



HAL
open science

Extensions de BPMN 2.0 et méthode de gestion de la qualité pour l'interopérabilité des données

Xabier Heguy

► **To cite this version:**

Xabier Heguy. Extensions de BPMN 2.0 et méthode de gestion de la qualité pour l'interopérabilité des données. Automatique. Université de Bordeaux, 2018. Français. NNT : 2018BORD0375 . tel-02122344

HAL Id: tel-02122344

<https://theses.hal.science/tel-02122344>

Submitted on 15 Jul 2019

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

SPECIALITÉ : PRODUCTIQUE

Par Xabier HEGUY

**EXTENSIONS DE BPMN 2.0 ET METHODE DE GESTION DE
LA QUALITE POUR L'INTEROPERABILITE DES DONNEES**

Sous la Direction de : Yves DUCQ

(Co-Directeur: Gregory ZACHAREWICZ et Co-Encadrant Saïd TAZI)

Soutenue le 13 Décembre 2018, à 10 :00

Membres du jury :

M. MOALLA, Nejib Maître de Conférences HDR	Université de Lyon	Rapporteur
M. EXPOSITO, Ernesto Professeur des Universités	Université Pau et des Pays de l'Adour	Rapporteur
M. CHEN, David Professeur des Universités	Université de Bordeaux	Président
M. GJURASEVIC, Pierre Directeur commercial	Société Onetik	Examineur
M. DUCQ, Yves Professeur des Universités	Université de Bordeaux	Directeur
M. ZACHAREWICZ, Gregory Professeur de l'Institut Mines Telecom	IMT - Mines d'Alès	Co-Directeur
M. TAZI, Saïd Maitre de conférences	Université Pau et des Pays de l'Adour	Co-Encadrant

RESUME

Les entreprises sont amenées, plus que jamais, à échanger des données entre différents systèmes d'information, que ce soit avec leurs partenaires, mais aussi au sein d'une même organisation. Elles sont alors confrontées au problème que constitue l'interopérabilité des systèmes et des données. Cette problématique reste mal identifiée par de nombreux acteurs présents à tous les niveaux hiérarchiques de l'entreprise, elle engendre cependant des pertes de temps sous forme de tâches à non-valeur ajoutée ainsi qu'une baisse sensible de la qualité des données. Notre contribution est composée de deux parties. Tout d'abord, nous proposons et implémentons des extensions de BPMN 2.0 qui ont pour but de faire visualiser les échanges de données présentant un problème d'interopérabilité ainsi que ceux pour lesquels le problème a été résolu. Elles permettent aussi d'afficher les mesures de la performance au niveau du coût, de la durée, de la qualité et de la disponibilité des données pour les tâches qui sont concernées par ces échanges de données. Ces extensions permettent de faire prendre la mesure de la problématique que représente l'interopérabilité, ainsi que la nécessité de son établissement et ce, pour tous les acteurs de l'entreprise. La deuxième contribution est une méthode dérivée de l'AMDEC qui permet de visualiser chacune des données échangées dans le détail et d'en calculer la criticité. On peut ainsi hiérarchiser les priorités des actions correctives à mener pour améliorer la qualité des données sous la forme d'un processus d'amélioration continue afin de garantir une qualité optimale des données dans la durée.

Mots clés : Interopérabilité, BPMN, performance, AMDEC, qualité des données.

ABSTRACT

Increasingly, enterprises have to interchange huge amount of data between different information systems, whether with their business partners or within an internal department. The crucial problem of data interoperability then arises. This issue, which is largely ignored by most of the resources of the enterprises at any hierarchical level, causes waste of time in the form of non-value added task and a significant decrease of data quality. Our contribution consists of two parts. First, we propose and implement BPMN 2.0 extensions whose goal is to make visible data interchanges with interoperability issues and solutions. As well, the proposed extensions allow displaying performance measurement values of cost, time, data quality and reliability of the tasks involved in these data interchanges. Thanks to these extensions, users can mark the current level of interoperability and anticipate by proposing interoperability solution in order to bring competitiveness to the enterprise. The second contribution is a method rooted in FMECA which allows visualizing each interchanged data in detail and calculating its criticality. We can then make a priority ranking of corrective actions to enhance data quality in the form of a process of continuous improvement in order to ensure an optimal data quality in the long run.

Keywords: Interoperability, BPMN, performance, AMDEC, data quality.

TABLE DES MATIERES

Introduction générale.....	9
Chapitre 1 : Problématique	13
1.1 Introduction	14
1.2 Interopérabilité	14
1.2.1 Problèmes d'interopérabilité	15
1.2.2 Niveaux d'interopérabilité.....	16
1.2.3 Approches de l'interopérabilité.....	17
1.3 Besoin de modélisation et d'évaluation de la performance	18
1.3.1 Besoin de modélisation	18
1.3.2 L'évaluation de la performance.....	22
1.4 Besoin de gestion de la qualité des données	25
1.5 Nécessité d'un outil support pour une entreprise	26
1.6 Conclusion.....	26
Chapitre 2 : État de l'art.....	28
2.1 Introduction	29
2.2 Langages de modélisation de processus.....	29
2.2.1 BPMN	30
2.2.2 ARIS	32
2.2.3 IDEF	35
2.2.4 GRAI.....	39
2.2.5 CIMOSA	44
2.2.6 Synthèse des langages de modélisation	46
2.3 Méthodes de mesure de la performance	48
2.3.1 Tableau de bord.....	48
2.3.2 Balanced Scorecard.....	49
2.3.3 OVAR	51
2.3.4 GIMSI.....	53

2.3.5 PRISM	54
2.3.6 ABC/ABM/ABB	57
2.3.7 Synthèse des méthodes de mesure de la performance	58
2.4 Conclusion	59
<i>Chapitre 3 : Contributions conceptuelles et méthodologiques par ajout de concepts à BPMN.....</i>	61
3.1 Introduction	62
3.2 Démarche globale	62
3.3 Extension de BPMN 2.0	65
3.3.1 Mécanisme d'extension de BPMN	66
3.3.2 Propositions d'extension	67
3.4 Méthodes d'agrégation de la performance	73
3.4.1 Réduction des processus	74
3.4.2 Agrégation des mesures de performance	75
3.5 Gestion de la qualité des données	78
3.5.1 AMDEC	78
3.5.2 Adaptation de l'AMDEC à la qualité des données	81
3.6 Synthèse générale	90
3.7 Conclusion	93
<i>Chapitre 4 : Implémentation dans BPMN2 Modeler</i>	95
4.1 Introduction	96
4.1 Choix de BPMN2 Modeler	96
4.2 Présentation de BPMN2 Modeler	97
4.2.1 EDI Eclipse	98
4.2.2 EMF	98
4.2.3 Graphiti	98
4.2.4 BPMN 2 Modeler	99
4.3 Implémentation des extensions	102
4.3.1 Spécifications	102
4.3.2 Fonctionnement de BPMN 2 Modeler	103

4.3.3 Implémentation	103
4.3.4 Interface étendue	106
4.4 Conclusion	108
Chapitre 5 : Cas d'étude chez ONETIK	110
5.1 Introduction	111
5.2 Présentation de l'entreprise Onetik.....	111
5.3 Problématique et solution.....	112
5.3.1 ERP et fonctionnement	112
5.3.2 Problème d'interopérabilité	112
5.3.3 Solution envisagée	112
5.4 Modèles de résolution	113
5.4.1 Données utilisées	113
5.4.2 Modèle AS IS	116
5.4.3 Modèle TO BE	119
5.4.4 Comparaison des modèles	121
5.4.5 Modèle global	121
5.5 Elaboration de l'AMDEC données	124
5.5.1 Étape 1 : initialisation.....	124
5.5.2 Étape 2 : analyse fonctionnelle.....	124
5.5.3 Étape 3 : analyse qualitative et quantitative.....	126
5.5.4 Étape 4 : synthèse	128
5.5.5 Phase d'amélioration continue.....	129
5.6 Conclusion.....	130
Conclusion Générale.....	132
Perspectives	134
Diffusion des Résultats	134
Bibliographie.....	135

INDEX DES ILLUSTRATIONS

Figure 1 : Enterprise Interoperability Framework	18
Figure 2 : Exemple de modèle BPMN	31
Figure 3 : Data Object	31
Figure 4 : Data Collection	31
Figure 5 : Data Input.....	32
Figure 6 : Data Output.....	32
Figure 7 : Data store.....	32
Figure 8 : Approche ARIS	33
Figure 9: Exemple de chaîne de processus événementielle	34
Figure 10: Exemple de modèle entité-association.....	35
Figure 11: Représentation SADT/IDEFO (basique)	36
Figure 12 : Modélisation IDEFO	37
Figure 13 : Modèle de flux de processus de IDEF3	38
Figure 14 : Exemple de diagramme IDEF1X.....	39
Figure 15 : Représentation de l'horizon et de la période.....	41
Figure 16 : Exemple de grille GRAI	42
Figure 17 : Exemple de réseau GRAI	43
Figure 18 : Cube CIMOSA.....	45
Figure 19 : Les quatre axes du Balanced Scorecard	51
Figure 20 : Déroulement de la méthode OVAR.....	52
Figure 21 : Prisme de la performance.....	55
Figure 22 : Les facettes du prisme	57
Figure 23 : Principe de la méthode ABC	58
Figure 24 : Etapes de la démarche globale.....	63
Figure 25 : Modèle AS IS	64
Figure 26 : Modèle TO BE	65
Figure 27 : Structuration en couches de BPMN	66
Figure 28 : Méta-modèle MOF des extensions dataInteroperabilityBarrier et dataInteroperabilityResolved	69
Figure 29 : dataInteroperabilityBarrier	69
Figure 30 : dataInteroperabilityResolved	70
Figure 31 : Modèle AS IS avec les points d'interopérabilité.....	70
Figure 32 : Modèle TO BE avec les points d'interopérabilité.....	71
Figure 33 : Méta-modèle MOF de l'extension performanceMeasurement	72
Figure 34 : performanceMeasurement	73
Figure 35 : Réduction d'activités séquentielles.....	74
Figure 36 : Réduction en OU.....	75
Figure 37 : Réduction en ET	75
Figure 38 : Modèle AS IS avec les mesures de la performance	77
Figure 39 : Modèle TO BE avec les mesures de la performance	77
Figure 40 : Critère de Farmer	87
Figure 41 : Schéma de synthèse des contributions conceptuelles de l'exemple illustratif	92

Figure 42 : Architecture de base de Graphiti	99
Figure 43 : Interface de création de modèles BPMN de BPMN 2 Modeler	100
Figure 44 : Code source XML du diagramme BPMN.....	101
Figure 45 : Interface de développement des plug-ins.....	102
Figure 46 : Classes du modèle ecore	104
Figure 47 : Icône de dataInteroperabilityBarrier	105
Figure 48 : Icône de dataInteroperabilityResolved.....	105
Figure 49 : Icône de performanceMeasurement.....	106
Figure 50 : Interface étendue de BPMN Modeler.....	107
Figure 51 : Catégorie "Interoperability"	107
Figure 52 : Fenêtre de propriétés de performanceMeasurement	108
Figure 53 : Modèle AS IS	118
Figure 54 : Grille d'évaluation AMDEC données	118
Figure 55 : Modèle TO BE	120
Figure 56 : Modèle global de la préparation des commandes	123
Figure 57 : Exemple de bon de préparation (BP)	125

INDEX DES TABLEAUX

Tableau 1 : Synthèse des langages de modélisation	47
Tableau 2 : Les étapes de la méthode GIMSI	54
Tableau 3 : Eléments d'extension de dataInteroperabilityBarrier et dataInteroperabilityResolved	68
Tableau 4 : Eléments d'extension de performanceMeasurement	72
Tableau 5 : Exemple de grille AMDEC	80
Tableau 6 : Exemple de grille AMDEC pour l'amélioration continue	80
Tableau 7 : Principales dimensions de la qualité des données	83
Tableau 8 : Proposition de grille AMDEC qualité	84
Tableau 9 : Grille d'évaluation de la fréquence de non-qualité.....	85
Tableau 10 : Grille d'évaluation de la non détection.....	86
Tableau 11 : Grille d'évaluation de la gravité	87
Tableau 12 : Grille d'évaluation de la criticité de l'exemple illustratif	88
Tableau 13 : Grille de synthèse	89
Tableau 14 : Grille de synthèse de l'exemple illustratif.....	90
Tableau 15 : Grille d'évaluation de la fréquence utilisée chez Onetik	127
Tableau 16 : Grille d'évaluation de la non détection utilisée chez Onetik.....	127
Tableau 17 : Grille d'évaluation de la gravité utilisée chez Onetik	128
Tableau 18 : Grille AMDEC données utilisée chez Onetik	128
Tableau 19 : Grille de synthèse utilisée chez Onetik	129
Tableau 20 : Phase d'amélioration continue.....	130

INTRODUCTION GENERALE

Le contexte économique mondial, basé sur une compétition perpétuelle, exige des entreprises d'acquiescer et de maintenir un système d'information performant pour les aider à prendre les bonnes décisions stratégiques, tactiques et opérationnelles. Un ERP adapté et bien paramétré est aujourd'hui une condition essentielle de la réussite ou même de la survie d'une entreprise. Les données, une fois intégrées dans le système d'information sont disponibles pour les différents départements de l'entreprise à travers des modules ad hoc. Dans cet objectif, les échanges d'informations entre divers systèmes d'information sont de plus en plus courants dans le monde de l'entreprise. L'augmentation de la circulation d'informations électroniques entre partenaires en est un exemple. On parle désormais de réseaux d'entreprises. On peut souhaiter mesurer et mieux orchestrer les échanges formels et informels croissants avec les clients ou avec les fournisseurs. Le développement de la sous-traitance amplifie ce phénomène, ainsi que le rachat et l'intégration d'entreprises.

Ce besoin d'accompagnement de l'échange d'information se présente couramment au sein d'une même entreprise, cependant l'implémentation d'un ERP couvrant l'ensemble des secteurs de l'entreprise se révèle parfois impossible pour différentes raisons. Cette situation amène à greffer au système existant un ou plusieurs modules ou systèmes hexogènes. Il se pose alors le problème crucial de l'interopérabilité des modules, notamment à travers l'interopérabilité des données. Les conséquences d'un manque d'interopérabilité peuvent s'avérer économiquement lourdes pour l'entreprise. Le domaine médical peut être cité en exemple où une telle carence peut entraîner un allongement des délais pour atteindre une ressource de santé, entre les soins, planifier des soins inutiles ou inappropriés, etc. [THE-HIEN 2016] [FIESCHI 2009].

Toutefois, la problématique de l'interopérabilité reste un concept abstrait ou mal identifié pour de nombreux acteurs des entreprises et des organisations. Cet état a pour conséquence de voir la résolution de ce problème comme un objectif non prioritaire. Ceci est particulièrement flagrant dans les PME qui ne disposent pas d'un département informatique qui pourrait repérer cet écueil et en alerter la direction. Afin de remédier à cela, il est nécessaire de comprendre le fonctionnement de l'entreprise d'un point de vue processus, système d'information et décisionnel pour identifier les problèmes et les traiter.

La modélisation des processus d'entreprise est un outil de représentation du système entreprise intéressant. Elle offre la possibilité de décrire graphiquement les pratiques de l'entreprise sous différentes facettes, ce qui permet de mettre en lumière des problématiques sous-jacentes au fonctionnement routinier. Cependant, la représentation de l'interopérabilité est absente dans la grande majorité des langages de modélisation. Il s'agit là d'un manque qu'il nous a paru intéressant de combler.

Toutefois, il nous paraît important de postuler que représenter l'interopérabilité ne suffit pas. Afin de permettre de prendre toute la mesure de la problématique du manque d'interopérabilité, il est essentiel de quantifier les conséquences qu'elle entraîne, et ce au moyen d'indicateurs de performance. A nouveau, bien souvent, les indicateurs ne sont pas présents dans les modèles d'entreprise de haut niveau, c'est-à-dire ceux dont le but est

d'offrir la possibilité de visualiser les processus d'une manière non technique, donc destinés à tous les acteurs intervenant dans un projet et non aux seuls experts dans ce domaine.

C'est pourquoi il nous est apparu nécessaire de faire figurer, en plus de l'interopérabilité, les mesures de la performance afin d'en faire apparaître les conséquences de manière lisible pour tous.

Si l'établissement de l'interopérabilité est une étape fondamentale dans l'optique d'une qualité optimale des données échangées entre différents systèmes d'information, elle ne constitue pas pour autant un objectif final. En effet, une donnée présentant un défaut de qualité avant d'être échangée le conservera après son transfert vers le système cible. Il faut donc compléter l'instauration de l'interopérabilité des données par une action complémentaire visant à améliorer leur qualité.

Nous avons donc ressenti le besoin d'élaborer une méthode de gestion de la qualité des données. Cette méthode offrira la possibilité de mener une action d'amélioration continue dans le domaine de la qualité des données en amont de l'établissement de l'interopérabilité.

Notre travail se décompose en cinq parties.

Dans la première partie, nous allons présenter la problématique de l'interopérabilité et la nécessité que représente la fiabilité des données. Nous allons ensuite souligner le besoin de modélisation et d'évaluation de la performance. Nous mettrons en évidence le besoin de disposer d'outils de modélisation adaptés à cette problématique.

Au cours de la deuxième partie, nous présenterons le travail bibliographique que nous avons réalisé dans le domaine de la modélisation d'entreprise et des méthodes d'indicateurs de performance. Nous ferons une analyse des différents langages et méthodes que nous avons passés en revue et en tirerons la conclusion qu'une extension du langage BPMN 2.0 comprenant des indicateurs de performance est nécessaire, ainsi qu'une méthode de fiabilisation des données.

Dans la troisième partie, nous introduirons les extensions de BPMN que nous proposons. Nous présenterons aussi les méthodes d'agrégation de la performance que nous allons utiliser pour déterminer les valeurs des mesures des différents indicateurs. Nous proposerons également une méthode de gestion de la disponibilité des données qui nous permettra de mettre en place une action d'amélioration continue.

Dans la quatrième partie nous justifierons le choix de l'outil BPMN2 Modeler pour implémenter nos extensions. Puis nous décrirons succinctement cette phase de notre travail.

La cinquième partie sera consacrée à un cas d'étude réalisé au sein de la fromagerie Onetik qui illustrera la pertinence des extensions de BPMN 2.0 ainsi que de la méthode de gestion de la qualité des données que nous proposons. Après avoir brièvement présenté cette entreprise, nous relaterons le déroulement du projet. Nous présenterons quelques-unes des différentes modélisations qui nous ont permis de mettre en relief la présence de problèmes

d'interopérabilité ainsi que la nécessité d'y remédier. Nous exposerons également la mise en place de la méthode d'amélioration continue que nous proposons ainsi que les bénéfices actuels et attendus de cette mise en place.

Enfin, nous terminerons par une conclusion et identifierons les perspectives d'amélioration de ce travail.

Chapitre 1 : PROBLEMATIQUE

<u>1.1 Introduction</u>	14
<u>1.2 Interopérabilité</u>	14
<u>1.2.1 Problèmes d'interopérabilité</u>	15
<u>1.2.2 Niveaux d'interopérabilité</u>	16
<u>1.2.3 Approches de l'interopérabilité</u>	17
<u>1.3 Besoin de modélisation et d'évaluation de la performance</u>	18
<u>1.3.1 Besoin de modélisation</u>	18
<u>1.3.2 L'évaluation de la performance</u>	22
<u>1.4 Besoin de gestion de la qualité des données</u>	25
<u>1.5 Nécessité d'un outil support pour une entreprise</u>	26
<u>1.6 Conclusion</u>	26

1.1 INTRODUCTION

Le volume d'information ne cesse de croître dans les entreprises. Ceci implique notamment des échanges de données entre différents systèmes d'informations de plus en plus nombreux. La connexion et l'orchestration des systèmes d'information passent par l'établissement du principe d'interopérabilité, condition sine qua non d'une collaboration « harmonieuse » des processus de l'entreprise.

Afin d'identifier et d'accompagner la mise en œuvre de l'interopérabilité, une phase de modélisation apparaît nécessaire. Cependant l'interopérabilité est absente dans une grande majorité des langages de modélisation. De plus, des indicateurs de performance intégrés aux modèles permettraient aussi d'appréhender et de quantifier les conséquences du manque d'interopérabilité et les bénéfices induits par leur résolution. Enfin, une fois l'interopérabilité établie, il serait intéressant d'effectuer un suivi de la qualité des données échangées à l'aide d'une méthode ad-hoc.

Les travaux présentés dans ce manuscrit s'inscrivent dans le cadre de l'amélioration de la collaboration entre systèmes d'informations. Une telle collaboration peut être souhaitée entre différentes entreprises ou au sein d'une même entreprise. Cette recherche tend donc plus précisément à améliorer la prise en compte, et donc la résolution, des problèmes engendrés par l'absence d'interopérabilité au niveau des données échangées dans les processus collaboratifs.

Dans ce chapitre, nous commencerons par définir et caractériser le concept d'interopérabilité et ferons apparaître le besoin de manipuler des données fiables. Nous ferons ensuite état du besoin de réaliser des modélisations et des évaluations de la performance afin de résoudre les problèmes qu'entraîne le manque d'interopérabilité. En synthèse, nous identifions le besoin d'une méthodologie de modélisation adaptée au problème décrit ci avant dont nous faisons un état de l'art.

1.2 INTEROPERABILITE

Le contexte économique mondial exige des entreprises d'acquérir et de maintenir un système d'information performant. Un ERP adapté et bien paramétré est aujourd'hui une condition essentielle à la réussite d'une entreprise. Mais les échanges d'information entre divers systèmes d'information sont de plus en plus nécessaires : échanges avec les clients, avec des prestataires, cas de la sous-traitance ou lors du rachat par une autre entreprise, mais aussi au sein d'une même entreprise dans le cas où l'implémentation d'un ERP couvrant l'ensemble des secteurs de l'entreprise se révèle impossible, ce qui induit de greffer entre eux plusieurs modules hexogènes. Se pose alors le problème crucial de l'interopérabilité.

Il nous faut tout d'abord définir ce qu'est l'interopérabilité. Une des définitions les plus courantes de l'interopérabilité nous dit qu'elle représente : « la capacité de deux (ou plus)

systèmes ou composants à échanger des informations et à les utiliser » [CORTEX-CORNAX 2014]. Le réseau d'excellence InterOp NoE (*Interoperability Research for Networked Enterprise Application and Software*) a défini l'interopérabilité comme étant la « capacité d'un système à travailler avec un autre système sans effort de la part de l'utilisateur » [GERACI 1991]. Pour compléter ces définitions on peut dire que l'interopérabilité est « *the ability of systems, natively independent, to interact in order to build harmonious and intentional collaborative behaviors without deeply modifying their individual structure or behavior* » (l'aptitude de systèmes, originellement indépendants, à interagir dans le but de construire des comportements collaboratifs harmonieux et intentionnels sans pour cela modifier radicalement leur structure ou comportement individuels) [TRUPTIL 2010].

