un impact négatif ou positif sur le degré d'efficacité d'une ressource. Par exemple, un opérateur moins disponible et moins expérimenté mettra plus de temps pour exécuter une tâche et vis-versa. Cela démontre que la prise en considération des caractéristiques de chaque type de ressource aidera à se poser les bonnes questions sur les décisions à prendre pour améliorer le processus.

Nous avons également proposé une démarche d'agrégation des performances des ressources composées en adoptant les concepts de décomposition, de réduction et d'agrégation de performance des processus métiers. Cela permet aux parties prenantes d'avoir des résultats de simulation globaux pour pouvoir mettre en évidence le chemin critique à prioriser pendant la phase d'analyse. De plus, les règles de calcul des structures en parallèle, séquentielles et alternatives sont différentes. Ainsi, la prise en compte de la configuration d'agencement des ressources qui ont effectuées une tâche permet de fournir des mesures plus précises et plus fiables, ce qui pourrait éviter d'apporter des changements inutiles et vice versa.

Nous avons également proposé la simulation de la ressource composée qui représente une interface graphique qui est composée d'un opérateur humain et d'une ressource IT. L'idée est d'attribuer à chaque champ du formulaire de tâche le temps de son remplissage par un utilisateur. Cette simulation de granularité plus fine permet de mieux concevoir les interfaces graphiques en fournissant uniquement les informations nécessaires pour gagner du temps et

réduire le taux d'erreur. Cette approche est encore plus utile dans le cas des processus administratif où les formulaires de tâche représentent leur noyau principal à l'instar des machines dans un processus de production.

Nous avons également mis l'accent sur l'importance de prendre en considération le chemin parcouru pendant la phase de simulation lors de l'analyse du processus à automatiser. Connaître les interactions des ressources impliquées pendant la simulation facilite l'émergence des idées de scénario à envisager pour son amélioration. De ce fait, nous avons ajouté deux options d'affichage en plus de celle existante dans l'outil eBPMN pour afficher les résultats du plus global au plus détaillé.

Notre travail a des limites, et par conséquent des perspectives sont envisageables. Notre apport se situe principalement dans la mise en valeur de l'intérêt de prendre en considération les différents types de ressource et leurs différentes caractéristiques pour fournir un mécanisme d'aide à la décision permettant de mieux réussir les projets d'automatisation de processus. Les implémentations développées sont des preuves de concept et ne représente pas un outil de modélisation et de simulation complètement abouti. De ce fait, les limites qui peuvent permettre de définir des perspectives à nos travaux sont à notre avis :

- Le premier point à étudier est la possibilité de remplacer les annotations textuelles par des éléments graphiques tout en gardant le modèle BPMN 2.0 intact et en conservant les recommandations de l'OMG. Comme nous l'avons énoncé dans le chapitre de l'état de l'art, l'intérêt d'utilisation des annotations textuelles et de pouvoir conserver le modèle du processus pour faciliter son échange tout en gardant les informations liées à ses besoins non-fonctionnels. Cependant, cette représentation n'est pas très « user-friendly », de plus, dans le cas des processus très complexes, ce mécanisme réduit la lisibilité du modèle. Nous avons également constaté pendant nos tests, que l'identification d'une erreur au niveau de la syntaxe, peut être la cause d'une perte de temps considérable. De ce fait, l'outil peut ne pas être largement accepté, d'où l'importance de trouver un mécanisme permettant d'utiliser des éléments graphiques pour ajouter les besoins non-fonctionnels tout en garantissant la portabilité du modèle et la conservation de tout type d'information. De plus cela permettra, d'automatiser l'utilisation des grilles de performance et d'améliorer la simulation des interfaces graphiques en impliquant l'utilisateur. Cela consiste à ce que l'utilisateur remplisse par lui-même les interfaces à simuler. Les approches d'ingénierie dirigées par les modèles, peuvent être une première prise de recherche
- Les indicateurs non-fonctionnels utilisés dans ce travail de thèse se limitent à la performance et à la fiabilité. L'ajout d'autres indicateurs serait intéressant pour mieux se rapprocher de l'atteinte des objectifs souhaités. Dans un premier temps, nous pensons que l'ajout des coûts en tant qu'indicateur serait utile à la prise de décision. L'indicateur du coût représente les dépenses nécessaires à l'exécution d'un processus métier. Ainsi, les indicateurs de performance, de la fiabilité et du coût sont interdépendants car les dépenses du processus sont liées à la quantité de temps requise pour effectuer/réparer une tâche par les ressources impliquées. Ainsi, l'analyste peut avoir une idée globale et

complète des coûts des différents scénarios de simulation joués. Ceci est d'autant plus important quand on prend en considération les différentes caractéristiques des ressources. Un opérateur expérimenté coûte plus cher qu'un opérateur junior.