De ces définitions, on retiendra la notion de collaboration harmonieuse, sans effort. Dans le domaine qui nous concerne, celui des échanges de données, on peut dire que l'interopérabilité est assurée lorsque l'utilisateur du système cible peut utiliser les données transférées sans avoir à y opérer un quelconque traitement préalable. Assurer l'interopérabilité des données revient donc à rendre invisible aux yeux de l'utilisateur le fait que les données proviennent d'un système d'information différent, tout en garantissant l'exactitude de ces données.

1.2.1 PROBLEMES D'INTEROPERABILITE

L'interopérabilité entre systèmes d'information ne coule pas de source. Les systèmes d'information sont développés par des sociétés qui utilisent différentes technologies et formats de données (non standardisés) et configurés pour répondre aux besoins d'entreprises qui ont des logiques de travail propres. Il n'est donc pas étonnant que de nombreux obstacles surviennent lors de la jonction de systèmes hétérogènes.

Les conséquences du manque d'interopérabilité sont loin d'être négligeables. Un rapport de la Fédération Française du Bâtiment [LEGLISE 2009] de décembre 2009 estime que les coûts directement liés au manque d'interopérabilité représentaient en France 1715 M €, soit 4% des 38 Md € du chiffre d'affaires 2008 de la construction hors logement individuel (en reconnaissant que, du fait des conditions de réalisation de cette étude, il s'agit d'une estimation basse). La résolution de ces problèmes est donc primordiale.

On peut classer les problèmes d'interopérabilité en trois catégories : les problèmes conceptuels, les problèmes technologiques et enfin les problèmes organisationnels [CHEN 2006].

- **Problèmes conceptuels** : ce sont les problèmes liés aux incompatibilités syntaxiques et sémantiques de l'information. Ils concernent la modélisation des informations à un haut niveau d'abstraction comme par exemple un modèle d'entreprise, ainsi que la façon de structurer les informations en vue d'un échange. Les incompatibilités syntaxiques se rencontrent lorsque différents systèmes utilisent des structures différentes pour représenter l'information. Les problèmes sémantiques apparaissent

quand les mots utilisés n'ont pas exactement la même signification dans les différentes structures.

- **Problèmes technologiques** : on rencontre de tels problèmes lors de l'utilisation de l'informatique pour l'échange de données. Ils concernent les incompatibilités de plateformes, d'architecture, de système d'exploitation, etc. Ils portent sur les normes de stockage, d'échange, de présentation et de communication des informations.
- **Problèmes organisationnels** : ils concernent les incompatibilités entre les structures organisationnelles et les techniques de management des différentes entreprises. On peut citer par exemple les domaines suivants : responsabilité, autorisation, confiance, législation (entre différents pays), propriété intellectuelle.

Nous avons décrit plus haut que, dans le cas des échanges de données, l'interopérabilité est assurée si l'utilisateur du système cible peut utiliser les données transférées sans avoir à y opérer un quelconque traitement préalable, donc que les données reçues sont prêtes à l'emploi. Cela veut dire que lorsqu'elle n'est pas assurée, des transformations sont nécessaires afin de pouvoir exploiter ces données. Ces traitements constituent des tâches à non-valeur ajoutée qu'il est important de supprimer. Dans la plupart des cas, il s'agit de ressaisie des données. Or, on le sait, la ressaisie, en plus du temps et des ressources qu'elle implique, est une importante source d'erreurs.

1.2.2 NIVEAUX D'INTEROPERABILITE

L'interopérabilité peut aussi être vue comme relevant de quatre différents niveaux [ATHENA 2005] :

- **Interopérabilité des données** : Il s'agit ici d'articuler différents modèles de données (hiérarchiques, relationnels, etc.) et différents langages de requêtes. Les données sont organisées selon des schémas conceptuels (par exemple des vocabulaires et des structures de données) conditionnés par des applications différentes, stockées sur des bases hétérogènes qui sont elles-mêmes hébergées par des systèmes d'exploitation différents.
- **Interopérabilité des services** : Il faut identifier, composer et faire interagir différentes applications qui ont été conçues et développées de manière indépendante. Pour cela il faut résoudre les problèmes induits par les différences syntaxiques et sémantiques et établir des connections entre des bases de données hétérogènes. Le terme "service" s'applique aux applications informatiques tout autant qu'aux services des entreprises.
- **Interopérabilité des processus** : le but est de faire collaborer différents processus. Un processus est un enchaînement ordonné d'activités à valeur ajoutée, déclenché par un événement et aboutissant à un résultat tangible pour son bénéficiaire. Les processus internes d'une entreprise doivent pouvoir s'interconnecter harmonieusement. Il en va

de même avec les processus entre différentes entreprises (fournisseurs, clients, sous-traitants, etc.).

- **Interopérabilité au niveau métier** : On parle ici du travail en harmonie au niveau des organisations et des entreprises malgré les différences de modes de prise de décision, des méthodes de travail, de législations, de cultures d'entreprise, de stratégies commerciales, etc.

L'interopérabilité des échanges de données relève bien évidemment du niveau de l'interopérabilité des données, mais certains autres niveaux l'impactent également. En effet, articuler entre eux différents services nécessite de s'intéresser aux données échangées et de régler les problèmes de différences syntaxiques et sémantiques. De même, la collaboration de différents processus nécessite des échanges de données (quantités produites, articles concernés etc.). Enfin, l'harmonisation au niveau métier entraîne aussi son lot d'échanges de données.

1.2.3 APPROCHES DE L'INTEROPERABILITE

Une fois les problèmes d'interopérabilité définis, et les niveaux où on peut les rencontrer identifiés, il s'agit de les résoudre. L'EIF (*Enterprise Interoperability Framework*) [CHEN 2006] définit trois approches dans cette optique :

- **L'approche intégrée** : Dans cette approche, les différentes parties utilisent le même format de données et les mêmes outils. Le format choisi n'est pas forcément un standard, mais il doit être respecté par tous les intervenants.
- **L'approche unifiée** : Les différentes parties définissent un format commun au niveau des métadonnées, c'est-à-dire qu'il sert de modèle aux différents formats de données de chacun. Les partenaires doivent alors traduire les données dans le modèle commun avant de les échanger.
- **L'approche fédérée** : Ici, aucun format de données n'est imposé. Chaque partie doit traduire les données reçues à la volée. Il faut tout de même partager une ontologie de référence afin de réaliser les traductions adéquates.

Chacune des approches correspond à des besoins et à des situations différentes. L'approche intégrée est la plus radicale, puisqu'au moins une des deux parties est amenée à refonder totalement son système d'information. Elle est aussi plus longue à mettre en place et exige des efforts financiers importants. Elle est envisageable lors de la fusion de deux organisations ou dans le cas d'une collaboration à long terme. Les deux autres approches offrent plus de souplesse et sont plus à même à répondre à un besoin à court terme et pour une durée qui peut être relativement courte.

La Figure 1 représente l'*Enterprise Interoperability Framework* et résume ce qui vient d'être énoncé.

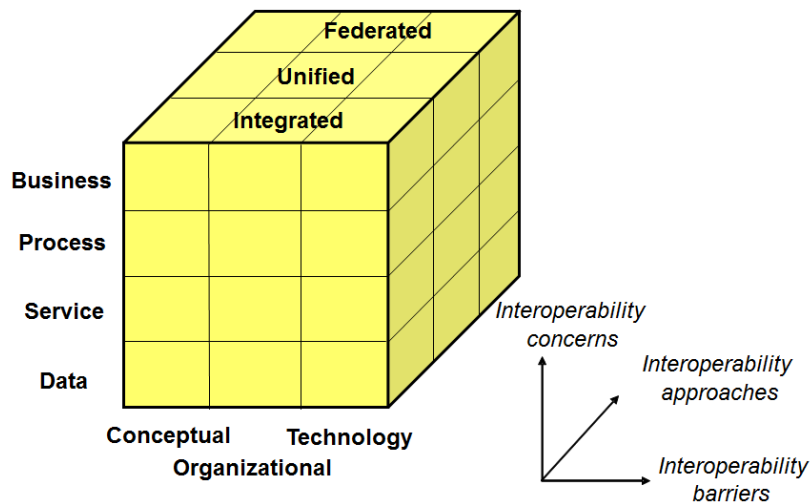


Figure 1 : Enterprise Interoperability Framework

1.3 BESOIN DE MODELISATION ET D'EVALUATION DE LA PERFORMANCE

Résoudre les problèmes inhérents au manque d'interopérabilité nécessite d'effectuer des modifications (au moins pour une des deux parties) au niveau du système d'information, mais aussi au niveau des processus. En effet, on a vu plus haut que l'existence de tels problèmes induisait des tâches à non-valeur ajoutée. Une fois ces problèmes résolus, ces tâches ne seront plus nécessaires, et les ressources qui y étaient précédemment affectées seront libérées, ainsi que le temps nécessaire à l'accomplissement de ces tâches. Le déroulement du processus, une fois l'interopérabilité assurée ne sera donc plus le même.

Il faut donc procéder à de la réingénierie des processus, et donc à une modélisation préalable.

1.3.1 BESOIN DE MODELISATION

1.3.1.1 Qu'est-ce que la modélisation

On trouve dans la littérature de nombreuses définitions de la modélisation. Nous en citerons trois :

Un modèle est une abstraction de la réalité dans le sens où il permet de représenter certains aspects de cette réalité dans un contexte précis. Ceci permet l'utilisation d'une vision du monde plus simplifiée, en évitant la complexité et l'irréversibilité des éléments du monde réel. La modélisation, au sens large, est donc l'utilisation d'une chose à la place d'une autre pour simplifier, faire abstraction de certaines préoccupations [ROTHENBERG 1989].

Un modèle est une représentation d'une abstraction d'une partie du monde réel, exprimée dans un langage de représentation [VERNADAT 1999].

Un modèle est une représentation abstraite et simplifiée (i.e. qui exclut certains détails), d'une entité (phénomène, processus, système, etc.) du monde réel en vue de le décrire, de l'expliquer ou de le prévoir [AUDIBERT 2013].

Une modélisation représente donc la réalité, c'est-à-dire une partie du monde réel, dans un objectif donné : le décrire, l'expliquer et/ou le prévoir. C'est une abstraction du monde réel, mais qui ne représente pas toute la réalité, uniquement certains aspects. En ne représentant que certains éléments de celle-ci, on écarte de la représentation les éléments qui ne sont pas utiles à sa compréhension. Si on prend l'exemple d'une carte routière, les arbres qui font pourtant partie du paysage n'y sont pas représentés. L'explication en est que, ces arbres ne sont pas nécessaires à l'utilisation que l'on fait de ce type de modèle. On évite ainsi la complexité du monde réel. Par contre, les arbres peuvent faire partie d'autres modèles, comme dans le cas d'un projet d'aménagement d'une place en centre-ville, car leur représentation amène une plus-value dans la compréhension.

1.3.1.2 La modélisation d'entreprise

La modélisation d'entreprise est la représentation de la structure et du fonctionnement de l'entreprise sous deux visions liées : la vision globale qui met en évidence les objectifs, les composants et leur interaction, les fonctionnalités, le pilotage de l'évolution et les liaisons avec l'environnement d'une part, et d'autre part la vision locale ou analytique qui est une vision détaillée selon les concepts d'activités et de processus. On modélise donc l'entreprise de manière globale.

La modélisation est largement utilisée au sein des entreprises en tant que méthode pour augmenter la prise de conscience et la connaissance des processus métiers et pour déconstruire la complexité organisationnelle [BANDARA 2005]. Elle permet de décrire graphiquement le déroulement des opérations des processus d'entreprise en représentant les tâches, les événements, les états et des logiques de contrôle de flux. Certains modèles peuvent aussi représenter les données intervenant dans le processus, mais aussi d'autres éléments comme des acteurs externes, des indicateurs de performance, etc. Elle permet aussi de représenter le système d'information ou encore le système décisionnel.

La modélisation doit permettre de représenter « *une partie déterminée d'une entreprise pour en expliquer la structure et le fonctionnement ou pour en analyser le comportement* » [VERNADAT 1999]. On ne modélisera donc que la ou les parties (systèmes) de l'entreprise que l'on souhaite étudier. De même, on n'intégrera dans la modélisation que les éléments utiles à la compréhension du phénomène observé, sans « polluer » la représentation par des éléments qui n'ont pas besoin d'être pris en considération. Ainsi, on sera plus à même de décrypter le fonctionnement des processus et donc d'en imaginer les améliorations possibles,

car au-delà de la compréhension, il s'agit bien d'apporter des améliorations à l'existant dans le but de l'optimiser.

Une modélisation est une abstraction du monde réel, et donc par cela subjective. En ne représentant que les éléments que nous sélectionnons du processus observé, nous en apportons un éclairage nouveau. Nous allons ainsi pouvoir l'observer sous l'angle qui nous convient et ainsi nous focaliser uniquement sur certaines activités (généralement celles que nous souhaitons optimiser). Il est en effet inutile de mettre en relief ce qui fonctionne correctement si on ne souhaite pas y apporter d'améliorations.

Nous retiendrons aussi le fait que la modélisation doit permettre de communiquer, d'expliquer à différents acteurs le fonctionnement de l'entreprise ou de la partie de l'entreprise représentée. Cette dimension est essentielle, car dans un projet, tout le monde doit aller dans le même sens. Or, dans une entreprise il est classique que les personnels des différents départements aient des points de vue différents, si ce n'est carrément opposés. De même, selon le poste occupé (direction, chef d'atelier, opérateur, etc.) la vision des choses différera. Pour que tout le monde aille dans la même direction, il faut avant tout que tout le monde comprenne la même chose. Il est donc primordial de décrire clairement (et donc simplement, c'est-à-dire de manière « non technique ») le fonctionnement de l'entreprise (du moins dans le domaine considéré). Les modèles doivent offrir cette possibilité.

Il est possible de synthétiser les techniques de modélisation en deux groupes [PHALP 1998] :

- Les techniques de modélisation de haut niveau : ces techniques graphiques intuitives ont pour but de représenter et comprendre les processus concernés par un projet et de permettre d'échanger au sujet des améliorations qui peuvent y être apportées. Leur public potentiel est large et va au-delà des analystes métiers, un public qui ne se préoccupe pas forcément de la manière dont l'implémentation des solutions sera réalisée. Un de leurs principaux objectifs est la communication entre divers publics.
- Les techniques de modélisation de bas niveau : leur finalité est l'analyse des processus par les analystes métiers. Elles sont beaucoup plus prescriptives et ne servent en général à communiquer qu'avec les techniciens qui implémenteront la solution.

Dans le cadre de cette thèse, nous nous focaliserons sur la modélisation de haut niveau car une de nos préoccupations majeures est la communication avec un public non expert.

La modélisation des processus au plus haut niveau doit donc permettre aux différents partenaires de comprendre l'articulation des différents processus de manière simple et intuitive. Or, la définition des besoins pour les utilisateurs n'est pas nécessairement orientée collaboration. Les utilisateurs n'appréhendent que rarement les différences entre les processus inter-organisations et les processus internes.

Les langages de modélisation peuvent représenter divers aspects complémentaires d'un processus. [CURTIS 1992] a défini quatre vues :

- La vue fonctionnelle représente les activités qui composent le processus.
- La vue comportementale exprime quand et comment sont exécutées les activités.
- La vue organisationnelle modélise les acteurs du processus et leur position dans l'organisation.
- La vue informationnelle représente les données intervenant dans le processus.

La vue comportementale est une modélisation d'un contrôle de flux entre les activités. On y décrit la chronologie de l'exécution des activités, leur flux (séquentiel ou parallèle), les points de synchronisation entre activités ou, au contraire, les points de disjonction. Un des grands intérêts de cette vue est qu'une modélisation de ce type peut servir de base à une simulation puisqu'elle autorise la représentation de la dimension temporelle.

La modélisation des processus métiers inter-organisations (cross-organizational business processes ou CBP) a pour but de décrire les interactions entre différentes organisations [ATHENA 2005], mais aussi entre différents systèmes au sein d'une même organisation.

1.3.1.3 Que représenter dans la modélisation d'entreprise ?

Nous venons de voir qu'il ne faut représenter que les domaines de l'entreprise qu'on souhaite améliorer, en édulcorant les éléments qui sont hors du champ du projet d'amélioration. Mais modéliser l'existant en soulignant les défauts ne suffit pas. Dans un processus de réingénierie, il est nécessaire de modéliser, en plus du système existant (ou AS IS), le système cible (TO BE), celui que l'on projette de mettre en place. On va donc réaliser deux types de modèles :

- **AS IS** : cette modélisation permet de faire l'état des lieux du fonctionnement actuel du système. Elle va permettre de pointer du doigt les points qu'il est nécessaire d'améliorer.
- **TO BE** : cette modélisation doit décrire un état idéal, c'est-à-dire un état dans lequel les problèmes détectés dans le AS IS qui empêchaient la totale réalisation des objectifs stratégiques sont résolus.

Dans le cas qui nous préoccupe, il est essentiel de représenter l'interopérabilité. Les échanges de données présentant un problème d'interopérabilité devront apparaître dans le modèle AS IS en tant que points à améliorer. Le fait de les rendre visibles permettra de communiquer à toutes les parties concernées par le projet, et notamment aux décideurs, la nécessité de leur résolution. On fera apparaître les échanges de données pour lesquels l'interopérabilité a été établie dans le modèle TO BE afin de signifier que la résolution de l'interopérabilité est un des éléments constitutifs d'un état idéal.

Cependant, faire apparaître un problème d'interopérabilité à un public non connaisseur ne suffit pas. Pour appréhender l'absolue nécessité de sa résolution, il faudrait quantifier les

conséquences de l'existence du problème et par là, les gains qu'entraînera sa résolution. Il faut donc faire apparaître, sur les modèles, l'évaluation de la performance que nous traitons dans le paragraphe suivant.

1.3.2 L'EVALUATION DE LA PERFORMANCE

1.3.2.1 La notion de performance

La littérature concernant la performance est foisonnante, mais aucune définition ne fait l'unanimité. Tout d'abord, parce que la notion de performance a évolué à travers les différentes phases du contexte industriel [CLIVILLE 2004]. Ensuite parce que cette notion est diversement appréciée selon l'optique à travers laquelle on l'aborde : « *La performance désigne la réalisation des objectifs organisationnels, quelles que soient la nature et la variété de ces objectifs. Cette réalisation peut se comprendre au sens strict (résultat, aboutissement) ou au sens large du processus qui mène au résultat (action). La performance est multidimensionnelle, à l'image des buts organisationnels ; elle est subjective et dépend des référents choisis (buts, cibles)* » [BOURGUIGNON 1997].

Selon A. Neely, « *Performance is the sum of all the processes that will lead managers to taking appropriate actions in the present that will create a performing organization in the future (i.e. one that is effective and efficient)* » [NEELY 2007]. Ainsi, l'appréciation de la performance va permettre de prendre des décisions concernant le pilotage de l'entreprise dans l'optique de l'atteinte d'objectifs. Il s'agira donc de mesurer cette performance. On utilise à cet effet des indicateurs de performance.

1.3.2.2 Les indicateurs de performance

Dans le but de modifier et améliorer un processus, il est nécessaire de connaître quelles sont les composantes de ce processus à optimiser. Ceci ne peut être correctement apprécié sans un ensemble d'indicateurs simples, mesurables, atteignables, réalistes et temporellement définis (SMART) [CPC 1997].

Les indicateurs de performance sont utilisés pour mesurer la performance. Selon la définition de l'Afnor : « *Un Indicateur de performance est une donnée quantifiée, qui mesure l'efficacité ou l'efficience de tout ou partie d'un processus ou d'un système, par rapport à une norme, un plan ou un objectif déterminé et accepté, dans le cadre d'une stratégie d'entreprise* » [CLIVILLE 2004]. Pour Lorino, un indicateur de performance est « *une information devant aider un acteur, individuel ou plus généralement collectif, à conduire le cours d'une action vers l'atteinte d'un objectif ou devant lui permettre d'en évaluer le résultat* » [LORINO 2001]. D'après Doumeingts, « *Un Indicateur de Performance est une donnée quantifiée qui mesure l'efficacité des actions sur les variables de décision des décideurs et le degré d'atteinte de leur objectifs définis par rapport à la stratégie d'entreprise* » [DOUMEINGTS 98].

Il s'agit donc de sélectionner et collecter les données relatives aux problèmes détectés (ou non) ainsi qu'aux objectifs de performance. Les décideurs utiliseront ensuite ces données pour piloter l'entreprise dans le but d'en obtenir la performance optimale.

Afin que ce pilotage soit le plus précis possible, il convient de choisir les indicateurs qui donneront les résultats les plus en adéquation avec les objectifs de l'entreprise.

Historiquement, les premiers indicateurs étaient uniquement d'ordre financier (comptabilité analytique et contrôle budgétaire). Le coût de production était mesuré dans l'optique de le réduire au minimum. Mais avec l'évolution du contexte industriel, on s'est aperçu que ce seul critère ne suffisait pas à comprendre, et donc améliorer, la performance globale de l'entreprise [CLIVILLE 2004]. Comme le dit C. Marmuse, « *La performance revêt (...) des aspects multiples, sans doute convergents, mais qui méritent d'être abordés dans une logique plus globale que la seule appréciation de la rentabilité pour l'entreprise ou pour l'actionnaire* » [MARMUSE 1997]. Dans un contexte de marché concurrentiel, la prise en compte de nouveaux critères a donc été considérée.

Le délai ou le temps est une des notions les plus courantes parmi les indicateurs de performance. Les délais de conception, de réalisation, de livraison, etc. doivent être réduits à leur minimum afin de satisfaire les clients. Ils permettent ainsi de les fidéliser et de gagner de nouveaux contrats.

La qualité est le troisième paramètre le plus présent dans les systèmes d'indicateurs de performance. En effet, il est inutile de produire moins cher et plus rapidement si une proportion importante de la production est à mettre au rebut ou si les clients, déçus, ne renouvellent pas les commandes.

Il existe de nombreux autres domaines de référence pour définir des indicateurs de performance selon le domaine de compétences ou encore les objectifs de l'entreprise. On citera par exemple la flexibilité, l'environnement, le service, etc. Ce qu'il faut retenir, c'est qu'il n'existe pas de système d'indicateurs de performance universel, mais que ces indicateurs doivent être sélectionnés en fonction de la stratégie et du champ d'activité de l'entreprise. « *Toute organisation est libre de définir des indicateurs pertinents pourvu qu'ils arrivent à une mesure et un pilotage efficaces de la performance* » [RAVELOMANANTSOA 2009].

1.3.2.3 Choix des indicateurs de performance

Comme on vient de le voir, il n'existe pas de système d'indicateurs de performance universel. Dans le cas de la résolution des problèmes d'interopérabilité, nous devons donc définir quels seront les domaines de performance qui seront pertinents.

De nombreuses méthodes ont été élaborées pour définir un système d'indicateurs de performance. Nous mentionnerons le Balanced Scorecard, la méthode ECOGRAI ou encore le Process Performance Measurement System [RAVELOMANANTSOA 2009]. La plupart de ces méthodes retiennent (de manière non exhaustive) les domaines de performance suivants : le coût, la qualité et le délai. Toutefois, ces méthodes ont pour but la mise en place d'un

système d'indicateurs qui mesure la performance globale de l'entreprise. Or notre but est différent. La résolution des problèmes entraînés par l'absence d'interopérabilité des données, n'implique pas la remise en cause totale du système de production. C'est dans ce cadre que nous devons définir nos indicateurs de performance. Nous devons nous concentrer sur les domaines de performance concernés par l'interopérabilité des données. Ces indicateurs concerneront uniquement les tâches impactées par les échanges de données comportant des problèmes d'interopérabilité (dans l'existant) ou pour lesquels ces problèmes ont été résolus (dans le modèle cible). La question que nous devons nous poser est donc : parmi les tâches impliquées par un échange de données présentant un problème d'interopérabilité, quels sont les domaines dont nous devons connaître la performance dans l'optique de prendre les bonnes décisions quant à l'optimisation de ces tâches par la résolution de ce problème.

Nous avons retenu les domaines suivants :

1.3.2.3.1 Le coût

Le coût est le montant financier qui permet de réaliser une tâche ou un ensemble de tâches. C'est un critère essentiel pour le management dans le monde de l'entreprise. Le coût peut être décomposé en divers coûts : coût initial (coût d'achat des matières premières, coût opératoire (d'utilisation), coût de maintenance, etc.

1.3.2.3.2 Le délai

Le délai est devenu un domaine de performance essentiel dans les systèmes d'indicateurs. Il s'agit du temps mis pour réaliser chaque tâche. Dans une économie concurrentielle, les délais de conception, de réalisation, de livraison, de service, etc. doivent être réduits à leur minimum pour répondre aux exigences des clients. Ce temps va varier en fonction du niveau d'établissement de l'interopérabilité.

1.3.2.3.3 La qualité

La norme ISO 9000 définit la qualité comme l'« aptitude d'un ensemble de caractéristiques intrinsèques à satisfaire des exigences ». Nous considérerons dans ce travail la qualité des données échangées. Les données sont de qualité si elles satisfont aux exigences des utilisateurs. Selon Batini, une donnée est de qualité si elle est exacte (correcte, fiable et certifiée, si elle correspond à la valeur du monde réel), complète (toutes les données nécessaires sont présentes), cohérente (toujours présente dans le même format) et actuelle (non obsolète). [BATINI 2009].

1.3.2.3.4 La disponibilité

La définition de la disponibilité selon l'AFNOR est la suivante « *Dans des conditions données d'utilisation, la disponibilité d'un équipement E est définie comme son aptitude à être en état d'accomplir une fonction requise, à un instant t donné ou pendant un intervalle de temps donné, en supposant que la fourniture des moyens extérieurs nécessaires soit assurée* »

[AFNOR, 1988]. Cette définition, transposée au domaine des données pourrait être simplifiée ainsi : la disponibilité répond au besoin suivant : offrir la bonne donnée au bon moment, c'est-à-dire au moment de son utilisation dans le processus ou dans la décision. Nous avons dit précédemment qu'établir l'interopérabilité représentait une nécessité. Mais l'interopérabilité n'est pas une fin en soi, ce n'est qu'un moyen de s'assurer que les données nécessaires à un processus sont présentes et n'entraînent aucun surplus de travail pour la personne qui les reçoit. Une interopérabilité établie nécessite donc la disponibilité des données. De la même manière, la question de la qualité d'une donnée non disponible ne se pose pas : si la donnée n'est pas présente, elle ne peut être de qualité. On peut donc dire que la disponibilité est une composante de la qualité, voire même un prérequis. Dans le cas classique d'un problème d'interopérabilité entraînant une ressaisie des données, si une donnée n'est pas disponible, la personne chargée de la ressaisie devra aller à sa recherche. Cette recherche prendra un temps plus ou moins long selon la nature de la donnée en cause, ce qui augmentera d'autant la tâche à non-valeur ajoutée que nous souhaitons éliminer. On ne pourra donc pas établir l'interopérabilité des données tant qu'on n'aura pas assuré leur disponibilité car on n'aura pas supprimé la tâche à non-valeur ajoutée. En conséquence, tenter d'établir l'interopérabilité des données sans assurer leur disponibilité s'avère une gageure. Il est donc essentiel de mesurer, outre la qualité, la disponibilité des données.

1.4 BESOIN DE GESTION DE LA QUALITE DES DONNEES

Assurer l'interopérabilité des données est une condition sine qua non d'une meilleure qualité des données. En effet, elle permet de conserver la qualité des données après leur échange entre différents systèmes, notamment en évitant leur ressaisie. Toutefois, la qualité des données n'en sera pas pour autant optimale. En effet, si une donnée présente un défaut de qualité avant les différents échanges, ce défaut perdurera tout au long du processus.

Les modélisations et les mesures de la performance que nous venons de définir au cours des précédents paragraphes vont nous permettre d'avoir une vue globale de la qualité des données. Même si cette vue globale est indispensable, elle nous permet d'appréhender uniquement l'existence et la nature du problème de la qualité des données car les valeurs de la qualité des différentes données sont fusionnées dans une valeur unique. De ce fait, notre approche est globale et qualitative. Or, si la perception d'un problème est une condition sine qua non de sa résolution, une connaissance des causes de ce problème en est une autre.

Il nous faut donc compléter notre proposition par un outil supplémentaire qui nous permettra d'avoir une approche détaillée et quantitative. Il devra nous permettre d'avoir une vision détaillée de chacune des données impliquées dans le processus. On sera ainsi plus à même d'en déterminer les motifs de la non-qualité dans le but d'élaborer une action corrective.

1.5 NECESSITE D'UN OUTIL SUPPORT POUR UNE ENTREPRISE

Après avoir exprimé le besoin d'effectuer des modélisations pour résoudre les problèmes d'interopérabilité et déterminé quels étaient les macro indicateurs de performance nécessaires à la résolution de ces problèmes, nous avons établi qu'il était aussi nécessaire de disposer d'un outil de gestion de la qualité des données.

Nous devons maintenant nous pencher sur la question des fonctionnalités que doit offrir le langage de modélisation que nous utiliserons pour réaliser ces modèles.

En tout premier lieu, les modèles que nous allons réaliser doivent être de haut niveau. En effet, comme nous l'avons dit plus haut, les langages de haut niveau ont pour but de représenter et d'aider à comprendre les processus, mais aussi de permettre d'échanger avec un public hétérogène qui ne se soucie pas de la manière dont l'implémentation des solutions sera réalisée. A l'inverse, les langages de bas niveau étant destinés à communiquer avec les ressources techniques qui implémenteront la solution ne répondent pas à notre besoin.

Ensuite, ces modèles doivent permettre de représenter les points d'interopérabilité. Si l'on souhaite communiquer sur ce sujet dont seuls les initiés ont connaissance, sa visualisation représente une condition sine qua non. Ils doivent donc évidemment aussi permettre de représenter les échanges de données de manière claire.

Les modèles doivent aussi permettre de représenter les mesures de la performance dans les domaines du coût, du délai, de la qualité et de la disponibilité. On ne peut pas prendre la décision de résoudre les problèmes d'interopérabilité si on ne réalise pas leurs conséquences. Et le meilleur moyen pour cela est de disposer d'indicateurs affichant de manière claire et évidente les mesures de la performance en interopérabilité.

Enfin, afin de laisser la porte ouverte à des travaux futurs en termes de simulation des processus, nous souhaitons que le langage de modélisation permette de réaliser des modèles selon une vue comportementale.

En résumé, nous avons besoin d'un langage de modélisation de haut niveau qui permet de représenter l'interopérabilité ainsi que les mesures de la performance au niveau du coût, du délai, de la qualité et de la disponibilité.

Pour ce qui est de l'outil de gestion de la qualité, il doit offrir la possibilité de visualiser chacune des données impliquées dans le processus afin de connaître les motifs de non-qualité. Cette visualisation doit pouvoir être possible pour tous les échanges de données dans lesquels ces données sont impliquées, ainsi que de manière globale pour la totalité du processus.

1.6 CONCLUSION

Dans cette première partie, nous avons appréhendé l'enjeu que représente l'interopérabilité dans les échanges de données entre différents systèmes d'information, que ce soit entre différentes organisations ou au sein d'une même organisation. Il apparaît clair que l'absence d'interopérabilité impose la réalisation de tâches à non-valeur ajoutée afin de pouvoir adapter, transformer et exploiter les données reçues pour le système cible, la réalisation de ces tâches étant pénalisante à double titre : elle mobilise des ressources et elle est source d'erreurs qui vont se répercuter dans la suite du processus. L'établissement de l'interopérabilité dans le cadre de l'échange des données entre systèmes différents est donc primordial. Il est déjà intéressant pour l'entreprise de pouvoir cartographier ces points de non interopérabilité afin recenser globalement ces points bloquants ou contraignants du processus informationnel de l'entreprise.

Nous avons vu par la suite que la modélisation était un passage indispensable dans la résolution de ces problèmes. En effet, le modèle de l'existant (AS IS) sert à mettre en évidence aux yeux des différents acteurs les problèmes occasionnés par l'absence d'interopérabilité, et celui du projet à bâtir (TO BE) a pour but d'exposer les avantages que présente cette résolution.

Nous avons aussi expliqué l'intérêt d'utiliser des indicateurs de performance couvrant les domaines du coût, du délai, de la qualité et de la disponibilité afin de faire réellement appréhender les conséquences directes induites par les problèmes d'interopérabilité ainsi que les bénéfices qu'entraîneraient leur résolution.

Nous avons également établi qu'un outil supplémentaire pouvait être nécessaire afin de répertorier et gérer la qualité des données et cela de manière individuelle pour chaque tâche.

Enfin, nous avons défini les différentes caractéristiques que devait posséder un langage de modélisation afin de nous offrir la possibilité de mettre en œuvre la résolution des problèmes d'interopérabilité.

Dans le chapitre suivant, nous allons réaliser un état de l'art au sujet des langages de modélisation ainsi que des méthodes de l'évaluation de la performance afin de sélectionner les langages et méthodes existants qui nous aideront dans la résolution de ces problèmes.

Chapitre 2 : ÉTAT DE L'ART

- 2.1 Introduction 29**
- 2.2 Langages de modélisation de processus..... 29**
 - 2.2.1 BPMN 30
 - 2.2.2 ARIS 32
 - 2.2.3 IDEF 35
 - 2.2.4 GRAI..... 39
 - 2.2.5 CIMOSA 44
 - 2.2.6 Synthèse des langages de modélisation 46
- 2.3 Méthodes de mesure de la performance 48**
 - 2.3.1 Tableau de bord..... 48
 - 2.3.2 Balanced Scorecard..... 49
 - 2.3.3 OVAR 51
 - 2.3.4 GIMSI..... 53
 - 2.3.5 PRISM 54
 - 2.3.6 ABC/ABM/ABB..... 57
 - 2.3.7 Synthèse des méthodes de mesure de la performance 58
- 2.4 Conclusion..... 59**

2.1 INTRODUCTION

Dans la première partie de cette thèse, nous avons commencé par définir la problématique globale de l'interopérabilité dans les échanges de données à travers les problèmes, les niveaux et les approches de l'interopérabilité. Nous avons ensuite établi le besoin de représenter l'interopérabilité dans les modélisations, puis celui de mesurer la performance des activités concernées par de tels échanges dans le but d'aider à appréhender la nécessité de résoudre les problèmes engendrés par le manque d'interopérabilité. Nous avons à ce propos défini les domaines de la performance qu'il nous faudrait mesurer et représenter dans les modèles. Nous avons enfin spécifié les caractéristiques que devaient posséder les langages de modélisation pour nous aider à régler les problèmes d'interopérabilité.

Dans ce chapitre, nous allons réaliser une revue de la littérature afin d'établir un état de l'art des langages de modélisation et des méthodes d'évaluation de la performance. Cette revue de la littérature a pour but la sélection du langage que nous allons utiliser pour réaliser les modèles dont nous avons besoin dans notre démarche d'établissement de l'interopérabilité dans les échanges de données.

Nous allons tout d'abord recenser plusieurs langages de modélisation des processus en entreprise en nous penchant plus particulièrement sur les possibilités qu'ils offrent dans la représentation des données. Ensuite nous en ferons une synthèse.

Dans une seconde partie, nous recensons différentes méthodes d'évaluation de la performance.

Enfin, sur la base de cet état de l'art, nous concluons sur les fondements de la solution que nous allons développer dans la suite de nos travaux.

2.2 LANGAGES DE MODELISATION DE PROCESSUS

Il existe de nombreux langages de modélisation, répondant à des besoins spécifiques et donnant un point de vue du système à modéliser. Nous allons en passer quelques uns en revue afin de déterminer celui que nous utiliserons dans la suite de nos travaux. Ce sont tous des langages de référence qui permettent d'effectuer de la modélisation de processus. La modélisation des processus offre la possibilité de visualiser les enchaînements (ou séquences) de tâches (humaines ou automatisées). Certains de ces langages sont graphiques facilitant leur utilisation par un acteur non expert.

Rappelons ce que nous avons défini comme prérequis pour le langage qui nous servira à faire les modèles :

- Être un langage de haut niveau afin que tous les acteurs (experts/non experts de modélisation) puissent appréhender les modèles sans difficulté.

- Permettre de représenter les points d'interopérabilité, et donc les échanges de données. Pouvoir quantifier cette notion, pour hiérarchiser les priorités d'amélioration.
- Permettre de réaliser des modèles selon une vue comportementale. Ceci permettant d'envisager éventuellement l'exécution de ces modèles.

2.2.1 BPMN

2.2.1.1 Présentation

Les premiers pas pour la modélisation standardisée de processus métiers informationnels en vue de leur informatisation sont attribués à la Workflow Management Coalition qui a proposé le langage XPDL [WfMC 1999]. Plusieurs initiatives ont permis de quantifier les activités pour observer le comportement de ces processus et par exemple les rendre exécutables, ex. [ZACHAREWICZ 2008]. Cependant, XPDL étant un langage textuel, la Business Process Management Initiative (BPMI) a créé en 2004, en supplantant le précédent, le langage graphique BPMN (Business Process Model and Notation). BPMN est aujourd'hui maintenu par l'Object Management Group [OMG 2003]. Il a depuis été adopté en tant que standard ISO (ISO 19510:2013) et constitue un standard *de facto* de la pratique professionnelle de la modélisation [BRAUN 2014]. Il est de plus le langage de modélisation le plus utilisé par les entreprises selon l'étude de P. Harmon (64 % d'utilisation) [HARMON 2016].

BPMN a été créé à l'origine pour donner une représentation graphique à BPEL [ANDREWS 2003]. Une exigence soulignée lors de sa création était le fait qu'il soit aisément compréhensible tous les types de publics (analystes métiers, utilisateurs des processus ou direction), ce qui en fait un outil de dialogue et d'échange tout en étant d'un usage intuitif.

BPMN permet de réaliser des diagrammes basés sur la technique du flowcharting, elle-même conçue pour représenter des modèles de processus métiers. Il permet donc de réaliser des modèles selon une vue comportementale.

Les différentes entités prenant part au processus (entreprises, services, etc.) sont représentées par des piscines (*pools*), elles-mêmes divisées en couloirs (*lanes*) qui figurent les participants (acteurs ou outils). A l'intérieur de ces couloirs, les activités sont reliées de manière séquentielle. Tout processus doit commencer par au moins un évènement de début (*start event*) et se terminer par au moins un évènement de fin (*end event*). Des branchements peuvent signifier une condition (respectée ou non) ou une réunification.

BPMN permet aussi de représenter les échanges de données entre différentes activités.

Les artefacts ont pour but d'enrichir les modèles par des informations supplémentaires. Ils sont constitués par :

- les groupes : éléments visuels dont le but est de regrouper des activités ;
- les annotations : éléments d'affichage de texte informationnel.

Parmi les nombreux autres éléments de BPMN, nous pouvons citer l'envoi et la réception d'évènement (messages, signaux...), les boucles (pour rééditer un comportement) ou encore les minuteries, pour vérifier qu'une activité s'est bien déroulée dans le temps imparti et bloquer l'exécution d'un processus tardif.

La Figure 2 illustre un exemple de modèle BPMN. Il nous présente un processus de prise de repas au restaurant, de la prise de commande au paiement :

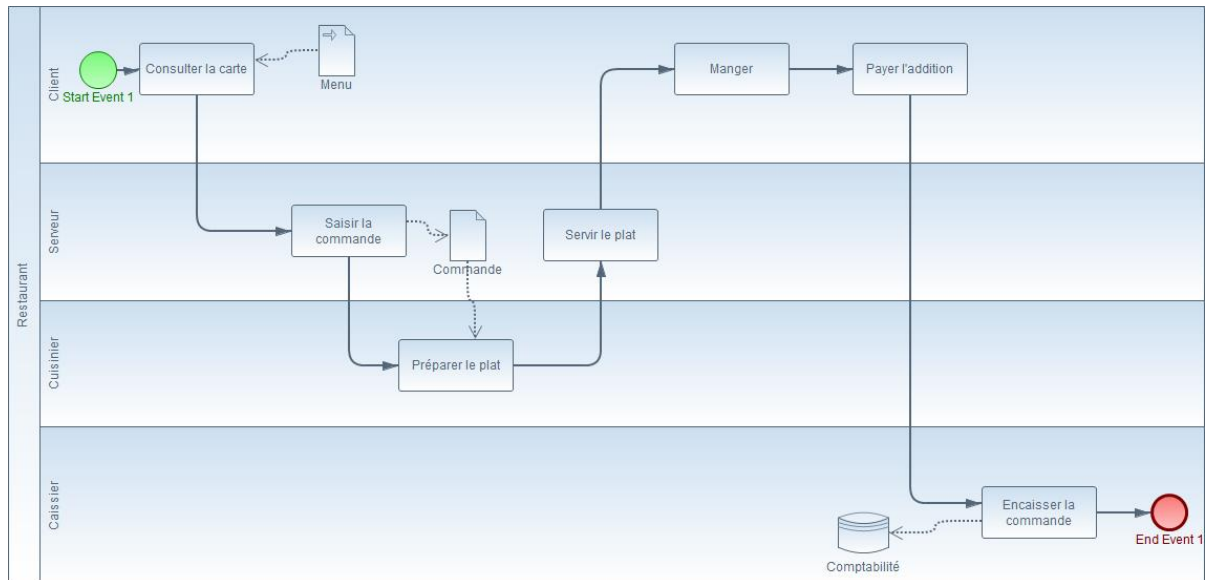


Figure 2 : Exemple de modèle BPMN

2.2.1.2 Perspective informationnelle de BPMN

En BPMN, l'élément qui permet de modéliser tout type d'entité informationnelle échangé dans le modèle, indépendamment de sa nature (document électronique, papier, etc.) est le Data Object. Il est à noter que le Data Object est enchainé à partir d'une tâche par une flèche pointillée représentant l'association. Il ne décrit par un flux séquentiel mais bien un flux informationnel et fonctionnel. Il peut cependant être parfois parallèle au flux de séquence. Cette nuance est importante en BPMN. Il existe cinq déclinaisons du Data Object :

- Data Object : information passant par le processus, voir la Figure 3 ;



Figure 3 : Data Object

- Data Collection : collection d'informations passant par le processus, voir la Figure 4 ;



Figure 4 : Data Collection

- Data Input : information entrant dans le processus, voir la Figure 5 ;



Figure 5 : Data Input

- Data Output : information sortant du processus, voir la Figure 6 ;

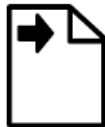


Figure 6 : Data Output

- Data Store : magasin de données utilisées par le processus, voir la Figure 7.



Figure 7 : Data store

Dans BPMN 2.0, les Data Objects deviennent des éléments de première classe, au même niveau que les tâches ou les activités. C'est une grande évolution par rapport à BPMN 1.x où ils n'étaient que des artefacts, c'est-à-dire de simples annotations. Les Data Objects sont à présent représentés directement dans le modèle BPMN de manière claire et intuitive.

BPMN 2.0 ne permet pas nativement de représenter l'interopérabilité, mais il est à noter qu'il jouit d'un système d'extension qui permet l'ajout d'éléments et attributs additionnels pour représenter des particularités inhérentes à un domaine spécifique ou pour combler une lacune de ce langage à un niveau plus général.

2.2.2 ARIS

ARIS (Architecture of Integrated Information Systems) a été développée dans les années 1990 par le professeur August-Wilhelm SCHEER. Elle a été proposée pour les phases d'analyse et de définition des besoins d'un système d'information de gestion [SCHEER 1994].

ARIS se base sur une approche :

- Multi-niveaux : conceptuel, technique, physique (on part d'une généralité pour ensuite entrer dans le détail) ;
- Multi-vues : organisation, données, contrôle, fonctions, produits/services.

La Figure 8 illustre cette approche.

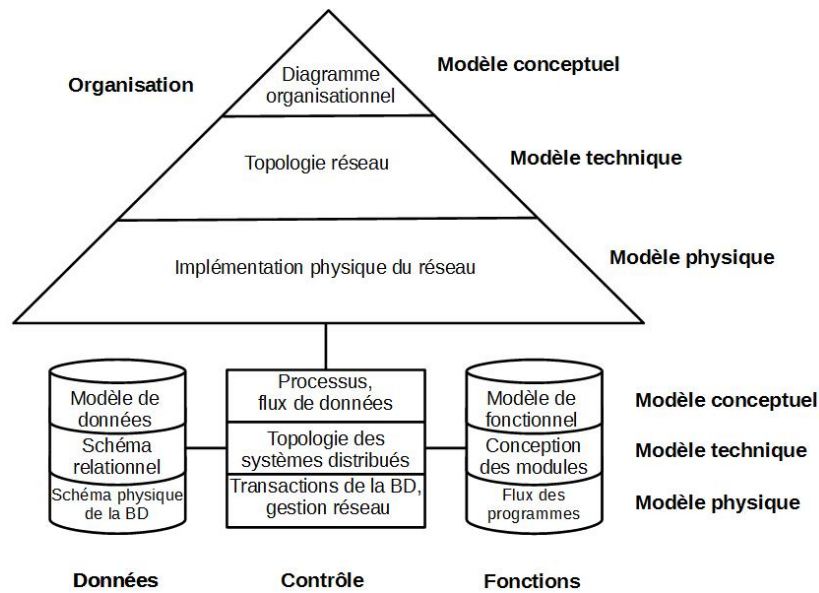


Figure 8 : Approche ARIS

En sortie du contrôle s'ajoutent les produits/services.

Chaque vue va répondre à une question liée au processus modélisé :

- Fonctions : « *What ?* » : par quoi est réalisé le processus. Pour y répondre, on peut utiliser un diagramme en arbre représentant les activités de l'entreprise ;
- Organisation : « *Who ?* » : chaque processus va être géré par un ou plusieurs acteurs. Ils peuvent donc être représentés sous la forme d'un organigramme, détaillant l'entreprise, les services, les responsables de chaque service, etc. de manière hiérarchique ;
- Produits/Services : « *Why ?* » : pour quelle raison ce processus existe-t-il, à quoi sert-il. On détaille à cette étape les produits et/ou les services concernés par ce processus, via un diagramme en arbre par exemple ;
- Données : « *Which information ?* » : quelles seront les informations nécessaires pour réaliser ce processus. Pour y répondre, nous pouvons utiliser l'ERM (*entity-relationship model*, ou modèle entité-association) ;
- Contrôle : « *How ?* » : comment sera géré le processus. Pour cela, il est possible d'exploiter l'EPC (*Event-driven Process Chain*, ou chaînes de processus événementielles). La Figure 9 nous offre un exemple d'EPC.