- La relation entre la performance et la fiabilité n'est pas prise en considération dans notre travail de thèse. Trouver l'association entre ces indicateurs peut aider à éclairer sur les liens de cause à effets des anomalies ce qui permet de fournir des informations plus pointues pour la prise de décision. Les techniques de « process mining » peuvent être une première piste de recherche.
- L'approche de modélisation et simulation proposée est encore très dépendante des informations disponibles et fournies par les parties prenantes. Coupler cette approche avec les techniques de « process mining » peut aider à identifier plus facilement des informations réelles sur la performance du processus, les chemins les plus parcourus, etc.

DIFFUSIONS DES RESULTATS

Article revue

- Kawtar Ougaabal, Grégory Zacharewicz, Yves Ducq et Said Tazi, “Visual Workflow Process Modeling and Simulation Approach Based on Non-Functional Properties of Resources”, Applied Sciences, MDPI, 2020, 10 (13), pp. 4664.

Articles conférences

- Kawtar Ougaabal, Grégory Zacharewicz, Yves Ducq et Said Tazi: “Path simulation in BPMN workflow using resource aggregation”, Workshop I-ESA, November 2020, Tarbes, France
- Kawtar Ougaabal, Grégory Zacharewicz, Yves Ducq et Said Tazi: “Distinguishing Resource Type in BPMN Workflows at Simulation Phase”, Spring Simulation Multi – Conference, April 2019.
- Kawtar Ougaabal, Grégory Zacharewicz, Yves Ducq et Said Tazi: “Functional and Non-Functional BPMN M&S at Design Time”, 27th IEEE International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprise, June 2018
- Kawtar Ougaabal, Grégory Zacharewicz, Yves Ducq et Said Tazi: “Extending Non-functional BPMN Properties for Simulation of Workflow Pathways (WIP)”, Spring Simulation Multi – Conference, April 2018

Conférences nationales

- Kawtar Ougaabal, Grégory Zacharewicz, Yves Ducq, Said Tazi. “Distinguishing resource type in BPMN”, NICST, September 2019, Bordeaux, France
- Kawtar Ougaabal, Grégory Zacharewicz, Yves Ducq, Said Tazi. “Distinction des ressources dans BPMN”, JIAE, May 2019, Bidart, France
- Kawtar Ougaabal, Grégory Zacharewicz, Yves Ducq, Said Tazi. “Simulation des propriétés fonctionnelles et non-fonctionnelles avec BPMN”, GDR MACS, May 2018, Lille, France

REFERENCES

- Abar, S., Theodoropoulos, G.K., Lemarinier, P., O'Hare, G.M.P., 2017. Agent Based Modelling and Simulation tools: A review of the state-of-art software. *Computer Science Review* 24, 13–33. <https://doi.org/10.1016/j.cosrev.2017.03.001>
- Amjad, A., Azam, F., Anwar, M.W., Butt, W.H., Rashid, M., Naeem, A., 2018. UMLPACE for Modeling and Verification of Complex Business Requirements in Event-Driven Process Chain (EPC). *IEEE Access* 6, 76198–76216. <https://doi.org/10.1109/ACCESS.2018.2883610>
- Amokrane, N., 2016. De l'ingénierie des Besoins à l'ingénierie des Exigences: Vers une Démarche Méthodologique d'Ingénierie de Systèmes Complexes, de Verification et de Validation Appliquée à l'Informatisation des PME. Université Montpellier, Montpellier.
- Arevalo, C., Escalona, M.J., Ramos, I., Domínguez-Muñoz, M., 2016. A metamodel to integrate business processes time perspective in BPMN 2.0. *Information and Software Technology* 77, 17–33. <https://doi.org/10.1016/j.infsof.2016.05.004>
- Aris, 2007. The Event-driven Process Chain, in: ARIS Design Platform. Springer London, London, pp. 105–125. https://doi.org/10.1007/978-1-84628-613-1_7
- Arnaud, S., 2014. Implémentation d'un éditeur BPMN au sein d'un outil de métamodélisation. Belgique.
- Awad, A., Grosskopf, A., Meyer, A., Weske, M., 2009. Enabling Resource Assignment Constraints in BPMN (Technical). Hasso Plattner Institute, Germany.
- Awad, A., Puhlmann, F., 2008. Structural Detection of Deadlocks in Business Process Models, in: Abramowicz, W., Fensel, D. (Eds.), *Business Information Systems, Lecture Notes in Business Information Processing*. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 239–250. https://doi.org/10.1007/978-3-540-79396-0_21
- Barbalho de Figueroa, L.H., Salman, R., Horkoff, J., Chauhan, S., Davila, M., De Oliveira Neto, F.G., Schliep, A., 2020. A Modeling Approach for Bioinformatics Workflows. SPRINGER NATURE, Luxembourg, Luxembourg.
- Barbier, G., 2013. Contribution de l'ingénierie dirigée par les modèles à la conception de modèles grande culture. Université Blaise Pascal – Clermont II.
- Barouch, G., 2013. Le guide du client satisfait: le management des processus pas à pas : TPE-PME. Ed. Livres à Vivre Afnor Bem, Issy-Les-Moulineaux (Hauts-de-Seine); Saint-Denis-La Plaine (Seine-Saint-Denis); Talence (Gironde).
- Barth, R., Meyer, M., Spitzner, J., 2012. Typical Pitfalls of Simulation Modeling - Lessons Learned from Armed Forces and Business. *JASSS* 15, 5. <https://doi.org/10.18564/jasss.1935>
- Bartolini, C., Bertolino, A., De Angelis, G., Ciancone, A., Mirandola, R., 2012. Non-functional analysis of service choreographies, in: 2012 4th International Workshop on Principles of Engineering Service-Oriented Systems (PESOS). Presented at the 2012 4th International Workshop on Principles of Engineering Service-Oriented Systems (PESOS), IEEE, Zurich, Switzerland, pp. 8–14. <https://doi.org/10.1109/PESOS.2012.6225947>
- Bastos, R.M., Ruiz, D.D.A., 2002. Extending UML activity diagram for workflow modeling in production systems, in: Proceedings of the 35th Annual Hawaii International Conference on System Sciences. Presented at the 35th Annual Hawaii International

- Conference on System Sciences, IEEE Comput. Soc, Big Island, HI, USA, pp. 3786–3795. <https://doi.org/10.1109/HICSS.2002.994510>
- Bastos, R.M., Ruiz, D.D.A., 2001. Towards an approach to model business processes using workflow modeling techniques in production systems, in: Proceedings of the 34th Annual Hawaii International Conference on System Sciences. Presented at the Hawaii International Conference on System Sciences. HICSS-34, IEEE Comput. Soc, Maui, HI, USA, p. 10. <https://doi.org/10.1109/HICSS.2001.927231>
- Bazoun, H., 2015. Contribution à une méthodologie pour la modélisation des systèmes de services et d'ingénierie grâce à une approche dirigée par les modèles : l'architecture, la transformation et la simulation du modèle. Université de Bordeaux, Bordeaux.
- BenSaid, I., Chaâbane, M.A., Bouaziz, R., Andonoff, E., 2016. A Version-based Approach to Address Flexibility of BPMN Collaborations and Choreographies:, in: Proceedings of the 13th International Joint Conference on E-Business and Telecommunications. Presented at the International Conference on e-Business, SCITEPRESS - Science and Technology Publications, Lisbon, Portugal, pp. 31–42. <https://doi.org/10.5220/0005967100310042>
- Bocciarelli, P., D'Ambrogio, A., 2011. A BPMN Extension for Modeling Non Functional Properties of Business Processes. Presented at the Symposium On Theory of Modeling and Simulation - DEVS Integrative M&S Symposium (TMS/DEVS 2011), USA.
- Bocciarelli, P., D'Ambrogio, A., Giglio, A., Paglia, E., 2016. A BPMN Extension to Enable the Explicit Modeling of Task Resources. Presented at the INCOSE Italia Conference on Systems Engineering, ITALY.
- Bocciarelli, P., D'Ambrogio, A., Giglio, A., Paglia, E., Gianni, D., 2014. Empowering business process simulation through automated model transformations. Presented at the Spring Simulation Multiconference, SpringSim'14, USA.
- Borshchev, A., Filippov, A., 2004. From System Dynamics and Discrete Event to Practical Agent Based Modeling: Reasons, Techniques, Tools. Presented at the The 22nd International Conference of the System Dynamics Society, England.
- Bourey, J.-P., Grangel, R., Ducq, Y., 2007. Report on Model Driven Interoperability (Technical). INTEROP Interoperability Research for Networked Enterprises Applications and Software.
- Bourlier, M., 2015. Développement spécifique ou progiciel ? Quels sont les critères de décision à prendre en compte dans le choix entre développement spécifique et progiciel ? I.A.E Grenoble.
- Bouzidi, S., 2017. Prise en compte des ressources humaines dans l'ordonnancement conjoint de production et de maintenance. Ecole Nationale Supérieur d'information - ESI - Alger, Alger.
- Brandenburg, H., Wojtyna, J.-P., 2006. L'approche processus, Mode d'emploi, 2nd ed. Eyrolles.
- Braun, R., Loucopoulos, P., Pastor, Ó., Petrounias, I. (Eds.), 2014a. Classification of Domain-Specific BPMN Extensions, Lecture Notes in Business Information Processing. Springer Berlin Heidelberg, Berlin, Heidelberg. <https://doi.org/10.1007/978-3-662-45501-2>
- Braun, R., Schlieter, H., Burwitz, M., Esswein, W., 2014b. BPMN4CP: Design and implementation of a BPMN extension for clinical pathways, in: 2014 IEEE International Conference on Bioinformatics and Biomedicine (BIBM). Presented at the 2014 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), IEEE, Belfast, United Kingdom, pp. 9–16. <https://doi.org/10.1109/BIBM.2014.6999261>
- Buse, R.P.L., Weimer, W., 2009. The road not taken: Estimating path execution frequency statically, in: 2009 IEEE 31st International Conference on Software Engineering. Presented at the 2009 31st International Conference on Software Engineering (ICSE