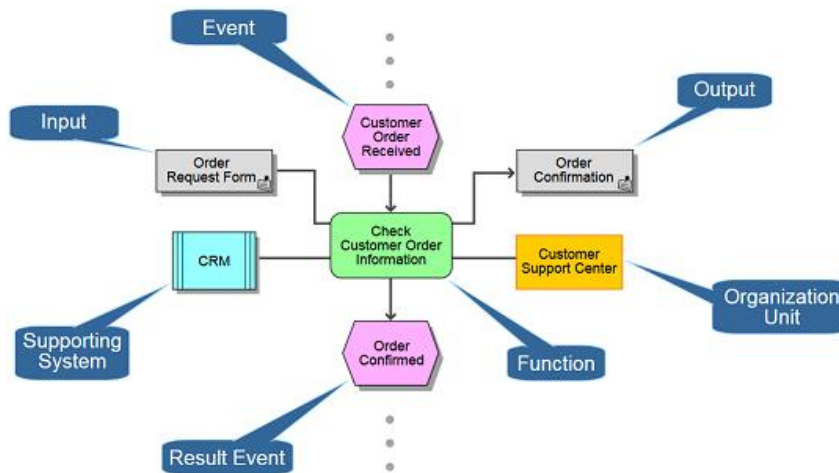


Figure 9: Exemple de chaîne de processus événementielle

2.2.2.1 Chaînes de processus événementielles (EPC)

Les fonctions et les événements permettent de représenter les processus sous la forme d'*Event Driven Process Chain*. Tout comme ARIS, EPC est issue des travaux d'August-Wilhelm SCHEER. D'autres modèles sont proposés par cette méthode dont une vue plus globale des processus permettant de représenter la chaîne de valeur (Value Chain).

La méthode EPC est utilisée dans le cas de la gestion du processus, et plus particulièrement dans l'implémentation d'ERP (notamment SAP). Sa modélisation représente donc des éléments d'un processus fonctionnant parallèlement et alternativement. Il permet donc de réaliser des modèles selon une vue comportementale.

En résumé, l'approche ARIS se base sur plusieurs méthodes de modélisation (ERM, EPC, etc.) pour représenter de la façon la plus détaillée possible les processus d'entreprises. Ainsi, ARIS se rapproche plutôt d'une méthodologie pour modéliser, qui englobe donc plusieurs méthodes de modélisations.

2.2.2.2 Perspective informationnelle de ARIS

Au sein de la méthode ARIS, on utilise ERM (Entity Relationship Model ou Modèle Entité-Association) pour modéliser les données. Ce modèle est proche du modèle conceptuel des données de Merise ou encore des diagrammes de classe UML. Son principe est de distinguer les entités (un client, un produit, une commande...), les attributs (nom, prénom...) et les associations (« contient » entre une entité « commande » et une entité « produit ») pour mettre en œuvre une représentation des données et de leur gestion. La Figure 10 nous montre un exemple simple d'ERM.

ARIS ne dispose pas de mécanisme de représentation de l'interopérabilité.

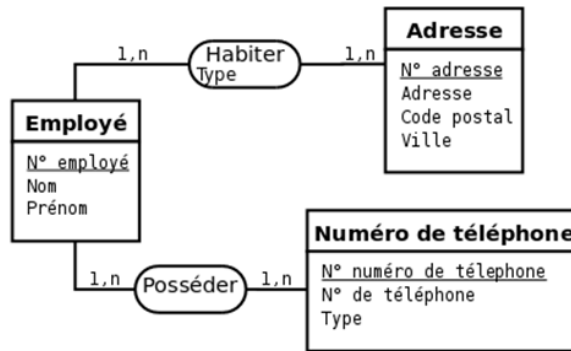


Figure 10: Exemple de modèle entité-association

2.2.3 IDEF

IDEF (Integration DEfinition Methods) est une famille de méthodes développées par l'US Air Force pour élaborer l'architecture de systèmes industriels. Elle se décline en quinze langages (de IDEF0 à IDEF14) qui couvrent un large éventail de domaines : modélisation fonctionnelle, informationnelle, d'ontologies, d'interfaces utilisateur, etc. Afin de couvrir l'ensemble des caractéristiques définies au paragraphe 1.4, nous allons présenter succinctement trois de ces langages : IDEF0, IDEF3 et IDEF1X.

2.2.3.1 IDEF0

IDEF0 (Integration Definition for Function Modeling, « version 0 ») ou SADT (Structured Analysis and Design Technique) est une méthode d'analyse fonctionnelle descendante. Le point de vue fonctionnel est la représentation structurée des fonctions, activités ou processus du système modélisé. Descendante signifie que l'analyse part d'un niveau global pour se diriger vers des niveaux plus détaillés. IDEF0 représente une fonction de transformation par une boîte entourée de flèches d'entrées et de sorties, comme on peut le voir dans la Figure 11.

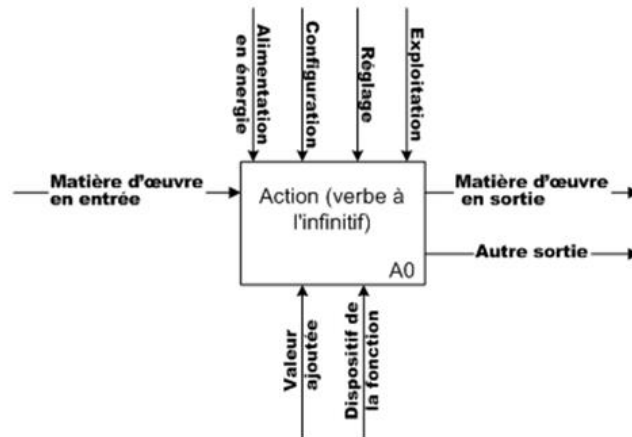


Figure 11: Représentation SADT/IDEFO (basique)

- Action : Description de la valeur ajoutée de cette étape ;
- Label : Désigne le niveau d'identification (ici A0 est le deuxième niveau le plus haut après A-0) ;
- Flèches d'entrée horizontales : Représentent les objets à traiter ;
- Flèches d'entrée verticales descendantes : Représentent les informations contraignant l'exécution de l'action (données de contrôle et de contrainte) ;
- Flèches d'entrée verticales ascendantes : Représentent les moyens nécessaires à l'exécution de l'action (ressources techniques et humaines) ;
- Flèches de sortie horizontales : Représentent l'apport de la fonction, le résultat de la transformation. Cet apport pourra ensuite être exploité pour une autre action, en suivant l'analyse descendante.

La représentation est hiérarchisée sous forme d'un enchaînement de diagrammes. Une approche IDEFO plus complète peut être représentée de la manière illustrée par la Figure 12.

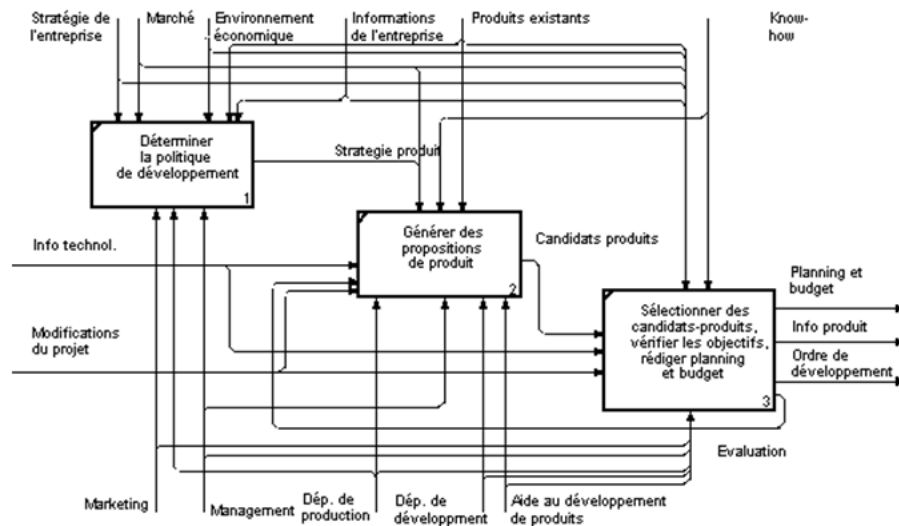


Figure 12 : Modélisation IDEF0

La principale force de IDEF0/SADT est son niveau de détail et sa simplicité. Son point faible est aussi sa simplicité avec une syntaxe assez pauvre (pas d'origine ni destination des infos par exemple). Le second problème est qu'il n'existe pas de point d'arrêt dans la modélisation, on peut descendre très bas en niveaux de détails.

Même si IDEF0 ne permet pas de réaliser des modèles selon une vue comportementale, la méthode IDEF3, qui peut l'accompagner, offre cette possibilité.

2.2.3.2 Perspective informationnelle de IDEF0

Les données dans IDEF0 sont représentées par des flèches d'entrée verticales descendantes. Mais ces flèches représentent aussi des éléments qui modulent la fonction du système : Présence d'énergie, programme, réglages, etc. Si ce mode de représentation des données est visuellement relativement clair, le fait que d'autres types d'éléments partagent cette représentation atténue cette clarté car il n'a pas été défini un véritable méta-modèle qui clarifie les concepts.

IDEF0 n'offre aucun moyen de représenter l'interopérabilité.

Le langage IDEF1X, partie prenante de la suite IDEF, sert à modéliser les données sémantiques. Nous le présentons dans la suite.

2.2.3.3 IDEF3

IDEF3 est la méthode de description des flux de processus de la suite IDEF. Elle permet de représenter ces flux ainsi que des diagrammes de changement d'état de manière graphique. Toutefois, il ne permet pas de décrire graphiquement une condition portant sur l'exécution du processus. La vue comportementale qu'il propose est par conséquent insuffisante pour servir de base à une simulation. A la différence d'IDEF0, les flèches sont des

liens de séquence comme dans BPMN et non des flux avec des noms des éléments supportés par les flux, comme on peut le voir dans la Figure 13.

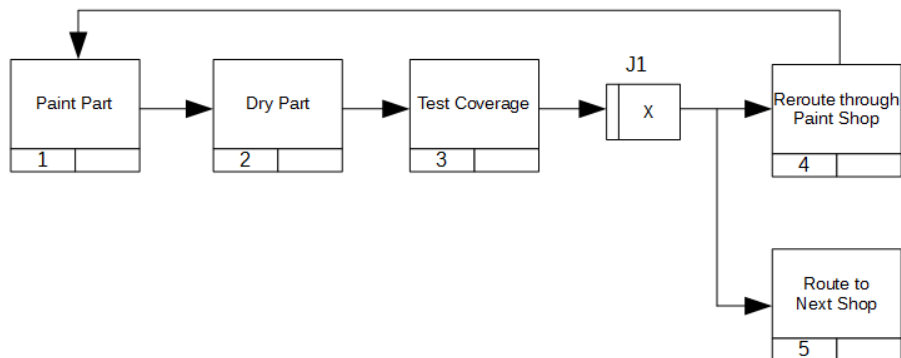


Figure 13 : Modèle de flux de processus de IDEF3

2.2.3.4 IDEF1X

IDEF1X est le langage de la suite IDEF qui a pour but la réalisation de modèles de données sémantiques. Il utilise une approche Entité-Association dont les éléments de base sont :

- les objets (personnes, lieux, évènements, etc.) représentés par des boîtes ;
- les associations entre ces boîtes, représentées par des flèches ;
- les caractéristiques ou attributs des objets, écrits à l'intérieur des boîtes.

Les boîtes dont les angles sont arrondis représentent des entités dont la clé étrangère dans son intégralité (c'est-à-dire toutes les colonnes de la table qui composent cette clé) est utilisée en tant que clé primaire. L'entité en question est alors dite « identifier-dépendant ». Sinon, elle est « identifier-indépendant », et les angles de la boîte qui la représente ne sont pas arrondis. La Figure 14 nous montre un exemple de diagramme IDEF1X.

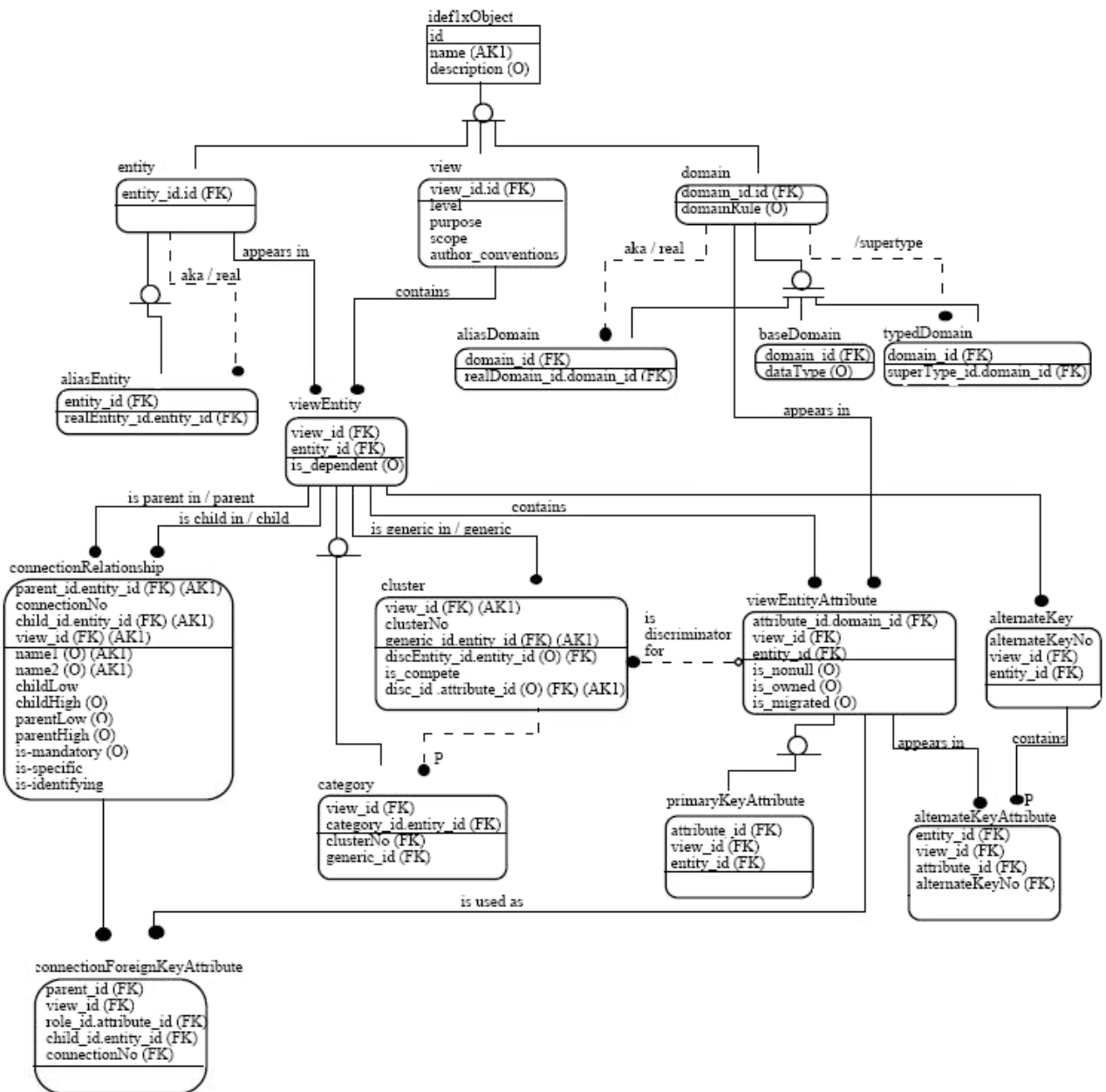


Figure 14 : Exemple de diagramme IDEF1X

Il est possible de constater sur cet exemple que IDEF1X n'est pas un langage de représentation des données de haut niveau.

De plus, il ne permet pas de représenter l'interopérabilité.

2.2.4 GRAI

La méthode GRAI (Graphe de Résultats et Activités Inter-reliés) a été développée par les Professeurs Lucas Pun et Guy Doumeingts de l'Université de Bordeaux [DOUMEINGTS 1984]. Elle est orientée pour la modélisation des systèmes de décision et est basée sur :

- Le modèle de référence (modèle GRAI), contenant les concepts de modélisation et leurs relations permettant de construire le modèle global de l'entreprise ;

- Les formalismes graphiques, représentant les concepts du modèle GRAI et facilitant la communication et l'interprétation : la grille GRAI et les réseaux GRAI ;
- Une démarche structurée et participative dans laquelle toutes les étapes et tous les acteurs de la méthode sont définis, permettant efficacité et gain de temps.

2.2.4.1 Le modèle GRAI

Au cours de cette étape, on définit la structure conceptuelle de référence du système de production, ainsi que les concepts de base et leurs interrelations.

Le premier concept important est la décision. Pour piloter un système, on doit prendre des décisions. Décider, c'est agir en fonction d'un certain nombre de connaissances : les objectifs, les contraintes, les résultats des décisions passées (indicateurs de performance), etc.

Le deuxième concept est la hiérarchisation. Il existe trois niveaux de décision : stratégique (lié aux objectifs globaux du système), tactique (lié aux décisions qui définissent les moyens pour atteindre les objectifs globaux) et opérationnel (lié aux décisions qui permettent d'agir en mettant en œuvre les moyens définis au niveau supérieur). On part des décisions qui ont un impact à long terme jusqu'à celles à court terme, qui concernent certaines formes de produits (familles pour le niveau stratégique, pièces pour l'opérationnel...) dans le but de détailler du mieux possible le fonctionnement des processus de l'entreprise et de limiter le volume des informations manipulées. Ainsi, plus l'espace de décision couvert par l'information est faible, plus le détail de l'information manipulée est grand, et vice-versa. Le but est d'être coordonné : chaque unité décisionnelle (niveau opérationnel) travaille avec les informations dont elle a besoin, en cohérence avec les objectifs du niveau stratégique.

Pour expliciter la notion de terme, on identifie l'horizon et la période des prises de décision. L'horizon est la vision que l'on se donne dans la prise de décision. La période est l'intervalle de temps après lequel la décision va être reconsidérée. La Figure 15 illustre la représentation de l'horizon et de la période.

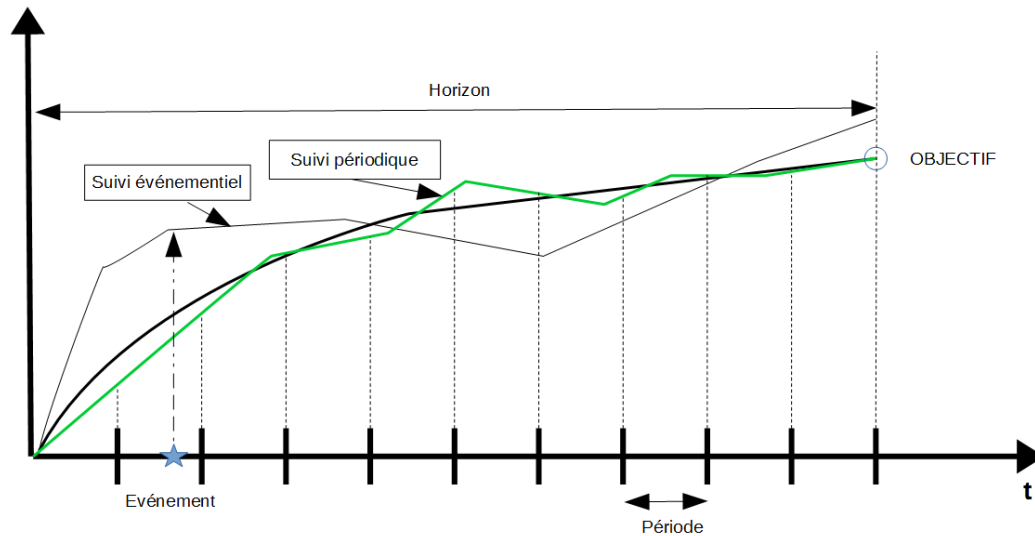


Figure 15 : Représentation de l'horizon et de la période

Toutes ces notions permettent de décomposer l'entreprise en plusieurs niveaux décisionnels et plusieurs fonctions : c'est ce que l'on appelle la grille GRAI.

2.2.4.2 La grille GRAI

La grille GRAI est réalisée à partir du modèle GRAI. Les colonnes de cette grille représentent les différentes fonctions du processus, tandis que les lignes figurent les niveaux de décision définis par des horizons et des périodes. Les cellules de la grille représentent des centres de décision, qui contiennent les décisions prises pour une fonction et pour un niveau donné. Plusieurs flux arrivent ou partent des centres de décision (cadre de décision dans le cas d'un lien hiérarchique descendant ou horizontal entre décisions (objectifs, contraintes, critères...), ou flux d'information, représentant le lien de fonctionnement entre les différents niveaux hiérarchiques et les différents services.

La Figure 16 présente un exemple de grille GRAI.

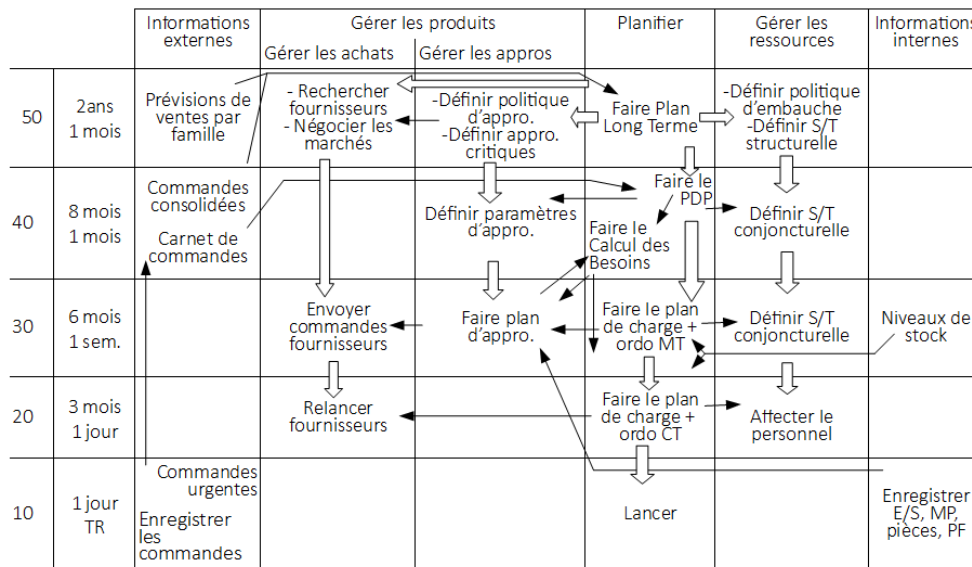


Figure 16 : Exemple de grille GRAI

2.2.4.3 Les réseaux GRAI

Les réseaux GRAI représentent le fonctionnement de tout ou une partie d'un centre de décision d'une grille GRAI. Le processus de prise de décision peut être représenté grâce à des activités de décision ou d'exécution.

Les réseaux sont composés de 3 éléments :

- A : Activité : partie du comportement du centre de décision, qui est nommée et numérotée. Il existe 2 natures d'activité :
 - Exécution (horizontal) : C'est une activité déterministe, même valeur du résultat suivant les entités convergentes (déclencheurs). Elle prépare en général l'information pour la prise de décision ou exploite le résultat de la décision. Il peut s'agir d'une activité gérée par une règle, une procédure, un programme, etc. ;
 - Décision (vertical) : C'est une activité non déterministe dans le sens où elle peut donner plusieurs résultats suivant les entités convergentes du fait du cadre de décision et non d'une règle précise. Il peut s'agir d'un choix, d'une planification...
- B : Entité : objet appartenant au système de conduite, qui sert au déroulement des activités ou est produit par celles-ci. Une entité est nommée. Il existe 7 natures d'identité : information, objectif (niveau de performance attendu), variable de décision, critère (aide au choix des actions), règle, ressource et indicateur de performance.
- C : Opérateur de renvoi : indique l'origine ou la destination d'une entité lorsque celle-ci est extérieure au centre de décision concerné. Il est représenté par le nom du service et le numéro de l'activité supérieure.

La Figure 17 présente un exemple de réseau GRAI.

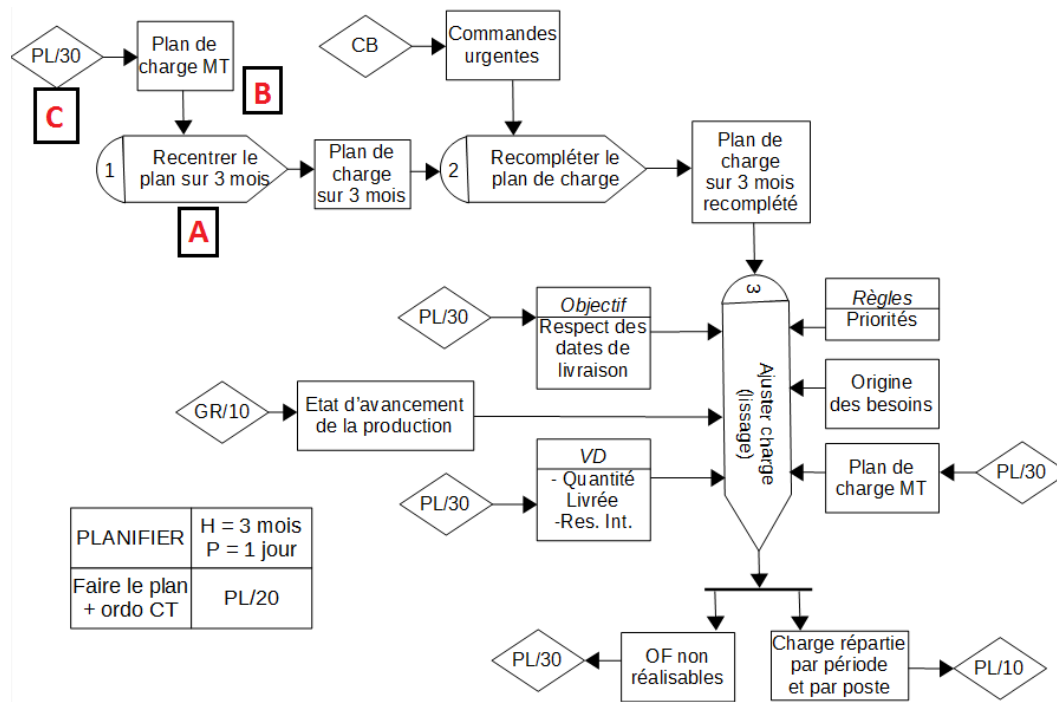


Figure 17 : Exemple de réseau GRAI

Toutefois, le réseau GRAI s'avère complexe pour représenter plusieurs activités dans le même modèle. De plus, il ne permet pas de représenter une vue comportementale.

2.2.4.4 Une démarche structurée

Le déroulement de cette démarche implique 4 groupes d'acteurs :

- Un groupe de pilotage composé de quelques décideurs qui valident les résultats intermédiaires (diagnostic et propositions pour le futur fonctionnement) ;
- Un groupe de synthèse regroupant des responsables des différents secteurs de l'entreprise qui ont pour mission d'élaborer la grille GRAI ;
- Un groupe de spécialistes de la démarche.
- Un ensemble de personnes interviewées qui vont fournir de la connaissance profonde pour élaborer les réseaux GRAI et consolider la grille GRAI.

La démarche se déroule selon un plan précis composé des phases suivantes :

- phase d'initialisation (définition des objectifs de l'étude, des groupes, du macro planning, rencontres et formation GRAI avec les membres des groupes...);
- phase de modélisation et d'analyse de l'existant, au cours de laquelle on établit les grilles et réseaux relatifs au fonctionnement existant en se basant sur le travail des

groupes et des interviews. Cette phase se termine par la mise en évidence des points à améliorer et un rapport d'analyse ;

- phase de précision du futur système en fonction des contraintes externes et internes ;
- phase de conception du futur système décisionnel.

L'intérêt des réseaux GRAI est de représenter les informations utiles pour les prises de décision qui seront collectées dans le système d'information.

2.2.4.5 Perspective informationnelle de GRAI

Les données sont présentes dans la grille GRAI dans les colonnes d'informations internes et externes et dans les réseaux GRAI en tant qu'entités.

Au niveau des réseaux GRAI, la représentation des données est similaire à celle d'IDEFO. Les données, ainsi que les éléments de contrôles, y sont en effet représentés par des flèches verticales descendantes.

Cependant, rien n'est prévu pour représenter l'interopérabilité, ni au niveau de la grille GRAI, ni à celui des réseaux GRAI.

2.2.5 CIMOSA

CIMOSA (*Computer Integrated Manufacturing Open System Architecture*) a pour but la construction de systèmes intégrés de production. Le cadre est fondé sur la notion de cycle de vie d'un système et offre un langage de modélisation, une méthodologie et une technologie de soutien pour appuyer ces objectifs.

2.2.5.1 L'architecture CIMOSA

Cette architecture comprend [VERNADAT 1999] :

- un cadre de modélisation (*Modeling FrameWork*, MFW) : il comporte une architecture de référence qui permet de construire un modèle générique et une architecture particulière (spécifique à un cas donné) ;
- une plate-forme d'intégration (*Integrating InfraStructure*, IIS) du système physique et des applications logicielles à partir des modèles précédents ;
- une méthodologie d'intervention qui couvre le cycle de vie du système de production et permet d'assurer la cohérence des différents modèles.

Le cadre de modélisation, appelé « cube CIMOSA » s'articule autour de trois principes fondamentaux et complémentaires :

- l'axe de généralité, qui se compose de trois niveaux : le niveau générique (définition des primitives de base du langage de modélisation, les « constructs »), le niveau partiel (structures prédéfinies et réutilisables pour un domaine d'application), et le

niveau particulier (modèles spécifiques de l'entreprise). Les niveaux générique et partiel constituent l'architecture de référence de CIMOSA (sujette à normalisation) ;

- l'axe de dérivation ou axe des modèles identifie le cycle de développement. Il définit trois niveaux de modélisation : le niveau de définition des besoins (écriture du cahier des charges), le niveau des spécifications de conception (analyse conceptuelle des solutions aux besoins exprimés), et le niveau de description de l'implantation (description précise de la solution retenue) ;
- l'axe de génération, ou axe des vues, qui définit l'entreprise selon quatre vues : les fonctions (fonctionnalités et comportement de l'entreprise en termes de processus, d'activités et d'opérations), les informations (description du système d'information), les ressources (moyens nécessaires à mettre en œuvre pour réaliser les fonctions de l'entreprise), et l'organisation (description de la distribution des responsabilités, de l'autorité dans la prise de décisions).

Le cube CIMOSA est représenté dans la Figure 18.

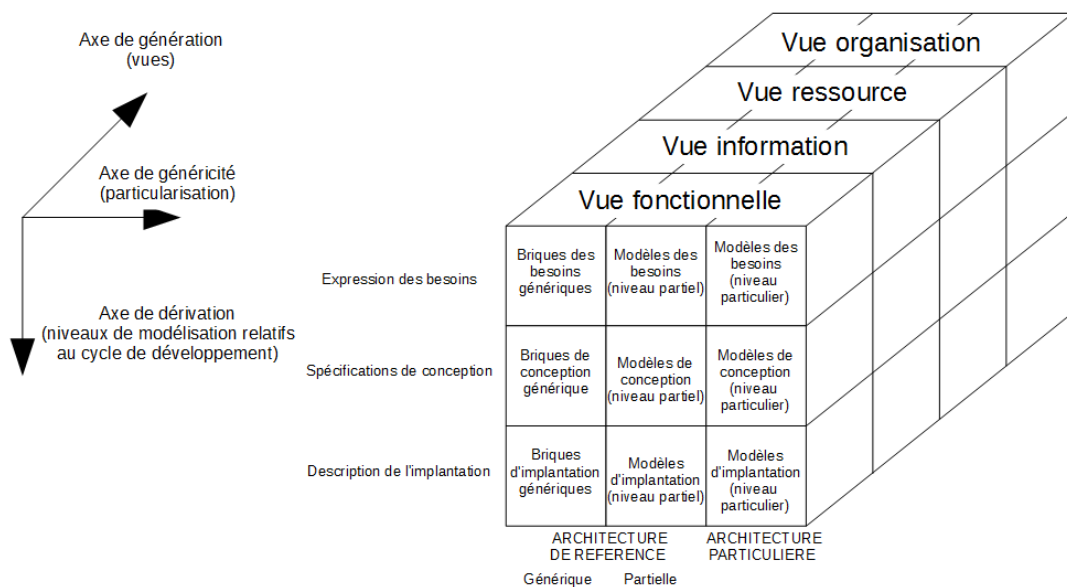


Figure 18 : Cube CIMOSA

A l'inverse des autres méthodes de modélisation en entreprise, CIMOSA ne préconise aucun formalisme graphique de représentation des processus. La justification d'un tel fait est que « *la modélisation graphique des processus (est) par nature ambiguë* » [VERNADAT 1998]. Le processus est modélisé selon un langage formel (*CIMOSA modelling language*), vérifiable et exploitable par ordinateur. Ce manque volontaire de représentation graphique dans la méthode est peut-être à l'origine d'une vision un peu nébuleuse de CIMOSA, puisqu'on ne "voit" pas ce qu'on modélise. C'est surtout la raison pour laquelle cette méthode de modélisation en entreprise n'a pas pénétré le milieu industriel de la même manière que des méthodes comme IDEF0.

2.2.5.2 La démarche de modélisation

La démarche de modélisation en entreprise avec CIMOSA passe par les étapes suivantes [VERNADAT 1998] :

- l'analyse des domaines fonctionnels de l'entreprise et leurs relations ;
- l'identification des processus maîtres à modéliser ;
- l'analyse détaillée des processus maîtres selon les principaux concepts (constructs) définis dans la méthode (à savoir les événements, sous-processus, activités, aptitudes et vues d'objets) ;
- la consolidation du modèle au niveau spécification de conception ;
- l'adaptation du modèle aux contraintes d'implantation ;
- la traduction du modèle dans le langage des systèmes utilisés.

Cette démarche s'inscrit dans une méthodologie plus complète, qui prend en compte l'ensemble du cycle de vie du système.

2.2.5.3 Perspective informationnelle de CIMOSA

La vue informationnelle de CIMOSA envisage les entrées et sorties des activités par des objets ayant certains états. Ces entrées et sorties sont représentés par deux éléments de modélisation :

- L'objet d'entreprise : entité de l'entreprise ayant son propre cycle de vie et décrit par un ensemble de propriétés pouvant aussi être des objets ;
- La vue d'objet : manifestation de l'état d'un ou plusieurs objets d'entreprise à un instant donné. Deux types de vues d'objets fondamentaux peuvent être distingués : une vue physique de l'objet et une vue informationnelle de l'objet. [BENAMA 2016].

Cependant, l'absence de formalisme graphique de représentation ne permet pas une visualisation claire des échanges de données pour un public non technicien.

CIMOSA n'offre aucun moyen de représenter l'interopérabilité.

2.2.6 SYNTHÈSE DES LANGAGES DE MODÉLISATION

Nous venons de passer en revue différents langages de modélisation. Le but de cette revue de la littérature était de sélectionner les outils qui nous permettraient de réaliser les modèles dont nous avons besoin pour représenter et résoudre les problèmes d'interopérabilité. Les fonctionnalités que nous demandons à ce langage sont :

- Être un langage de haut niveau afin que tous les publics puissent l'appréhender sans difficulté.

- Permettre de représenter les points d'interopérabilité, et donc les échanges de données.
- Permettre de réaliser des modèles selon une vue comportementale.

Il est maintenant temps pour nous de réaliser une synthèse de cet état de l'art concernant les langages de modélisation.

Pour nous aider dans cette tâche, nous avons réalisé un tableau de synthèse qui présente les fonctionnalités pour chaque langage étudié.

A noter que pour la colonne « Représentation des données » nous avons choisi de n'autoriser que trois valeurs : « Évidente », « Moyenne » et « Peu évidente ». Ces valeurs sont à replacer dans le contexte d'une visualisation des modèles par un public non spécialiste. Le Tableau 1 présente les résultats.

	Haut niveau	Vue comportementale	Représentation des données	Représentation de l'interopérabilité
BPMN 2.0	Oui	Oui	Évidente	Non
ARIS	Oui	Oui	Peu évidente	Non
IDEF	Oui	Non	Moyenne	Non
GRAI	Oui	Non	Moyenne	Non
CIMOSA	Oui	Non	Peu évidente	Non

Tableau 1 : Synthèse des langages de modélisation

On peut tout d'abord noter que tous ces langages sont de haut niveau, mais qu'aucun d'entre eux ne permet de représenter l'interopérabilité.

Penchons-nous donc sur la fonctionnalité de représentation des données. Nous avons vu que BPMN 2.0 offrait de représenter de manière très intuitive les données échangées par le biais d'icônes spécifiques. Du côté d'IDEF0 et du modèle GRAI, un public non averti peut visualiser les données impliquées dans le processus, mais nous avons noté que les mêmes représentations étaient utilisées pour d'autres types d'éléments (contraintes). Ce comportement nous semble propice à la confusion. C'est pourquoi nous leur avons attribué la valeur « Moyenne ». Dans le cas d'ARIS, les données sont représentées de manière séparée par rapport aux processus, et la représentation sur un modèle entité-association n'est pas adaptée à un public non technicien. Quant à CIMOSA, la représentation par des objets et des vues d'objets ne nous paraît pas non plus adaptée à notre cas.

En ce qui concerne la possibilité de réaliser des modèles selon une vue comportementale, seuls BPMN 2.0 et ARIS satisfont cette condition.

Un autre argument joue en faveur de BPMN 2.0 : sa très grande popularité au sein des entreprises. De ce fait, les résultats de notre travail pourront être utilisés sans peine par le plus grand nombre.

Enfin, il dispose d'un méta modèle indiscutable et de nombreux outils informatiques supports.

Au vu des résultats ci-dessus, notre préférence se porte sur le langage BPMN 2.0.

Après avoir passé en revue différents langages de modélisation il nous faut nous pencher sur les méthodes de mesure de la performance.

2.3 METHODES DE MESURE DE LA PERFORMANCE

Comme dans le cas des langages de modélisation, nous allons faire une revue de la littérature afin de déterminer la méthode de mesure de la performance qui répond le mieux à notre besoin. Notre besoin est, rappelons-le, de réaliser des modèles qui permettent à un public hétérogène d'évaluer de manière évidente les conséquences des problèmes d'interopérabilité.

La fonctionnalité que nous recherchons donc est la possibilité d'afficher les mesures de la performance des activités impliquées par les échanges de données concernées par l'interopérabilité au niveau du coût, du délai, de la qualité et de la disponibilité de manière intuitive.

Nous allons donc passer en revue des méthodes de mesure de la performance qui comptent parmi les plus emblématiques.

2.3.1 TABLEAU DE BORD

Le tableau de bord a été conçu en France au début du XX^e siècle.

« Le tableau de bord est un ensemble d'indicateurs peu nombreux (5 à 10) conçus pour permettre aux gestionnaires de prendre connaissance de l'état de l'évolution des systèmes qu'ils pilotent et d'identifier les tendances qui les influenceront sur un horizon cohérent avec la nature de leurs fonctions. » [BOUQUIN 2008].

Un tableau de bord est donc un outil permettant d'afficher les mesures de la performance. Il a été développé dans le but d'améliorer les processus de production et de mieux appréhender les effets des actions mises en place et leurs impacts sur le niveau de performance des processus. Le décideur peut ainsi apprécier la justesse (ou non) des actions mises en place et de les corriger le cas échéant. Les décisions pourront ainsi être prises en connaissance de cause.

Le nombre d'indicateurs affichés doit être restreint afin de ne pas noyer l'information. Bien entendu, les indicateurs affichés doivent être Simples, Mesurables, Atteignables, Réalistes et Temporellement définis (SMART) [CPC 1997].

Le tableau de bord doit comprendre des informations concernant [DAUM 2005] :

- Le statut des éléments qui, en tant que facteurs majeurs de succès (étapes critiques), jouent un rôle majeur dans l'atteinte des objectifs définis (variables d'action) ;
- les mesures et initiatives qui ont été mises en place dans le but d'atteindre les objectifs ou étapes définis, en influençant le statut des facteurs de succès (plans d'action) ;
- les performances actuelles des systèmes respectifs en ce qui concerne l'atteinte des objectifs (résultats).

On peut dégager cinq rôles d'un tableau de bord :

- Réduire l'incertitude : Le tableau de bord sert à structurer l'information et à faciliter le passage au sein des différents acteurs d'une entreprise ou d'un processus ; de ce fait il aide à se focaliser sur un contexte et des décisions bien précises ;
- Stabiliser l'information : L'outil concentre une certaine quantité d'informations, qui est potentiellement variable avec le temps ; le tableau de bord fige ce temps, permettant de se concentrer sur un moment précis de l'activité ;
- Faciliter la communication : Le tableau de bord est un référentiel commun : en effet, chaque acteur de l'organisme concerné se base sur un modèle identique pour chacun, c'est-à-dire que tout le monde a accès aux mêmes données représentées par un même tableau de bord ;
- Dynamiser la réflexion : L'un des avantages d'un bon tableau de bord est le fait de pouvoir représenter des données de manière claire et structurée ; si cet affichage est suffisamment précis et pertinent par rapport à la valeur mesurée (exemple : une courbe pour représenter une évolution...), alors il sera beaucoup plus simple de s'y consacrer et d'en réaliser une réflexion ;
- Maîtriser le risque : Chaque prise de décision entraîne des conséquences, et donc des prises de risques. Grâce au tableau de bord, le contexte de la décision est plus facile à prendre en compte, et sert donc à mieux identifier les risques potentiels de chaque décision.

La conception d'un tableau de bord comprend les étapes suivantes :

- analyse et définition de la stratégie de l'entreprise globale et de chacune de ses unités,
- détermination des indicateurs et des actions à mener pour atteindre les objectifs stratégiques.

2.3.2 BALANCED SCORECARD

Le Balanced Scorecard (BSC, ou tableau de bord prospectif) est une méthode développée à partir de 1982 par les économistes Robert S. KAPLAN et David NORTON pour mettre fin aux tableaux de bord uniquement financiers. Ce tableau de bord équilibré, prospectif et stratégique a pour principal objectif de mesurer les activités de l'entreprise pour faciliter la conception de la stratégie et d'en assurer le déploiement à l'échelle opérationnelle. Il permet de :

- Traduire la stratégie de l'entreprise et en faciliter la communication auprès de chaque employé ;
- Identifier les processus clés du succès de la mise en œuvre stratégique ;
- Aligner les investissements humains, technologiques et organisationnels afin qu'ils œuvrent dans le sens de la stratégie ;
- Mettre en lumière les écarts de mise en œuvre de la stratégie et ainsi en faciliter la correction.

L'origine de la méthode repose sur le principe que la performance financière n'est plus le seul élément à prendre en compte lors de la conception des tableaux de bord. En effet, ce type de performance n'est mesurable qu'à long terme, et leur consultation donne une vision « retardée » de la performance de l'entreprise. L'utilisation de ce seul critère entraîne donc un manque de réactivité qui peut être préjudiciable à la productivité de l'entreprise.

D'autre part, la particularité du BSC est incarnée en grande partie par l'adjectif « Balanced », que l'on peut traduire par « équilibré ». Cette notion d'équilibre concerne les objectifs à court/moyen/long terme, ou les indicateurs financiers/non-financiers.

Les indicateurs sont répartis selon quatre axes [KAPLAN 1996] :

- Financier : Permet d'évaluer les conséquences économiques des actions mises en œuvre. Indicateurs possibles : Rentabilité, chiffre d'affaire, bénéfices, etc.
- Clients : Satisfaction clientèle, rentabilité par segment, fidélité, etc.
- Processus internes : Quels sont les processus clés pour l'entreprise ? Indicateurs possibles : délai de production, qualité, délai de livraison, etc.
- Apprentissage organisationnel et développement : Quels sont les moyens (humains, organisationnels, ...) à mettre en œuvre pour progresser ?

La Figure 19 nous présente les quatre axes et illustre le lien de causalité qui existe entre eux.

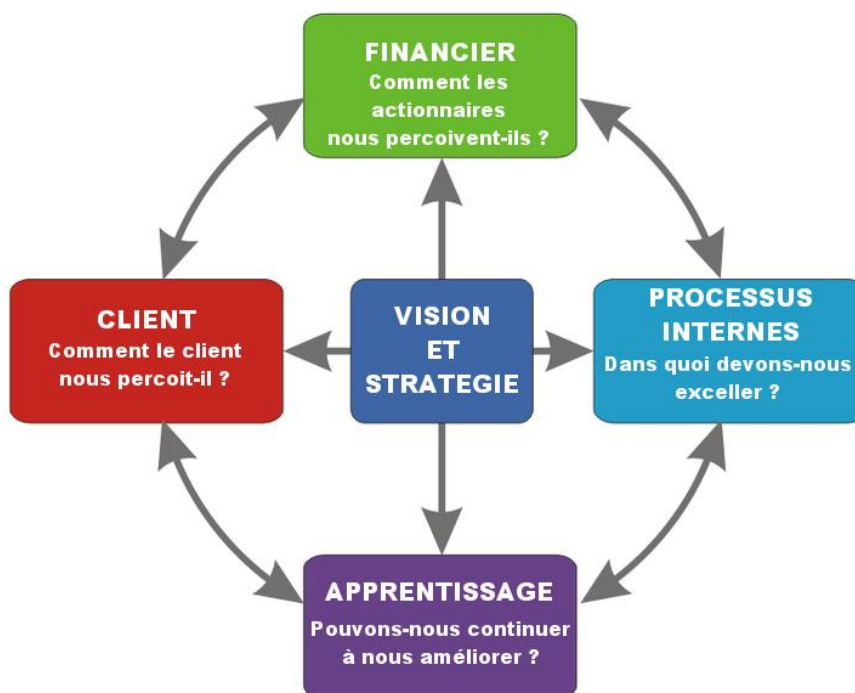


Figure 19 : Les quatre axes du Balanced Scorecard

En fait, ces axes sont inter-liés dans le sens où une amélioration de l'apprentissage organisation va améliorer les processus internes qui vont améliorer le nombre de clients qui vont améliorer les résultats financiers.