- 2009), IEEE, Vancouver, BC, pp. 144–154. <https://doi.org/10.1109/ICSE.2009.5070516>
- Cabanillas, C., Knuplesch, D., Resinas, M., Reichert, M., Mendling, J., Ruiz-Cortés, A., 2015. RALph: A Graphical Notation for Resource Assignments in Business Processes, in: Zdravkovic, J., Kirikova, M., Johannesson, P. (Eds.), *Advanced Information Systems Engineering, Lecture Notes in Computer Science*. Springer International Publishing, Cham, pp. 53–68. https://doi.org/10.1007/978-3-319-19069-3_4
- Cardoso, J., Sheth, A., Miller, J., Arnold, J., Kochut, K., 2004. Quality of service for workflows and web service processes. *Journal of Web Semantics* 1, 281–308. <https://doi.org/10.1016/j.websem.2004.03.001>
- Cauffriez, L., Willaëys, D., Defrenne, J., 1997. Mesure des indicateurs de performances de lignes de production: présentation d’une méthode et retour d’expérience. *Journal Européen des systèmes Automatisés*.
- Čerić, V., Hlupić, V., 1993. Modelling a Solid-Waste Processing System by Discrete Event Simulation. *Journal of the Operational Research Society* 44, 107–114. <https://doi.org/10.1057/jors.1993.22>
- Chen, D., Vallespir, B., Doumeingts, G., 1997. GRAI integrated methodology and its mapping onto generic enterprise reference architecture and methodology. *Computers in Industry* 33, 387–394. [https://doi.org/10.1016/S0166-3615\(97\)00043-2](https://doi.org/10.1016/S0166-3615(97)00043-2)
- Ciancone, A., Filieri, A., Drago, M.L., Mirandola, R., Grassi, V., 2011. Klapersuite: An Integrated Model-Driven Environment for Reliability and Performance Analysis of Component-Based Systems, in: Bishop, J., Vallecillo, A. (Eds.), *Objects, Models, Components, Patterns, Lecture Notes in Computer Science*. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 99–114. https://doi.org/10.1007/978-3-642-21952-8_9
- Combemale, B., 2008. *Ingénierie Dirigée par les Modèles (IDM)–État de l’art*.
- Compliance Consulting, 2017. *L’ingénierie des Exigences pour le Système d’Information*. URL <https://www.valerieliu.fr/extraits/compliance/livre-blanc/livreblanc.pdf> (accessed 10.3.20).
- Curtis, B., Kellner, M.I., Over, J., 1992. Process modeling. *Commun. ACM* 35, 75–90. <https://doi.org/10.1145/130994.130998>
- David, A., Larsen, K.G., Legay, A., Mikučionis, M., Poulsen, D.B., 2015. Uppaal SMC tutorial. *Int J Softw Tools Technol Transfer* 17, 397–415. <https://doi.org/10.1007/s10009-014-0361-y>
- Daw, Z., Cleaveland, R., 2016. An extensible formal semantics for UML activity diagrams. *arXiv:1604.02386 [cs]*.
- Debauche, B., Mégard, P., 2004. *BPM Business Process Management : Pilotage métier de l’entreprise*. Hermès science publications.
- Di Marco, A., Pompilio, C., Bertolino, A., Calabrò, A., Lonetti, F., Sabetta, A., 2011. Yet another meta-model to specify non-functional properties, in: *Proceedings of the International Workshop on Quality Assurance for Service-Based Applications - QASBA ’11*. Presented at the the International Workshop, ACM Press, Lugano, Switzerland, p. 9. <https://doi.org/10.1145/2031746.2031751>
- Dolques, X., 2010. *Génération de Transformations de Modèles : une approche basée sur les treillis de Galois*. Université de Montpellier II.
- Domingos, D., Respício, A., Martinho, R., 2016. Using Resource Reliability in BPMN Processes. *Procedia Computer Science* 100, 1280–1288. <https://doi.org/10.1016/j.procs.2016.09.243>
- Dosimont, D., 2006. *Agrégation spatiotemporelle pour la visualisation de traces d’exécution*.
- Doumeingts, G., Ducq, Y., Vallespir, B., Kleinhans, S., 2000. Production management and enterprise modelling. *Computers in Industry* 42, 245–263. [https://doi.org/10.1016/S0166-3615\(99\)00074-3](https://doi.org/10.1016/S0166-3615(99)00074-3)

- Ducq *, Y., Vallespir, B., 2005. Definition and aggregation of a performance measurement system in three aeronautical workshops using the ECOGRAI method. *Production Planning & Control* 16, 163–177. <https://doi.org/10.1080/09537280512331333066>
- Ducq, Y., 1999. Contribution à une méthodologie d'analyse de la cohérence des Systèmes de Production dans le cadre du Modèle GRAI. Université de Bordeaux.
- Dumas, M., Aalst, W. van der, Ter Hofstede, A. (Eds.), 2005. *Process-aware information systems: bridging people and software through process technology*. Wiley-Interscience, Hoboken, N.J.
- Edoze, E.S., 2010. *Materials management and organizational effectiveness: A study of Selected Telecommunication Firms in Delta State*.
- Elmounadi, A., Berbiche, N., Sefiani, N., 2013. Architecture dirigée par les modèles : Méthodes et outils de transformation de modèles. Presented at the La cinquième édition des Journées Doctorales en Technologies de l'Information et de la Communication (JDTIC'13), Kenitra, Maroc.
- Entringer, T.C., Nascimento, D.C. de O., Ferreira, A. da S., Siqueira, P.M.T., Boechat, A. de S., Cerchiaro, I.B., Mendonça, S.B., Ramos, R.R., 2019. Comparative Analysis main Methods Business Process Modeling: Literature Review, Applications and Examples. *IJAERS* 6, 100–116. <https://doi.org/10.22161/ijaers.6.5.15>
- Freire, J.-C., Giraudin, J.-P., Front, A., 1997. Atelier MODSI: Un Outil de M'éta-Modélisation et de Multi-Modélisation. Presented at the Séminaire vérification de Toulouse, Toulouse.
- Garredu, S., Vittori, E., Santucci, J.-F., Poggi, B., 2014. A Survey of Model-Driven Approaches Applied to DEVS - A Comparative Study of Metamodels and Transformations:, in: *Proceedings of the 4th International Conference on Simulation and Modeling Methodologies, Technologies and Applications*. Presented at the 4th International Conference on Simulation and Modeling Methodologies, Technologies and Applications, SCITEPRESS - Science and Technology Publications, Vienna, Austria, pp. 179–187. <https://doi.org/10.5220/0005041001790187>
- Geiger, M., Harrer, S., Lenhard, J., Casar, M., Vorndran, A., Wirtz, G., 2015. BPMN Conformance in Open Source Engines, in: *2015 IEEE Symposium on Service-Oriented System Engineering*. Presented at the 2015 IEEE Symposium on Service-Oriented System Engineering (SOSE), IEEE, San Francisco Bay, CA, USA, pp. 21–30. <https://doi.org/10.1109/SOSE.2015.22>
- Greasley, A., 2009. A Comparison of System Dynamics and Discrete Event Simulation. Presented at the Summer Computer System Simulation Conference, Turkey.
- Gu, F., Zhang, X., Chen, M., Große, D., Drechsler, R., 2016. Quantitative Timing Analysis of UML Activity Diagrams Using Statistical Model Checking, in: *Proceedings of the 2016 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. Presented at the Proceedings of the 2016 Design, Automation & Test in Europe Conference & Exhibition (DATE), Research Publishing Services, pp. 780–785. https://doi.org/10.3850/9783981537079_0339
- Hardebolle, C., 2008. *Composition de modèles pour la modélisation multi-paradigme du comportement des systèmes*. Université Paris-Sud XI, Paris.
- Heguy, X., 2018a. thèse suivante Extensions de BPMN 2.0 et méthode de gestion de la qualité pour l'interopérabilité des données. Université de Bordeaux.
- Heguy, X., 2018b. Extensions de BPMN 2.0 et méthode de gestion de la qualité pour l'interopérabilité des données. Université de Bordeaux, Bordeaux.
- Hellriegel, D., Slocum, J.W., 2006. *Management des organisations*, 2e Éd. ed, Business school. de Boeck&Larcier, Bruxelles.
- Hollingsworth, D., 1995. *Workflow management coalition. The Workflow Reference Model*. ISO15288, 2008. *Systems Engineering - System Life-Cycle Processes*.