Des tableaux de bord spécifiques sont répartis à chaque niveau de l'entreprise pour que l'ensemble du personnel puisse apprécier les performances correspondant à sa tâche.

Pour élaborer les tableaux de bord, on utilise des Strategy Maps, que l'on peut traduire par le terme carte de stratégie. Cet outil représente les différents déterminants de performance de chaque axe et les liens de causalité existant entre eux.

2.3.3 OVAR

La méthode OVAR (Objectifs, Variables d'Action et Responsables) a été conçue par trois professeurs du groupe HEC (Daniel Michel, Michel Fiol et Hugues Jordan). Il s'agit d'une démarche structurée de conception d'un système de tableaux de bord dans le but de déployer la stratégie au sein des différents niveaux hiérarchiques de l'entreprise. OVAR a pour objectif de créer de la cohérence entre les équipes, donner du sens aux objectifs, impliquer l'ensemble des collaborateurs et responsabiliser les équipes, et enfin piloter l'ensemble pour répondre aux objectifs initiaux.

Le principe général est de définir au plus haut niveau (niveau N) des objectifs globaux et des variables d'action (OVA) pour l'entreprise. Les variables d'action pour le niveau N deviennent des objectifs pour le niveau N – 1. Des responsabilités (R) sont alors données au

niveau N-1 qui, à son tour, développe ses propres OVAR, et ce jusqu'au plus bas niveau hiérarchique [BOURGUIGNON 2001]. On définit ensuite des indicateurs qu'on présente dans des tableaux de bord et on constitue des équipes de pilotage qui redéfinissent des objectifs. C'est une démarche qui ressemble dans son déroulement au PDCA, comme l'illustre la Figure 20.

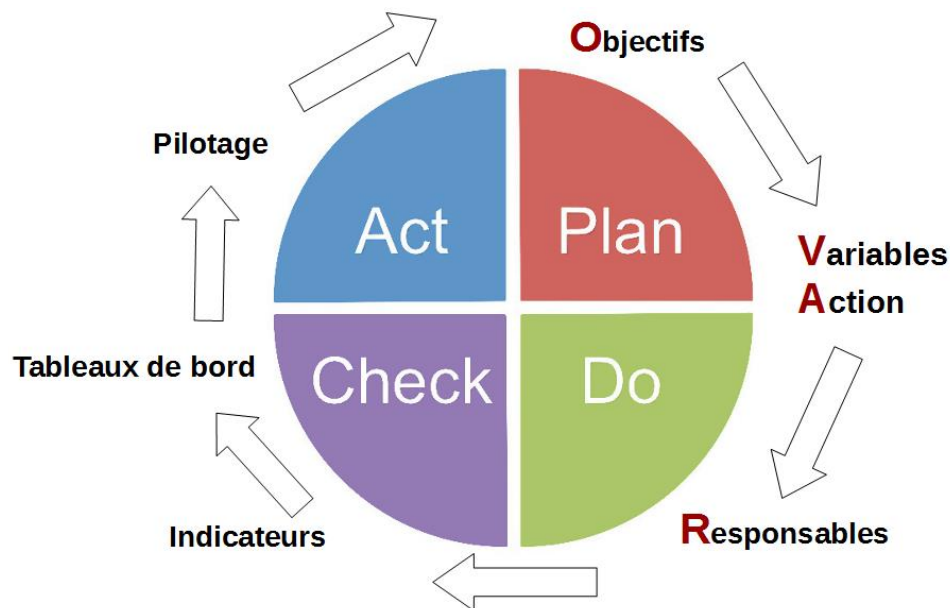


Figure 20 : Déroulement de la méthode OVAR

Le déploiement se fait en 5 étapes :

- Phase d'intelligence et de conception. Les dirigeants (niveau N) fixent les objectifs globaux de l'entreprise puis définissent les OVA ;
- Attribution des responsabilités. Chaque individu doit élaborer ses objectifs ; pour cela, chacun constitue une grille, appelée grille de cohérence, regroupant les objectifs ainsi que les variables d'action jugées nécessaires pour répondre à ces objectifs. Ainsi, chaque variable d'action doit être gérée par un responsable de l'entreprise ;
- Le partage des perceptions et la mise en place des objectifs cohérents. Une équipe comprend entre autres un dirigeant et plusieurs responsables. Le dirigeant, après avoir établi sa grille (grille mère), va attribuer une variable d'action à un responsable d'équipe qui développera lui-même sa propre grille (grille fille). Ainsi, chaque variable d'action attribuée par le dirigeant devient un objectif à atteindre, qui sera à son tour décliné en variable d'action et attribué à un responsable ;
- La définition des indicateurs. Suite à la définition des objectifs et des variables d'action de chacun, plusieurs types d'indicateurs vont se dégager : des indicateurs de résultats (objectifs et variables d'action), d'impacts (objectifs et variables d'action) et d'efforts (variables d'action) ;
- La mise en place des tableaux de bord. Une fois les indicateurs déterminés, une étude est menée pour déterminer si les objectifs initiaux et si l'avancée des plans d'action

sont cohérents. Si l'ensemble est cohérent alors le projet est réussi, mais dans le cas contraire, l'étape Act du PDCA entre en jeu : analyser et commenter les données, mettre en place des actions correctives...

Cette méthode a pour principal intérêt l'implication du personnel à tous les niveaux dans la définition des objectifs et des plans d'action. Ceci entraîne une grande cohérence dans ces objectifs et plans d'action, ainsi qu'une plus grande implication des équipes dans la mise en place des plans d'action.

2.3.4 GIMSI

Élaborée par le consultant expert Alain Fernandez [FERNANDEZ 1999], la méthode GIMSI (Généralisation de l'accès aux Informations décisionnelles en s'appuyant sur une Méthodologie d'inspiration Systémique facilitant l'expression des Individualités de l'entreprise) est utilisée dans le cadre de la conduite d'un projet de pilotage de la performance centré sur les acteurs, car ce sont eux qui, en fin de compte, prennent les décisions. Le créateur de la méthode part du principe que si ce type de démarche est motivé par un certain nombre d'indicateurs, c'est bel et bien l'homme qui est responsable de la tenue du projet ; ainsi, s'il y a un désintérêt, si les décisions à prendre ne sont pas comprises ou acceptées, ou encore si les décisions ne coïncident pas avec le besoin, alors le projet ne pourra pas être mené à bien. Pour lui, il est malheureusement fréquent de constater qu'aujourd'hui, certains projets sont menés dans l'unique but d'optimiser les processus (et donc les dépenses), sans se soucier de l'importance des hommes dans les démarches d'amélioration continue. Dans cette méthode, l'implication et la mise en commun des idées, de l'opinion de tous les acteurs du projet, des managers jusqu'aux opérateurs, est capital.

La mise en place d'un projet GIMSI se déroule en quatre phases :

- Identification (Quel est le contexte ?) : En se basant sur l'environnement et la structure de l'entreprise pour notamment installer le périmètre du projet ;
- Conception (Que faut-il faire ?) : Ici se concentrent les actions liées à la définition des objectifs, le choix du tableau de bord et des indicateurs, puis la collecte et l'analyse des données ;
- Mise en œuvre (Comment le faire ?) : Cette phase passe par le choix d'un outil adapté à la Business Intelligence (par exemple un progiciel) en se basant sur notre besoin et nos données, puis le déploiement de la solution ;
- Suivi permanent (Le système correspond-t-il toujours aux attentes ?) : Grâce à un système d'audit, une vérification de ce qu'apporte la solution par rapport aux besoins initiaux sera réalisée.

Ces quatre phases sont divisées en étapes (dix au total), chacune d'elles marquant un seuil identifiable dans l'avancement du projet. Le Tableau 2 nous présente ces étapes :

Phase	N°	Étape	Objectif
Identification	1	Environnement de l'entreprise	Analyse de l'environnement économique et de la stratégie de l'entreprise afin de définir le périmètre et la portée du projet
	2	Identification de l'entreprise	Analyse des structures de l'entreprise pour identifier les processus, activités et acteurs concernés
Conception	3	Définition des objectifs	Sélection des objectifs tactiques de chaque équipe
	4	Construction du tableau de bord	Définition du tableau de bord de chaque équipe
	5	Choix des indicateurs	Choix des indicateurs en fonction des objectifs choisis
	6	Collecte des informations	Identification des informations nécessaires à la construction des indicateurs
	7	Le système de tableau de bord	Construction du système de tableaux de bord, contrôle de la cohérence globale
Mise en œuvre	8	Le choix des progiciels	Élaboration de la grille de sélection pour le choix des progiciels adéquats
	9	Intégration et déploiement	Implantation des progiciels, déploiement à l'entreprise
Amélioration permanente	10	Audit	Suivi permanent du système

Tableau 2 : Les étapes de la méthode GIMSI

Le projet est basé sur une communication permanente entre les concepteurs et les utilisateurs. Il est mené par un comité de pilotage placé sous la tutelle d'un architecte des systèmes et qui coordonne plusieurs groupes de projet.

2.3.5 PRISM

La méthode Prism (ou prisme de la performance) est issue des travaux d'Andy Neely et Chris Adams [NEELY 2002], Professeurs à Cranfield University. Cette méthode est décrite par ses créateurs comme étant la seconde génération de système de management de la performance, au travers de 3 raisons :

- Il n'est plus possible pour une organisation de tenir seulement compte des besoins d'une ou deux parties prenantes. Une majorité des cadres de mesure de la performance se concentre uniquement sur les besoins des propriétaires et des clients de l'organisation. Les autres parties prenantes comme les employés ou les fournisseurs ont tendance à être oubliés ;
- La majorité des cadres de mesure de la performance ignorent les changements qui doivent être réalisés pour la stratégie, les processus et la capacité de l'entreprise permettant de répondre aux besoins des parties prenantes ;
- Les parties prenantes doivent contribuer d'une certaine manière à l'organisation. Il y a un quiproquo entre l'entreprise et les parties prenantes, notamment sur le fait que ces derniers attendent quelque chose de l'organisation, mais celle-ci attend également quelque chose en retour des parties prenantes. La mesure de la performance doit donc prendre en considération qu'une partie prenante apporte bien ce qui est souhaité par l'entreprise.

Neely et Adams pensent que beaucoup d'organisations sont obsédées par les mesures. Les managers mesurent trop de choses, dans le but initial de pouvoir maîtriser l'ensemble de l'organisation, cependant cette hausse des mesures tendait à voir apparaître des micro-gestions, concentrées sur un point bien précis. Cela entraînait pour les managers une perte de temps, en plus d'une perte de vue des objectifs stratégiques de l'entreprise.

Le principe du prisme de la performance est le suivant : la performance d'une organisation dépend de cinq domaines de performance représentés par les cinq facettes d'un prisme interagissant entre elles comme nous le montre la Figure 21.

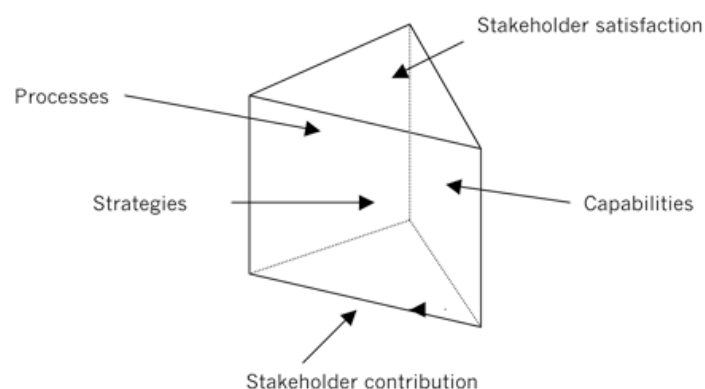


Figure 21 : Prisme de la performance

- Satisfaction des parties prenantes : Qui sont les parties prenantes, que veulent-ils et quels sont leurs besoins ? En général, les parties prenantes les plus importantes sont

les investisseurs (retour sur investissement, récompenses sous forme de dividendes ou d'intérêts, retour sur la santé de l'entreprise, etc.), les clients (souhaitent des produits ou services de bonne qualité, le moins cher possible, délivré au bon moment, etc.), les employés (recherchent un travail intéressant, veulent une rémunération décente), les fournisseurs (recherchent une relation leur permettant d'étendre leur business, d'obtenir des retours sur leur performance) et les régulateurs (surveillent le respect de la législation de l'entreprise, son bilan et son compte de résultats).

- Stratégies : Quelles stratégies faut-il mettre en œuvre pour satisfaire ces demandes et ces besoins ? Une fois les stratégies appropriées identifiées, l'objectif est d'avoir des mesures permettant de montrer la bonne implémentation des stratégies, de communiquer ces stratégies au sein de l'organisation, d'encourager l'implémentation des stratégies par les managers et enfin de vérifier si les stratégies sont toujours respectées.
- Processus : De quels processus avons-nous besoin pour exécuter ces stratégies ? Beaucoup d'entreprises classifient 4 types de processus : Développement des produits et des services, demande générée, demande accomplie et planification et gestion de l'entreprise.
- Capacités : De quelles capacités avons-nous besoin pour exploiter nos processus de la manière la plus efficace et la plus efficiente possible ? Par exemple, dans le cas du traitement d'une commande client dans l'industrie, il est nécessaire de maîtriser la gestion de la demande du client, la gestion de la planification, l'approvisionnement, la fabrication, la distribution et la facturation.
- Contribution des parties prenantes : Que voulons-nous et en quoi avons-nous besoin des parties prenantes pour développer et maintenir ces capacités ? Par exemple, les investisseurs (capital pour se développer, pour prendre plus de risques), les employés (flexibilité, talents multiples), les régulateurs (meilleure compréhension du secteur d'activité, bonnes relations de travail, baisse de la bureaucratie), etc.

Les facettes du prisme sont liées et doivent se compléter. Les stratégies exigées sont identifiées, puis les processus nécessaires le sont, suivis des capacités. Le développement le plus important dans le prisme de la performance est d'identifier les besoins d'une large gamme de parties prenantes, de même qu'identifier ce que l'organisation souhaite obtenir en retour de ces dernières. Ainsi, les mesures de performance ne doivent pas découler de la seule stratégie.

Le prisme est représenté dans la Figure 22.

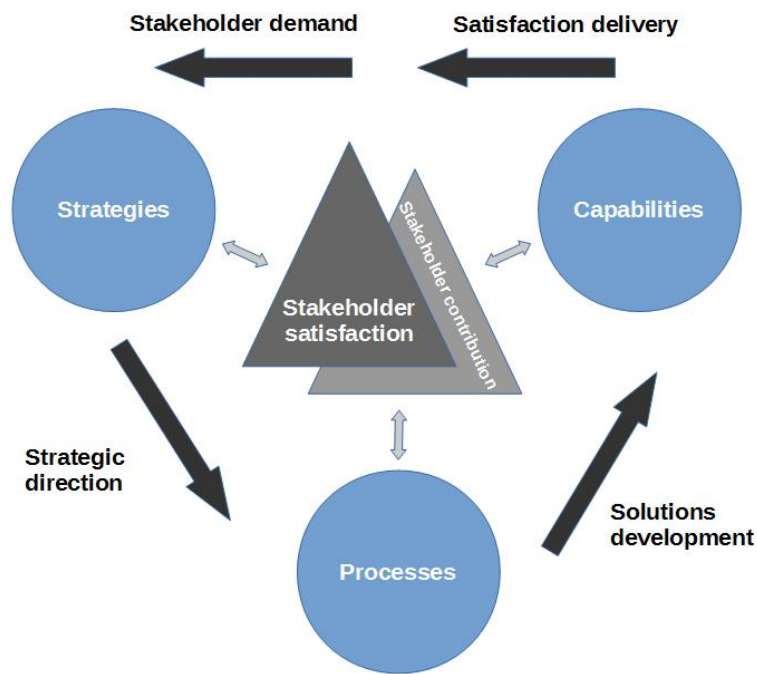


Figure 22 : Les facettes du prisme

La grande originalité de cette méthode réside dans la prise en compte de la satisfaction des parties prenantes.

2.3.6 ABC/ABM/ABB

La méthode ABC (Activity Based Costing), ou méthode des coûts par activité, est un outil axé sur la compréhension de la formation des coûts de revient, leurs causes et leurs variations. Elle a été créée au milieu des années 1980 par Cooper et Kaplan [COOPER 1988] suite à une réflexion menée par le Consortium for Advanced Manufacturing International (CAM-I), pour mesurer les seuls critères économiques des activités. C'est un outil décisionnel dont le but est de décloisonner l'entreprise, grâce à une gestion par activité. Alors que la comptabilité traditionnelle considère que les produits consomment des ressources, la méthode ABC se base sur le concept suivant : « Les produits consomment des activités, les activités consomment des ressources » [THYSSEN 2006]. C'est le principe qui considère que les activités consomment toutes les charges et que les produits ou les services consomment toutes les activités [RAVELOMANANTSOA 2009].

Le principe de la méthode ABC est visible dans la Figure 23.

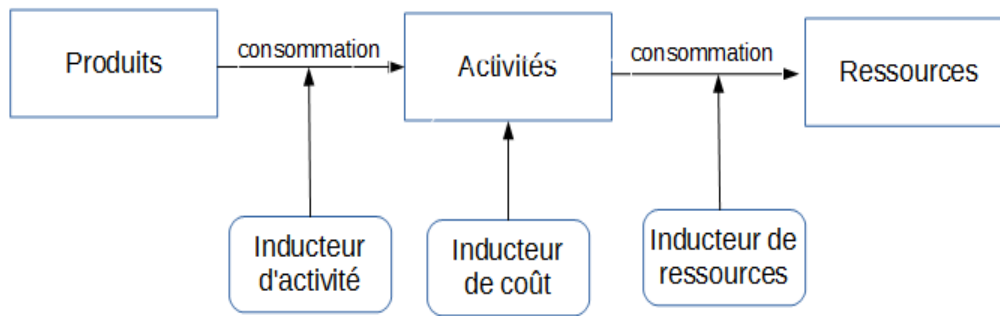


Figure 23 : Principe de la méthode ABC

Le ABC permet de calculer des coûts de revient plus représentatifs du monde industriel. Grâce à cette méthode, une entreprise peut connaître l'intégralité des coûts de sa chaîne de valeur, en identifiant au préalable ses processus (achats, fabrication...), et qualifiant ses activités, permettant ainsi de supprimer ou réduire des gaspillages ou de la non-valeur ajoutée (temps de réglage des machines, stock, etc. ABC est par ailleurs extrêmement efficace pour déterminer la rentabilité des processus, en mesurant les activités et leurs fonctions, en déterminant les coûts de revient, en prenant en compte les fonctions support (gestion de la qualité...) et en identifiant les leviers d'amélioration possibles.

La méthode est appliquée en suivant les cinq étapes suivantes [RAVELOMANANTSOA 2009] :

- établissement d'une cartographie des processus,
- affectation des charges et des temps de travail aux différentes activités,
- choix des inducteurs et des indicateurs de performance pour chaque activité,
- identification des quantités de ressources consommées par activité pour un produit ou un service,
- détermination du coût de revient du produit, du coût total, du coût unitaire détaillé par activité.

ABC est complétée par les méthodes ABM (Activity Based Management, ou gestion par activité), pour remettre en cause l'organisation d'une entreprise et les pratiques des acteurs, et ABB (Activity Based Budget), pour définir le budget et se projeter dans l'avenir. Dans les faits, ABM utilise les coûts des activités et des ressources calculés par ABC pour prendre les décisions permettant de réduire ces coûts et ainsi atteindre les objectifs de l'entreprise.

2.3.7 SYNTHÈSE DES MÉTHODES DE MESURE DE LA PERFORMANCE

A travers cette revue de la littérature sur le sujet des mesures de la performance, nous cherchions à déterminer quelle méthode nous servirait dans la suite de nos travaux. Notre besoin est une représentation intuitive des mesures de la performance des activités

impliquées par les échanges de données concernées par l'interopérabilité au niveau du coût, du délai, de la qualité et de la disponibilité.

Les différentes méthodes de mesure de la performance que nous avons passées en revue ont en commun le fait qu'elles ont pour but de piloter la performance de la totalité de l'entreprise dans le but d'atteindre ses objectifs globaux (même si elles proposent toutes la décomposition des indicateurs et de la stratégie à chacun des niveaux de l'entreprise). De ce fait, leurs méthodologies de mise en place exigent une analyse détaillée de toutes les parties de l'entreprise.

Dans le cas qui nous occupe, il ne s'agit pas de piloter la totalité de l'entreprise, mais de résoudre un problème localisé. De plus, le choix des indicateurs a déjà été défini. La mise en place d'une de ces méthodes est loin d'être élémentaire et représenterait donc une autre approche plus globale.

Nous avons également consulté les travaux de Michel Ravelomanantsoa [RAVELOMANANTSOA 2009] dans lesquels 35 méthodes de mesure de la performance sont passées en revue. Cependant aucune d'entre elles ne répond aux besoins spécifiques énoncés précédemment.

En conséquence, nous avons choisi de n'utiliser aucune de ces méthodes de mesure de la performance.

2.4 CONCLUSION

Au cours de cette revue de la littérature nous avons tout d'abord présenté un certain nombre de langages de modélisation afin de déterminer lequel répondrait à notre besoin. Nous avons tout d'abord pu nous rendre compte qu'aucun d'entre eux ne permettait de représenter nativement l'interopérabilité. Les autres critères qui étaient le fait d'être un langage de haut niveau, de permettre de représenter des vues comportementales et d'afficher de manière claire les données échangées ont fait pencher notre choix sur BPMN 2.0.

Nous avons ensuite passé en revue diverses méthodes de mesure de la performance et avons constaté qu'aucune d'entre elles n'apportait une réponse satisfaisante à notre problème spécifique. Ce sont en effet toutes des méthodes globales dont le but est de représenter les performances de l'entreprise dans sa totalité. Leur mise en place représente donc un effort disproportionné par rapport à la fonction qu'elles ont à remplir.

Au vu de ce constat, nous en avons déduit qu'il était nécessaire d'enrichir un langage de modélisation existant afin d'être en mesure de réaliser les modèles appropriés à la résolution des problèmes d'interopérabilité. Notre choix final s'est porté sur BPMN 2.0 du fait qu'il s'agit d'un langage de haut niveau (et donc compréhensible par tout type de public) qui offre la possibilité de représenter de manière évidente les échanges de données, qu'il offre la possibilité de réaliser des modèles selon une vue comportementale et qu'il dispose d'un mécanisme d'extension. Ce mécanisme nous permettra de réaliser des extensions qui

représenteront l'interopérabilité dans les modèles, mais aussi d'y afficher les mesures de la performance des tâches concernées par ces échanges de données.