- Jander, K., Braubach, L., Pokahr, A., Lamersdorf, W., 2011. Validation of Agile Workflows Using Simulation, in: Dastani, M., El Fallah Seghrouchni, A., Hübner, J., Leite, J. (Eds.), *Languages, Methodologies, and Development Tools for Multi-Agent Systems*, Lecture Notes in Computer Science. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 39–55. https://doi.org/10.1007/978-3-642-22723-3_3
- Jeong, K.-Y., Wu, L., Hong, J.-D., 2009. IDEF method-based simulation model design and development framework. *JITEM* 2, 337–359. <https://doi.org/10.3926/jiem.2009.v2n2.p337-359>
- Jin, G., 2019. *Management of the design process : human resource allocation in factory of the future*. Université de Bordeaux, Bordeaux.
- Karolina, C., Pacana, A., Dwornicka, R., 2020. Improvement Of The Production Process With The Use Of Selected Kpis. *System Safety: Human - Technical Facility - Environment* 2, 307–315. <https://doi.org/10.2478/czoto-2020-0038>
- Kettinger, W.J., Teng, J.T.C., Guha, S., 1997. Business Process Change: A Study of Methodologies, Techniques, and Tools. *MIS Quarterly* 21, 55. <https://doi.org/10.2307/249742>
- Kherbouche, M.O., 2013. *Contribution à la gestion de l'évolution des processus métiers*. Université du littoral Côte d'Opale.
- Kopyltsov, A.V., 2020. Selection of metrics in software quality evaluation. *J. Phys.: Conf. Ser.* 1515, 032018. <https://doi.org/10.1088/1742-6596/1515/3/032018>
- Korherr, B., List, B., 2007. Extending the EPC with performance measures, in: *Proceedings of the 2007 ACM Symposium on Applied Computing - SAC '07*. Presented at the the 2007 ACM symposium, ACM Press, Seoul, Korea, p. 1265. <https://doi.org/10.1145/1244002.1244275>
- Krogstie, J., 2012. Perspectives to Process Modeling – A Historical Overview, in: Bider, I., Halpin, T., Krogstie, J., Nurcan, S., Proper, E., Schmidt, R., Soffer, P., Wrycza, S. (Eds.), *Enterprise, Business-Process and Information Systems Modeling*, Lecture Notes in Business Information Processing. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 315–330. https://doi.org/10.1007/978-3-642-31072-0_22
- Kromm, H., 2002. *Contribution à une méthodologie d'analyse de la cohérence entre les objectifs de conception et d'exploitation d'un système de production*. Université de Bordeaux.
- Le Moigne, J.-L., 1973. *Les systèmes d'information dans les organisations*.
- Leymann, F., Roller, D., 2000. *Production workflow : concepts and techniques*. (Prentice Hall PTR Upper Saddle River).
- Lianjun An, Jun-Jang Jeng, 2005. On Developing System Dynamics Model for Business Process Simulation, in: *Proceedings of the Winter Simulation Conference, 2005*. Presented at the Winter Simulation Conference, 2005., IEEE, Orlando, FL. USA, pp. 2068–2077. <https://doi.org/10.1109/WSC.2005.1574489>
- Marier, S., 1996. *Modélisation et évaluation des performances des processus industriels semi-structurés*. Institut national polytechnique de Grenoble.
- Martinho, R., Domingos, D., Varajão, J., 2015. CF4BPMN: A BPMN Extension for Controlled Flexibility in Business Processes. *Procedia Computer Science* 64, 1232–1239. <https://doi.org/10.1016/j.procs.2015.08.509>
- McReady, S., 1992. There is more than one kind of Workflow software. *Computerworld* 86–90.
- M.Dijkman, R., Dumas, M., Ouyang, C., 2007. *Formal Semantics and Automated Analysis of BPMN Process Models*. Computer Science.
- Mendling, J., Neumann, G., Nüttgens, M., 2005. Yet Another Event-Driven Process Chain, in: van der Aalst, W.M.P., Benatallah, B., Casati, F., Curbera, F. (Eds.), *Business Process*

- Management, Lecture Notes in Computer Science. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 428–433. https://doi.org/10.1007/11538394_35
- Mendling, J., Nüttgens, M., 2006. EPC markup language (EPML): an XML-based interchange format for event-driven process chains (EPC). *ISeB* 4, 245–263. <https://doi.org/10.1007/s10257-005-0026-1>
- Mili, H., Tremblay, G., Jaoude, G.B., Lefebvre, É., Elabed, L., Boussaidi, G.E., 2010. Business process modeling languages: Sorting through the alphabet soup. *ACM Comput. Surv.* 43, 1–56. <https://doi.org/10.1145/1824795.1824799>
- M.Law, A., Kelton, W.D., 2000. *Simulation Modeling and Analysis*.
- Narayan Debnath, Riesco, D., Montejano, G., Acosta, E., Uva, M., 2003. Including workflow concurrent modelling in an extension of the UML activity diagram metamodel, in: *ACS/IEEE International Conference on Computer Systems and Applications, 2003. Book of Abstracts. Presented at the ACS/IEEE International Conference on Computer Systems and Applications, IEEE, Tunis, Tunisia, p. 51.* <https://doi.org/10.1109/AICCSA.2003.1227484>
- Object Management Group, n.d. *Business Process Model and Notation (BPMN) – Version 2.0*.
- O’Brien, J., Marion, G., 1997. *Les systèmes d’information de gestion*.
- OMG, 2014. *Object Constraint Language (OCL) v2.4*.
- OMG, 2003. *MDA Guide Version 1.0.1*.
- OMG, 2002. *Meta Object Facility (MOF) Specification Version 1.4*.
- Ouazene, Y., 2013. *Maîtrise des systèmes industriels : optimisation de la conception des lignes de production Spécialité : Optimisation et Sûreté des Systèmes*. Université de technologie Troyes, Troyes.
- Paglia, E., 2017. *Innovative Approaches for Business Process Modeling and Simulation*. University Of Rome “Tor Vergata,” Italy.
- Paschke, A., Schnappinger-Gerull, E., 2006. A Categorization Scheme for SLA Metrics. Presented at the *Service Oriented Electronic Commerce*, Passau.
- Pétin, J.-F., 2007. *Méthodes et modèles pour un processus sûr d’automatisation*. Université Henri Poincaré – Nancy I.
- Pika, A., Wynn, M.T., Fidge, C.J., ter Hofstede, A.H.M., Leyer, M., van der Aalst, W.M.P., 2014. An Extensible Framework for Analysing Resource Behaviour Using Event Logs, in: Jarke, M., Mylopoulos, J., Quix, C., Rolland, C., Manolopoulos, Y., Mouratidis, H., Horkoff, J. (Eds.), *Advanced Information Systems Engineering, Lecture Notes in Computer Science*. Springer International Publishing, Cham, pp. 564–579. https://doi.org/10.1007/978-3-319-07881-6_38
- Polderdijk, M., Vanderfeesten, I., Erasmus, J., Traganos, K., Bosch, T., van Rhijn, G., Fahland, D., 2018. A Visualization of Human Physical Risks in Manufacturing Processes Using BPMN, in: Teniente, E., Weidlich, M. (Eds.), *Business Process Management Workshops, Lecture Notes in Business Information Processing*. Springer International Publishing, Cham, pp. 732–743. https://doi.org/10.1007/978-3-319-74030-0_58
- Radloff, M., Schultz, M., Nüttgens, M., 2015. Extending different Business Process Modeling Languages with Domain Specific Concepts: The Case of Internal Controls in EPC and BPMN. Presented at the *International Workshop on Enterprise Modelling and Information Systems Architectures*, Innsbruck, Austria.
- Ravindran, B., 2002. Engineering dynamic real-time distributed systems: architecture, system description language, and middleware. *IEEE Trans. Software Eng.* 28, 30–57. <https://doi.org/10.1109/32.979988>
- Reid, I., Bamford, D., Ismail, H., 2019. Reconciling engineer-to-order uncertainty by supporting front-end decision-making. *International Journal of Production Research* 57, 6856–6874. <https://doi.org/10.1080/00207543.2018.1552370>

- Revel, L., Habchi, G., Maire, J.L., 2004. Analyse du processus d'élaboration d'un projet de simulation. Presented at the MOSIM'04, Nantes, France.
- Rodet, J., 2009. Verticalité, horizontalité et changement dans les organisations. Pensée disjonctive, pensée conjonctive et pensée complexe. URL <http://jacques.rodet.free.fr/vhchgt.pdf> (accessed 10.3.20).
- Rosemann, M., Muehlen, M. z., 2005. Integrating Risks in Business Process Models. Presented at the Information Systems (ACIS), Australia.
- Roy, D., Anciaux, D., Monteiro, T., Ouzizi, L., 2004. Multi-agent architecture for supply chain management. *Jnl of Manu Tech Mngmnt* 15, 745–755. <https://doi.org/10.1108/17410380410565339>
- Russell, N., van der Aalst, W.M.P., ter Hofstede, A.H.M., Edmond, D., 2005. Workflow Resource Patterns: Identification, Representation and Tool Support, in: King, R. (Ed.), *Active Flow and Combustion Control 2018, Notes on Numerical Fluid Mechanics and Multidisciplinary Design*. Springer International Publishing, Cham, pp. 216–232. https://doi.org/10.1007/11431855_16
- Sahraoui, S., Berrah, L., Montmain, J., 2007. Techniques d'optimisation pour la définition d'une démarche d'amélioration industrielle : une approche par analyse et agrégation des performances.
- Schionig, S., Cabanillas, C., Claudio, D.C., Jablonski, S., Mendling, J., 2016. Mining resource assignments and teamwork compositions from process logs.
- Scowen, R.S., 1998. Extended BNF | A generic base standard. Presented at the Software Engineering Standards Symposium, pp. 6–2.
- Smith, H., Neal, D., Ferrara, L., Hayden, F., 2002. The emergence of business process management. White paper.
- Speck, A., Pulvermuller, E., Heuzeroth, D., 2003. Validation of Business Process Models. Presented at the Workshop on Correctness of Model-based Software Composition (CMC), ECOOP 2003, Germany.
- Spyrou, S., D Bamidis, P., Pappas, K., 2005. Extending UML Activity Diagrams for Workflow Modelling with Clinical Documents in Regional Health Information Systems. Presented at the Connecting Medical Informatics and Bio-Informatics: Proceedings of the 19th Medical Informatics Europe Conference, Geneva, Switzerland.
- Stroppi, L.J.R., Chiotti, O., Villarreal, P.D., 2015. Defining the resource perspective in the development of processes-aware information systems. *Information and Software Technology* 59, 86–108. <https://doi.org/10.1016/j.infsof.2014.10.006>
- Taillard, P., 2008. Mesurer les performances système de production. *Technologie, sciences et technologies industrielles*.
- TNO, 2012. Checklist Physical Load. <https://www.fysiekebelasting.tno.nl/en/instrumenten/checklist-physical-load/>. Accés 03/10/2020.
- Tudor, C.H., 2016. Modélisation des processus métiers : Etat de l'art et conseils pratiques. Centre de Recherche Public Henri Tudor.
- Uhrmacher, A.M., 2012. Seven pitfalls in modeling and simulation research, in: *Proceedings Title: Proceedings of the 2012 Winter Simulation Conference (WSC)*. Presented at the 2012 Winter Simulation Conference - (WSC 2012), IEEE, Berlin, Germany, pp. 1–12. <https://doi.org/10.1109/WSC.2012.6465321>
- Van Der Aalst, W., Van hee, K., 2002. *Workflow Management: Models, Methods, and Systems*. The MIT Press. <https://doi.org/10.7551/mitpress/7301.001.0001>
- van der Aalst, W.M.P., 2015. Business Process Simulation Survival Guide, in: vom Brocke, J., Rosemann, M. (Eds.), *Handbook on Business Process Management 1*. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 337–370. https://doi.org/10.1007/978-3-642-45100-3_15