Dans le chapitre suivant, nous allons tout d'abord présenter le mécanisme d'extension de BPMN 2.0. Puis, nous détaillerons les extensions que nous proposons en adéquation avec les conclusions du chapitre 1.

Ensuite, nous introduirons les méthodes d'agrégation de la performance qui nous permettront de comparer les performances d'une tâche à celles d'un ensemble de tâche.

Nous allons aussi définir une méthode de gestion de la qualité des données qui nous permettra de nous pencher plus spécifiquement sur ce problème.

Chapitre 3 : CONTRIBUTIONS CONCEPTUELLES ET METHODOLOGIQUES PAR AJOUT DE CONCEPTS A BPMN

<u>3.1 Introduction</u>	62
<u>3.2 Démarche globale</u>	62
<u>3.3 Extension de BPMN 2.0</u>	65
<u>3.3.1 Mécanisme d'extension de BPMN</u>	66
<u>3.3.2 Propositions d'extension</u>	67
<u>3.4 Méthodes d'agrégation de la performance</u>	73
<u>3.4.1 Réduction des processus</u>	74
<u>3.4.2 Agrégation des mesures de performance</u>	75
<u>3.5 Gestion de la qualité des données</u>	78
<u>3.5.1 AMDEC</u>	78
<u>3.5.2 Adaptation de l'AMDEC à la qualité des données</u>	81
<u>3.6 Synthèse générale</u>	90
<u>3.7 Conclusion</u>	93

3.1 INTRODUCTION

Au cours du chapitre 1, nous avons mis en évidence le problème que représentait l'absence d'interopérabilité dans les échanges de données entre différents systèmes d'information. Puis, nous avons vu que l'établissement de modèles était une étape incontournable sur le chemin de la résolution de ce problème, tout en constatant que les langages de modélisation souffraient de lacunes dans cette finalité : l'impossibilité de représenter l'interopérabilité et la performance. Nous en avons déduit que l'enrichissement des langages de modélisation était indispensable dans l'optique de cette résolution.

Dans le chapitre 2, nous avons fait un état de l'art de différents langages de modélisation ainsi que de méthodes de mesure de la performance. Nous en avons tiré la conséquence d'enrichir le langage BPMN 2.0 pour lui apporter des fonctionnalités d'affichage de l'interopérabilité et de mesures de la performance.

Dans ce chapitre, nous allons présenter le mécanisme d'extension de BPMN 2.0 qui permet de créer de nouveaux éléments de ce langage. Puis nous ferons une proposition d'extension du modèle BPMN 2.0 pour représenter les échanges de données présentant un problème d'interopérabilité et ceux pour lesquels ce problème a été résolu, et enfin pour afficher dans les modèles les mesures de la performance des tâches concernées par ces échanges de données.

Nous allons aussi présenter les méthodes de réduction des processus et d'agrégation de la performance qui nous permettront de comparer dans les modèles AS IS et TO BE les performances d'une tâche à celles d'un ensemble de tâche regroupées au sein d'un sous-processus.

Nous allons enfin définir une méthode de gestion de la qualité des données. Cette méthode, basée sur l'AMDEC nous permettra d'une part de définir les priorités dans le traitement des problèmes de qualité, et d'autre part de mettre en place une démarche d'amélioration continue sur ce sujet.

Ces différents outils seront utilisés dans une démarche globale dont le but est d'améliorer la qualité des données qui sont échangées dans un contexte d'interopérabilité.

3.2 DEMARCHE GLOBALE

Les différentes étapes de cette démarche sont exposées dans la Figure 24.

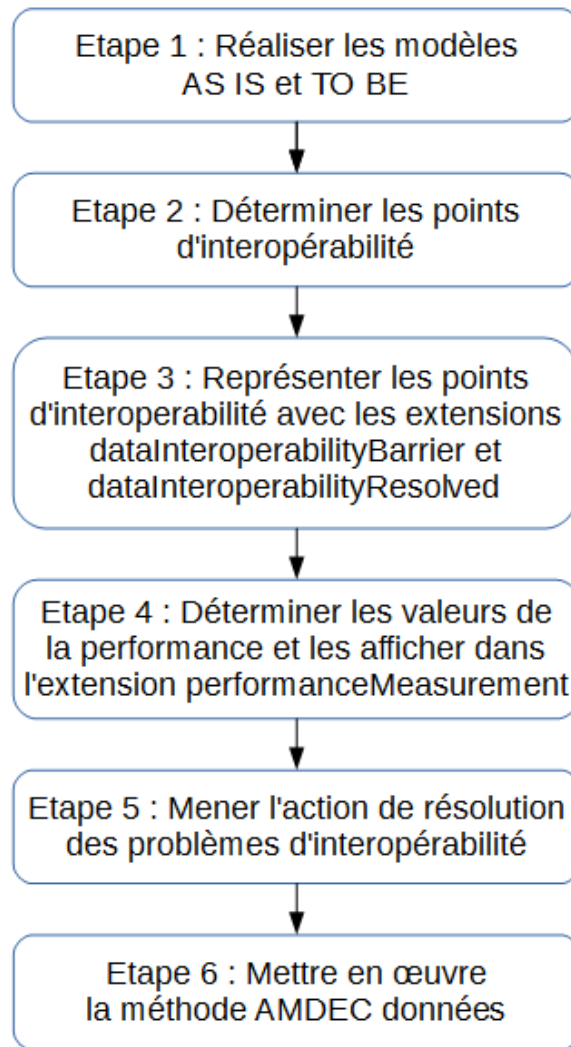


Figure 24 : Etapes de la démarche globale

- Etape 1 : on commence par modéliser le processus existant, puis le processus souhaité. Ces modélisations doivent être faites dans l'optique de la résolution des problèmes d'interopérabilité, il n'est donc pas nécessaire de détailler la totalité des tâches, uniquement celles qui sont impliquées dans les échanges de données.
- Etape 2 : parmi les différents échanges de données présents dans le processus, on détermine ceux qui présentent un problème d'interopérabilité. Il s'agira de ceux qui interviennent entre différents systèmes d'informations.
- Etape 3 : on représente les échanges repérés à l'étape précédente à l'aide des extensions *dataInteroperabilityBarrier* (dans le modèle AS IS) et *dataInteroperabilityResolved* (dans le modèle TO BE).
- Etape 4 : on détermine les valeurs de la performance des tâches impliquées par ces échanges de données et on les place dans l'extension

performanceMeasurement. Les calculs d'agrégation de la performance sont faits automatiquement.

- Etape 5 : on établit l'interopérabilité des données. Cette étape technique est hors du cadre de cette thèse.
- Etape 6 : on met en œuvre la méthode AMDEC données. Cette étape vise deux objectifs :
 - S'assurer que les actions d'établissement de l'interopérabilité sont suivies d'effets.
 - Mettre en place une démarche d'amélioration continue de la qualité des données.

Afin d'illustrer cette méthode, nous allons utiliser un exemple de processus dans lequel un problème d'interopérabilité existe. Le cas que nous allons évoquer est réaliste car il s'inspire d'un cas réel, mais il est imaginaire, car l'établissement de l'interopérabilité n'y a pas encore été entrepris. Il s'agit du processus d'approvisionnement en fromages du dépôt des expéditions de la fromagerie Onetik. Le processus d'expédition des commandes d'Onetik nous servira de cas d'étude au chapitre 5. L'exemple d'illustration que nous utilisons dans le présent chapitre se situe en amont du processus étudié dans le cas d'étude.

Le fonctionnement du processus d'approvisionnement est illustré par la Figure 25.

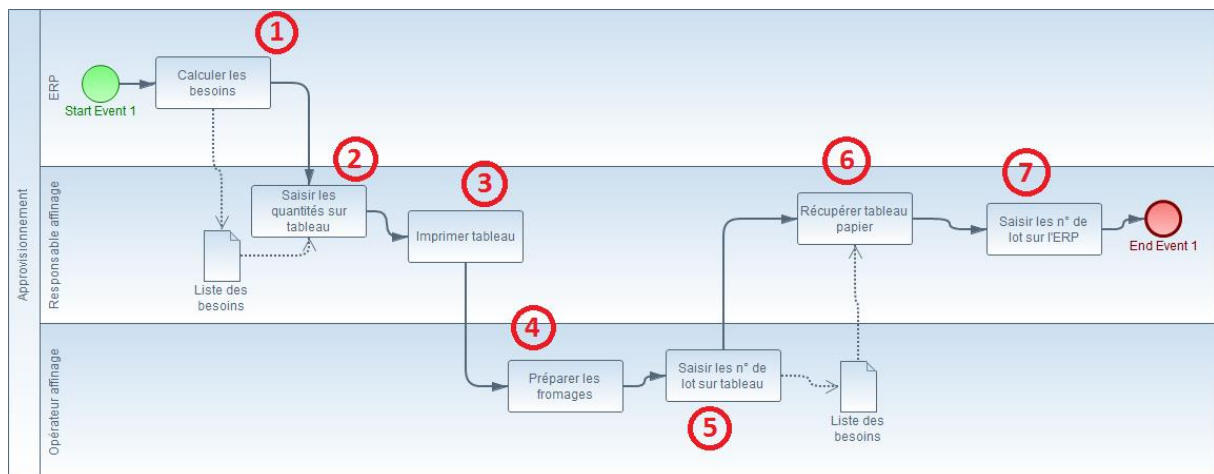


Figure 25 : Modèle AS IS

1. Les besoins en fromages sont calculés par l'ERP.
2. Le responsable affinage ressaisit les quantités des fromages sur un tableau puis l'imprime.
3. Le responsable affinage imprime le tableau rempli.
4. L'opérateur d'affinage prépare et transporte les fromages au dépôt expéditions.
5. Il note les lots utilisés dans le tableau papier.

6. Le responsable affinage va chercher le tableau papier.
7. Le responsable affinage saisit les numéros de lot sur l'ERP.

La solution envisagée est de munir les opérateurs d'affinages de tablettes connectées à l'ERP. Elle est illustrée par la Figure 26.

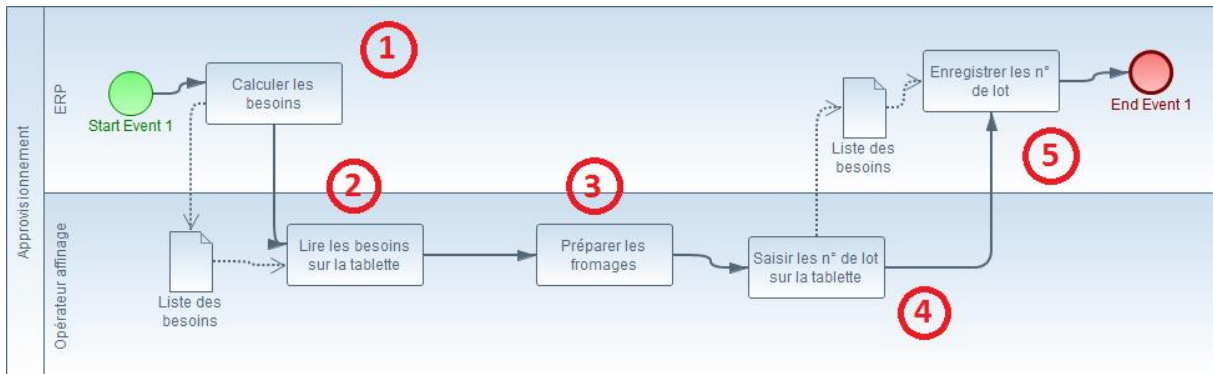


Figure 26 : Modèle TO BE

1. Les besoins en fromages sont calculés par l'ERP.
2. L'opérateur d'affinage lit les besoins sur la tablette.
3. L'opérateur d'affinage prépare et transporte les fromages au dépôt expéditions.
4. L'opérateur d'affinage saisit les numéros de lot sur la tablette.
5. Les numéros de lot sont automatiquement enregistrés dans l'ERP.

Les modèles AS IS et TO BE qui nous ont servi à présenter l'exemple illustratif sont ceux qui sont réalisés dans l'étape 1 de la démarche globale.

Dans l'étape 2, on détermine quels sont les points d'interopérabilité. Il s'agit ici des deux échanges de la liste des besoins présents dans chacun des modèles.

Dans l'étape 3, on doit représenter ces points d'interopérabilité. Nous devons pour cela étendre le méta-modèle de BPMN 2.0.

3.3 EXTENSION DE BPMN 2.0

Notre but est de créer trois extensions durables de BPMN 2.0. Les deux premières permettront de représenter l'interopérabilité dans les échanges de données entre différents systèmes d'information. La première permettra de visualiser de manière explicite et intuitive les échanges de données présentant un problème d'interopérabilité, tandis que la deuxième représentera ceux pour lesquels ce problème aura été résolu. La troisième servira à afficher la performance au niveau du coût, du délai, de la qualité et de la disponibilité pour une tâche ou un ensemble de tâches.

3.3.1 MECANISME D'EXTENSION DE BPMN

La structuration technique de BPMN est basée sur le concept de couches extensibles autour d'un noyau d'éléments simples appelé BPMN Core (voir Figure 27). Chaque couche est construite en ajoutant de nouveaux éléments qui étendent ceux de la couche inférieure la plus proche.

Cette extensibilité permet de définir une surcouche d'éléments afin de mieux représenter des concepts inhérents à un secteur d'activité spécifique. Chaque nouvelle couche est construite en extension d'une couche plus basse. BPMN fournit des éléments génériques d'extension dans le méta-modèle. Les extensions BPMN servent le plus souvent à représenter les besoins spécifiques d'un secteur d'activité [OMG 2003]. Mais ils peuvent aussi servir à combler une lacune de ce langage à un niveau plus généraliste. C'est ce que nous nous proposons de faire en étendant les modèles BPMN en y ajoutant deux extensions représentant les données concernées par l'interopérabilité et une troisième permettant d'afficher la performance.

La Figure 27 représente la structuration en couches de BPMN.

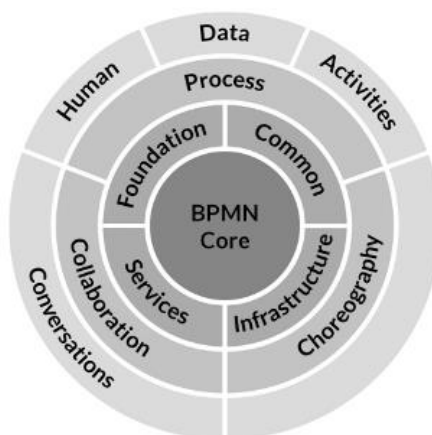


Figure 27 : Structuration en couches de BPMN

Le noyau central (BPMN Core) contient les éléments les plus fondamentaux.

La couche 1 est composée des « packages » suivants :

- Infrastructure : contient la classe « *Definitions* ».
- Foundation : classes partagées entre les autres packages du noyau (*Base Element, Documentation, Extensibility, etc.*).
- Service : les constructs fondamentaux nécessaires pour les services de modélisation et les interfaces.

Common : Elements utilisables par tous les types de diagramme *Flow elements, Events, Gateways*, etc.

La couche 2 contient les extensions de chacun des types de diagrammes : *Process, Choreography* et *Collaboration*.

Enfin, la couche 3 définit les extensions relatives aux objets suivants : *human, data, activities* et *conversations*.

Le mécanisme d'extension BPMN consiste en un ensemble d'éléments d'extension qui permettent l'ajout d'éléments et attributs additionnels aux éléments BPMN standards et existants. Ces éléments d'extension sont : *ExtensionDefinition, ExtensionAttributeDefinition, ExtensionAttributeValue* et *Extension*.

L'élément *Extension* est utilisé pour relier au modèle BPMN une extension dont la structure est définie en utilisant l'élément *ExtensionDefinition*. Celui-ci regroupe des attributs additionnels utilisés pour étendre le modèle BPMN en les attachant à n'importe quel élément BPMN. La définition de chaque attribut inclut le nom et le type de l'attribut ; et elle est donnée par l'élément *ExtensionAttributeValue* correspondant. Dans un élément étendu BPMN, *ExtensionAttributeValue* est utilisé pour assigner une valeur à un attribut d'extension particulier qui a été précédemment défini dans *ExtensionDefinition* en utilisant l'élément *ExtensionAttributeDefinition*.

La structure de BPMN est décrite à l'aide de deux représentations : un méta-modèle Meta-Object Facility (MOF) qui décrit les concepts et un XML Schema Definition (XSD) qui établit le format d'échanges [OMG 2003].

3.3.2 PROPOSITIONS D'EXTENSION

3.3.2.1 Représentation de l'interopérabilité

Afin de combler la lacune de BPMN dans la visibilité des échanges de données présentant un problème d'interopérabilité, nous introduisons deux extensions : *dataInteroperabilityBarrier* et *dataInteroperabilityResolved*. Ces deux extensions étendent l'objet *DataObject*.

La première (*dataInteroperabilityBarrier*) représente un échange de données présentant un problème d'interopérabilité non résolu. Elle permet de représenter cet écueil de manière à ce que tous les intervenants du projet (managers, techniciens informatique, propriétaires du processus, etc.) puissent se rendre compte qu'il existe un problème à résoudre. Son but n'est pas de rentrer dans des considérations techniques. Idéalement, elle ne devrait être utilisée que dans la modélisation de l'existant (AS IS), puisque la réalisation du projet devrait comprendre la résolution de la difficulté. Mais on peut supposer que dans certains cas, pour diverses raisons (manque de financement ou de temps, choix de priorités par exemple) aucune solution ne sera mise en place pour résoudre certains problèmes

d'interopérabilité. Cette extension pourrait donc aussi éventuellement faire partie de la modélisation du futur système (TO BE).

La seconde extension (*dataInteroperabilityResolved*) servira à représenter un problème d'interopérabilité surmonté. A ce titre, elle ne devra avoir sa place que dans la modélisation du projet du futur système (TO BE). L'intérêt de cette représentation est de permettre de s'assurer par la suite que le problème d'interopérabilité ne se reproduira pas.

3.3.2.1.1 Extension

dataInteroperabilityBarrier et *dataInteroperabilityResolved* étendent *dataObject*. Ces deux classes héritent donc de ses attributs et modèles d'association.

Aucun attribut additionnel n'est défini, car ces extensions ont un rôle uniquement représentatif. Elles servent uniquement à faire apparaître visuellement le problème ou la résolution de l'interopérabilité.

Nous avons donc défini les éléments *ExtensionDefinition* suivants : *dataInteroperabilityBarrier* et *dataInteroperabilityResolved*. Nous n'avons pas eu besoin de définir d'*ExtensionAttributeDefinition* ni d'*ExtensionAttributeValue* du fait qu'il ne s'agit que de représentations purement visuelles, qui ne nécessitent pas d'attributs supplémentaires.

Le Tableau 3 nous présente les éléments *ExtensionDefinition*.

Nom de l'extension	<i>dataInteroperabilityBarrier</i>	<i>dataInteroperabilityResolved</i>
<i>ExtensionDefinition</i>	<i>dataInteroperabilityBarrier</i>	<i>dataInteroperabilityResolved</i>
<i>ExtensionAttributeDefinition</i>		
<i>ExtensionAttributeValue</i>		

Tableau 3 : Eléments d'extension de *dataInteroperabilityBarrier* et *dataInteroperabilityResolved*

Le diagramme de classes MOF du méta-modèle de BPMN est visible dans le manuel de spécification. Il est divisé en différentes sections. Nous ne représenterons ici que celle qui concerne les objets de données. Les classes correspondant aux extensions que nous proposons sont représentées en gris dans la Figure 28. Ce sont les classes *dataInteroperabilityBarrier* et *dataInteroperabilityResolved*.

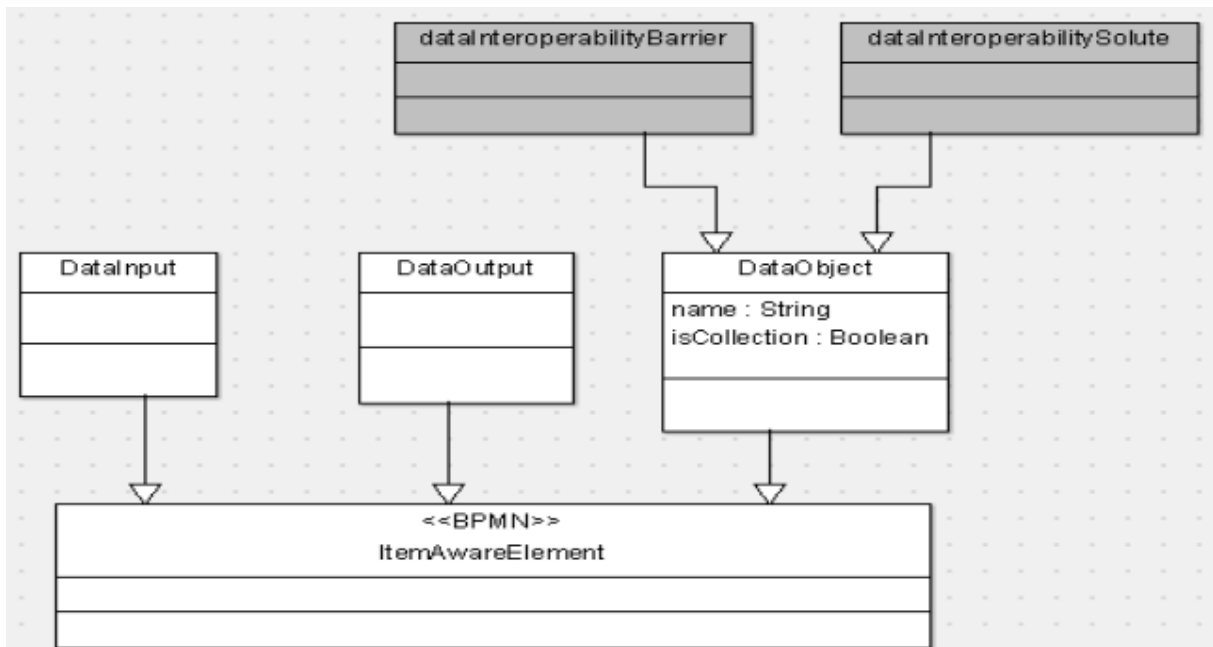


Figure 28 : Méta-modèle MOF des extensions *dataInteroperabilityBarrier* et *dataInteroperabilityResolved*

Il eut été intéressant d'associer une contrainte d'association à ces deux extensions : afin qu'elles ne puissent être reliées qu'à des tâches appartenant à des couloirs différents. En effet, un problème d'interopérabilité ne peut exister qu'entre différents systèmes, représentés par des couloirs distincts. Mais dans la structuration de base de BPMN, toute tâche source peut être reliée à toute tâche cible. Nous avons donc respecté cette flexibilité. A charge donc aux modeleurs de tenir compte de cette flexibilité et de n'utiliser ces extensions qu'à bon escient, c'est-à-dire entre deux tâches appartenant à des couloirs différents.