- van der Aalst, W.M.P., 2000. Workflow Verification: Finding Control-Flow Errors Using Petri-Net-Based Techniques, in: van der Aalst, W., Desel, J., Oberweis, A. (Eds.), *Business Process Management, Lecture Notes in Computer Science*. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 161–183. https://doi.org/10.1007/3-540-45594-9_11
- Van Der Aalst, W.M.P., 1998. The Application Of Petri Nets To Workflow Management. *J Circuit Syst Comp* 08, 21–66. <https://doi.org/10.1142/S0218126698000043>
- van der Aalst, W.M.P., ter Hofstede, A.H.M., Kiepuszewski, B., Barros, A.P., 2003. Workflow Patterns. *Distributed and Parallel Databases* 14, 5–51. <https://doi.org/10.1023/A:1022883727209>
- van der Aalst, Wil M. P., ter Hofstede, A.H.M., Weske, M., 2003. Business Process Management: A Survey, in: van der Aalst, W.M.P., Weske, M. (Eds.), *Business Process Management, Lecture Notes in Computer Science*. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 1–12. https://doi.org/10.1007/3-540-44895-0_1
- Vanderperren, Y., Mueller, W., Dehaene, W., 2008. UML for electronic systems design: a comprehensive overview. *Des Autom Embed Syst* 12, 261–292. <https://doi.org/10.1007/s10617-008-9028-9>
- Varenne, F., Silberstein, M., 2013. *Modéliser and simuler, tome 1 Epistémologies et pratiques de la modélisation et de la simulation*. Editions Matériologiques, Paris.
- Vasilecas, O., Laureckas, E., Rima, A., 2014. Analysis of Using Resources in Business Process Modeling and Simulation. *Applied Computer Systems* 16, 19–25. <https://doi.org/10.1515/acss-2014-0009>
- Verbeek, H.M.W., 2001. Diagnosing Workflow Processes using Woflan. *The Computer Journal* 44, 246–279. <https://doi.org/10.1093/comjnl/44.4.246>
- Vernadat, F., 1999. *Techniques de Modélisation en Entreprise: Applications aux Processus Opérationnels*, Economica. ed.
- Vernadat, F.B., 1997. Enterprise Modelling Languages, in: Kosanke, K., Nell, J.G. (Eds.), *Enterprise Engineering and Integration*. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 212–224. https://doi.org/10.1007/978-3-642-60889-6_24
- Weske, M., 2007. *Business Process Management*. Springer Berlin Heidelberg, Berlin, Heidelberg. <https://doi.org/10.1007/978-3-540-73522-9>
- WFMC, 1999. The Workflow Mangement Coalition, *Workflow Management Coalition Terminology and Glossary*, technical report WfMC-TC-1011.
- Wohed, P., van der Aalst, W.M.P., Dumas, M., ter Hofstede, A.H.M., Russell, N., 2005. Pattern-Based Analysis of the Control-Flow Perspective of UML Activity Diagrams, in: Delcambre, L., Kop, C., Mayr, H.C., Mylopoulos, J., Pastor, O. (Eds.), *Conceptual Modeling – ER 2005, Lecture Notes in Computer Science*. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 63–78. https://doi.org/10.1007/11568322_5
- Xu, J., Anders, S., Pruttianan, A., France, D., Lau, N., Adams, J.A., Weinger, M.B., 2018. Human performance measures for the evaluation of process control human-system interfaces in high-fidelity simulations. *Applied Ergonomics* 73, 151–165. <https://doi.org/10.1016/j.apergo.2018.06.008>
- Yende, R., 2019. *Support De Cours De Génie Logiciel*.
- Zacharewicz, G., 2006. *Un Environnement G-DEVS/HLA : Application A La Modelisation Et Simulation Distribuée De Workflow*. Université Paul Cézanne Aix-Marseille III.
- Zarour, K., Benmerzoug, D., Guermouche, N., Drira, K., 2019. A systematic literature review on BPMN extensions. *BPMJ ahead-of-print*. <https://doi.org/10.1108/BPMJ-01-2019-0040>
- Zhou, K., Rong, G., 2010. Study of supply chain monitoring system based on IDEF method, in: *2010 International Conference on Logistics Systems and Intelligent Management (ICLSIM)*. Presented at the 2010 International Conference on Logistics Systems and

Intelligent Management (ICLSIM), IEEE, Harbin, China, pp. 278–281.
<https://doi.org/10.1109/ICLSIM.2010.5461420>

Zor, S., Leymann, F., Schum, D., 2011. A Proposal of BPMN Extensions for the Manufacturing Domain. Presented at the Proceedings of 44th CIRP International Conference on Manufacturing Systems.