3.3.2.1.2 Représentation graphique :

Nous proposons la représentation graphique visible dans la Figure 29 pour l'extension *dataInteroperabilityBarrier* :



Figure 29 : *dataInteroperabilityBarrier*

Nous avons repris la représentation graphique du Data Object à laquelle nous avons ajouté deux flèches inverses, représentant l'interopérabilité, barrées, ce qui signifie que l'interopérabilité n'est pas assurée. Cette représentation, comme celle de

dataInteroperabilityResolved ci-dessous, est placée dans le modèle BPMN avec les mêmes règles et contraintes que celle des Data Object (avec la contrainte supplémentaire que sa source et sa cible doivent être présentes dans des couloirs différents).



Figure 30 : *dataInteroperabilityResolved*

Dans le cas de l'extension *dataInteroperabilityResolved*, la représentation est similaire, à la différence près que les flèches représentant l'interopérabilité ne sont pas barrées, signe que l'interopérabilité est assurée. Cette représentation graphique est présentée par la Figure 30.

Reprenons notre exemple illustratif, car à ce stade on peut réaliser l'étape 3, c'est-à-dire représenter les points d'interopérabilité sur les modèles à l'aide de ces deux extensions.

Le modèle AS IS est illustré par la Figure 31.

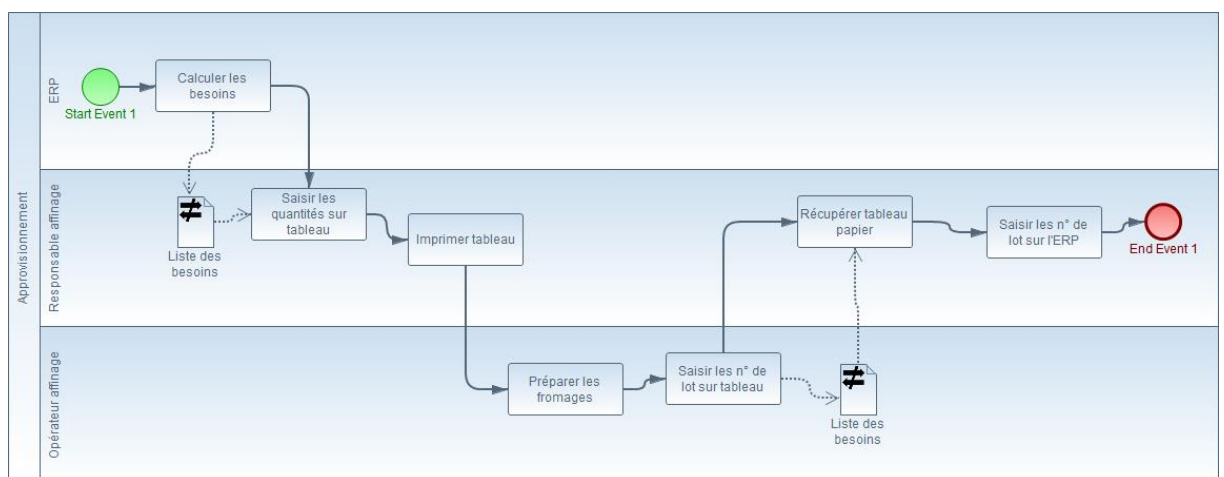


Figure 31 : Modèle AS IS avec les points d'interopérabilité

Ce modèle est identique au modèle AS IS que nous avons réalisé lors de l'étape 1, à la différence que les échanges de données sont représentés par l'extension *dataInteroperabilityBarrier* du fait qu'ils présentent un problème d'interopérabilité.

Le modèle AS IS est illustré par la Figure 32.

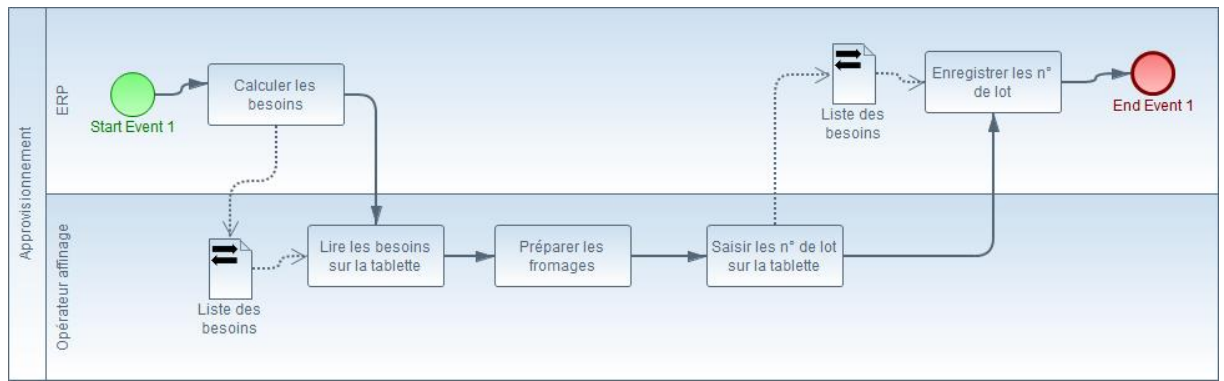


Figure 32 : Modèle TO BE avec les points d'interopérabilité

Ce modèle est identique au modèle TO BE que nous avons réalisé lors de l'étape 1, à la différence que les échanges de données sont représentés par l'extension *dataInteroperabilityResolved* car l'interopérabilité a été établie.

Afin de réaliser l'étape 4 de la méthode globale, nous devons étendre le méta-modèle de BPMN 2.0 pour y représenter la performance.

3.3.2.2 Représentation de la performance

Pour permettre la représentation de la performance dans les modèles BPMN, nous proposons l'extension *performanceMeasurement*.

Il sera possible à l'aide de cette extension de visualiser les valeurs des mesures de performance dans les domaines suivants : coût, délai, qualité et disponibilité.

Cet objet doit pouvoir être relié à une tâche ou à un ensemble de tâches. En effet, le but est de pouvoir comparer les valeurs des performances du modèle AS IS et du modèle TO BE. Or, les performances qui nous intéressent dans ces deux modèles sont celles des tâches directement impliquées par les échanges de données présentant un problème d'interopérabilité. Nous avons vu plus haut que lorsqu'il existait un problème d'interopérabilité dans un échange de données, une (ou plusieurs) tâche à non-valeur ajoutée était nécessaire afin de rendre les données utilisables pour la suite du processus. Donc, dans le modèle de l'existant, les tâches impliquées par l'échange des données seront la tâche de réception des données et la tâche de mise en conformité des données (tâche à non-valeur ajoutée). Le modèle TO BE représentant le système cible (idéal), l'échange de données représenté ne présente plus de problème d'interopérabilité. Aucune tâche à non-valeur ajoutée n'est donc nécessaire pour adapter les données. La seule tâche impliquée dans l'échange de données est donc la tâche de réception. Ainsi, dans le premier cas, notre extension doit être reliée à un sous-processus alors que dans le deuxième elle ne concerne qu'une seule tâche.

Les tâches dans BPMN sont représentées par l'objet *Task*. Les sous-processus sont visuellement représentés par l'objet *SubProcess*.

3.3.2.2.1 Extension

Cette extension a pour but de représenter les valeurs de la performance dans différents domaines. En conséquence, contrairement aux extensions précédentes, nous allons définir des attributs correspondant aux domaines concernés.

Nous avons défini les éléments *ExtensionDefinition* suivants : *Cost*, *Quality*, *Availability* et *Time* pour représenter les dimensions définies plus haut. Les *ExtensionAttributeDefinition* correspondants sont *costValue* pour *Cost*, *qualityValue* pour *Quality*, *availabilityValue* pour *Availability* et *timeValue* pour *Time*.

Le Tableau 4 nous présente les éléments *ExtensionDefinition*.

Nom de l'extension	performanceMeasurement			
ExtensionDefinition	performanceMeasurement			
ExtensionAttributeDefinition	Cost	Quality	Availability	Time
ExtensionAttributeValue	costValue	qualityValue	availabilityValue	timeValue

Tableau 4 : Eléments d'extension de performanceMeasurement

Nous pouvons observer le diagramme MOF de cette extension dans la Figure 33.

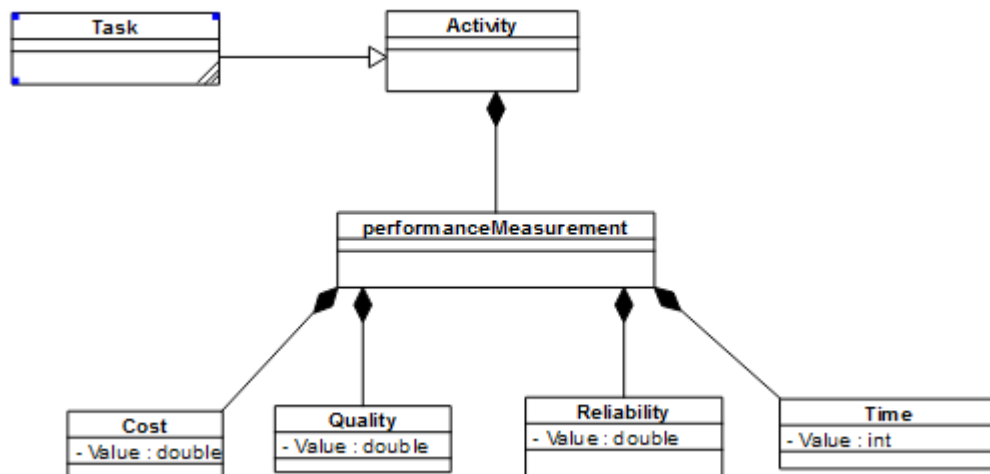


Figure 33 : Méta-modèle MOF de l'extension performanceMeasurement

3.3.2.2.2 Représentation graphique

Nous proposons la représentation graphique présentée par la Figure 34 pour l'extension *performanceMeasurement*.

C	10€
T	135s
Q	90%
A	95%

Figure 34 : performanceMeasurement

L'unité du coût est l'euro (mais elle peut être modifiée selon la région d'utilisation) et celui du temps est la seconde.

La qualité est calculée en divisant le nombre d'occurrences de données de qualité par le nombre total de données échangées. La disponibilité est calculée en divisant le nombre d'occurrences de données présentes par le nombre total de données échangées

Dans la partie suivante, nous allons présenter les méthodes d'agrégation de la performance que nous avons utilisées dans nos travaux

3.4 METHODES D'AGREGATION DE LA PERFORMANCE

Nous avons vu qu'un problème d'interopérabilité induisait la réalisation d'une tâche à non-valeur ajoutée dans le but de permettre d'exploiter les données échangées dans la suite du processus. Notre but étant la résolution des problèmes d'interopérabilité, nous devons être en mesure de comparer les performances des tâches impliquées dans l'échange de données du système existant (AS IS) avec celles du système futur (TO BE). Or, une fois le problème d'interopérabilité résolu, la tâche à non-valeur ajoutée aura été supprimée et n'apparaîtra donc pas dans le modèle TO BE. Nous allons donc comparer les mesures de la performance d'une tâche (dans le TO BE) à celles d'un ensemble de tâches (dans le AS IS). Dans le cas du système existant, il nous faut donc pouvoir agréger les mesures de la performance de plusieurs tâches.

Nous proposons donc dans cette section un modèle d'agrégation des mesures de performance.

Nous allons pour cela nous appuyer sur les travaux d'Yves Ducq [DUCQ 1999] et de Jorge Cardoso [CARDOSO 2002] pour déterminer les différentes typologies d'agrégation dans le cadre de l'interopérabilité.

3.4.1 REDUCTION DES PROCESSUS

La réduction des processus consiste à appliquer un ensemble de règles de réduction à un ensemble de tâches jusqu'à ce qu'il ne reste qu'une tâche atomique. Les mesures de la performance de cette tâche correspondent à celles de l'ensemble des tâches qui ont été réduites. [CARDOSO 2002] a développé l'algorithme *Stochastic Workflow Reduction* (SWR) qui utilise six règles de réduction : séquentielle, parallèle, conditionnelle, tolérante aux fautes, boucle, et réseau. Toutefois, dans le cadre de notre recherche, et sur la base des travaux d'Yves Ducq [DUCQ 1999], nous n'allons retenir que les trois premières.

En effet, les réductions de systèmes tolérants aux fautes et en boucle correspondent au monde purement informatique. Reste le cas de la réduction de système de type réseau. Ici, un système de type réseau correspond à un sous-processus. Or, dans le cas qui nous préoccupe, il s'agira dans la plupart des cas de deux tâches : la tâche de transmission des données et la tâche de mise en conformité de ces données (tâche à non-valeur ajoutée). La notion de sous-processus est clairement inappropriée pour un nombre si peu important de tâches.

Nous ne retiendrons donc que les trois règles de réduction suivantes : la réduction séquentielle, la réduction conditionnelle (réduction en OU) et la réduction parallèle (réduction en ET).

On réduit ainsi les activités A_i en l'unique activité A.

La réduction séquentielle : dans ce type d'agrégation, chaque activité se déroule à la suite de la précédente et une fois celle-ci terminée. Dans la Figure 35, l'activité A_1 se déroule intégralement, puis c'est au tour de l'activité A_2 .

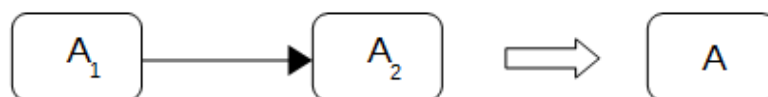


Figure 35 : Réduction d'activités séquentielles

La réduction en OU : dans ce type d'agrégation, on peut effectuer indifféremment l'activité 2 ou 3, les deux activités ayant des caractéristiques de performance différentes. La réduction en OU est illustrée dans la Figure 36.

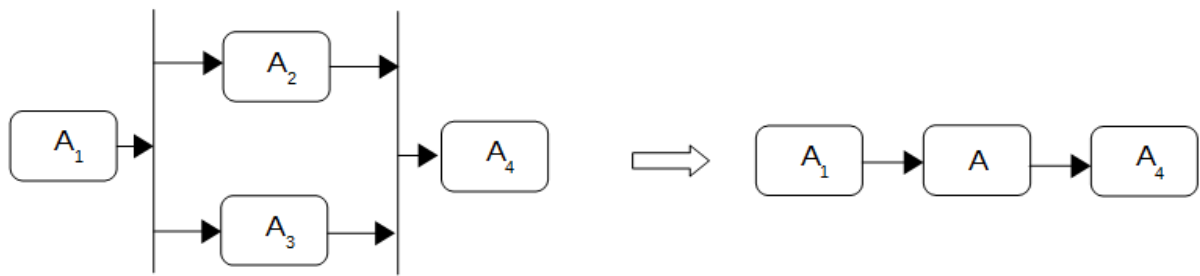


Figure 36 : Réduction en OU

La réduction en ET : dans ce type d'agrégation, on doit effectuer les activités 2 et 3, les deux activités ayant des caractéristiques de performance différentes, comme on peut le voir dans la Figure 37 :

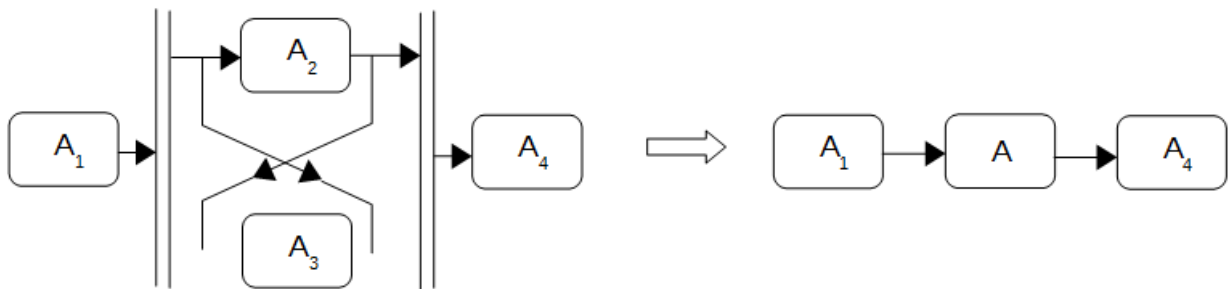


Figure 37 : Réduction en ET

3.4.2 AGREGATION DES MESURES DE PERFORMANCE

A chaque type de réduction correspond une formule de calcul de l'agrégation de performance pour chacune des dimensions. Nous allons définir les formules qui nous serviront à calculer les valeurs de la performance des activités agrégées.

L'agrégation séquentielle :

Le coût et le délai sont égaux à la somme des coûts et des durées respectivement, tandis que la qualité et la disponibilité sont égales aux produits respectifs :

$$\text{Coût : } C(A) = \sum_{i=1}^N C(A_i)$$

$$\text{Délai : } D(A) = \sum_{i=1}^N D(A_i)$$

$$\text{Qualité : } Q(A) = \prod_{i=1}^N Q(A_i)$$

$$\text{Disponibilité : } Di(A) = \prod_{i=1}^N Di(A_i)$$

L'agrégation en OU :

Les valeurs du coût et du délai sont égales au maximum des coûts et des délais. Celles de la qualité et la disponibilité sont égales au minimum (on se place dans le cas le plus pessimiste) :

$$\text{Coût : } C(A) = \text{Max}(C(A_i))$$

$$\text{Délai : } D(A) = \text{Max}(D(A_i))$$

$$\text{Qualité : } Q(A) = \text{Min}(Q(A_i))$$

$$\text{Disponibilité : } Di(A) = \text{Min}(Di(A_i))$$

L'agrégation en ET :

La valeur du coût est égale à la somme des coûts. La valeur du délai est égale au maximum des valeurs du délai. Celles de la qualité et la disponibilité sont égales au produit des qualités et des durées :

$$\text{Coût : } C(A) = \sum_{i=1}^N C(A_i)$$

$$\text{Délai : } D(A) = \text{Max}(D(A_i))$$

$$\text{Qualité : } Q(A) = \prod_{i=1}^N Q(A_i)$$

$$\text{Disponibilité : } Di(A) = \prod_{i=1}^N Di(A_i)$$

Revenons à notre exemple illustratif car nous sommes maintenant en mesure d'afficher sur les modèles les mesures de la performance dans l'étape 4.

Le modèle AS IS est représenté par la Figure 38.

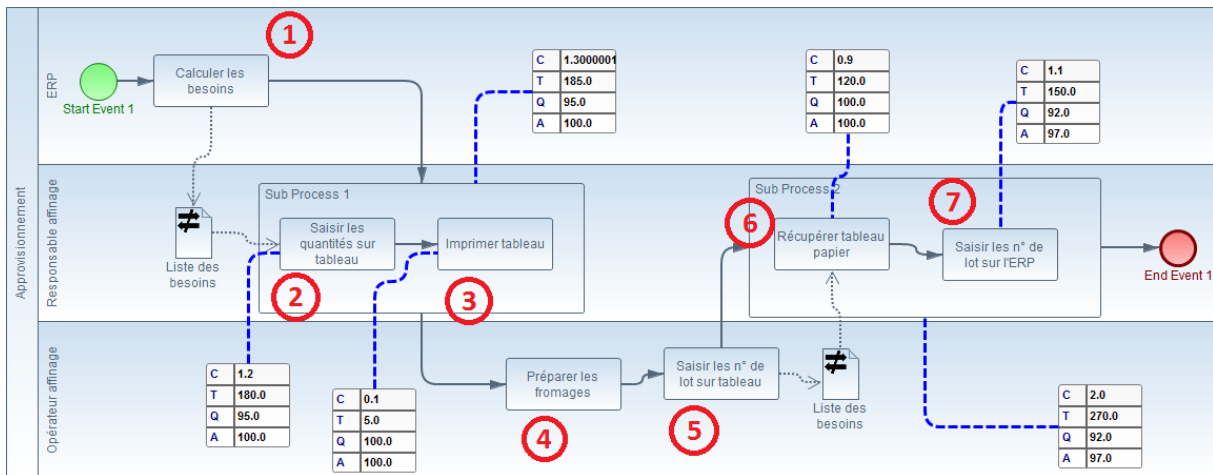


Figure 38 : Modèle AS IS avec les mesures de la performance

Pour chacun des échanges de données, deux tâches sont impliquées. Les tâches 2 et 3 pour le premier échange et les tâches 6 et 7 pour le second. Ces deux couples de tâches ont donc été regroupés dans deux sous-processus. Les mesures de la performance ont été ajoutées pour chacune des tâches impliquées par les échanges de données, ainsi que pour les deux sous-processus. Les valeurs affichées dans les tableaux associés aux sous-processus ont été calculées à l'aide des formules présentées ci-avant. Il s'agit ici d'agrégation séquentielle, les tâches étant accomplies l'une après l'autre.

Le modèle TO BE est représenté dans la Figure 39.

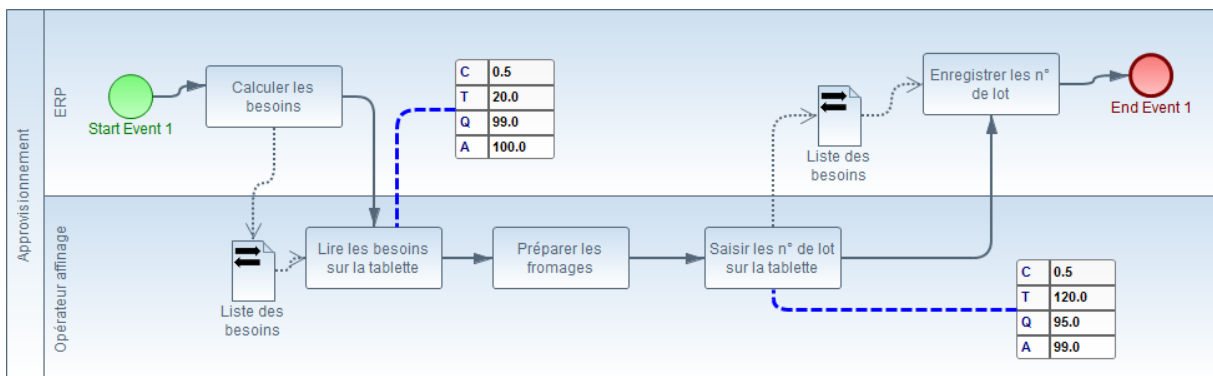


Figure 39 : Modèle TO BE avec les mesures de la performance

Ici, les tâches à non-valeur ajoutée ayant été éliminées, une seule tâche est impliquée dans chacun des échanges de données. Une extension *performanceMeasurement* est associée à chacune d'entre elles.

Nous sommes donc en mesure de comparer les valeurs de la performance des tâches impliquées dans les échanges de données avant et après l'établissement de l'interopérabilité.

Après avoir introduit les méthodes d'agrégation de la performance et présenté les formules permettant de la calculer, nous allons aborder le sujet de la gestion de la qualité des

données et présenter la méthode que nous proposons. Cette méthode constitue l'étape 6 de la démarche globale (l'étape 5, étant hors du contexte de cette thèse, ne sera pas abordée).

3.5 GESTION DE LA QUALITE DES DONNEES

L'intérêt d'assurer l'interopérabilité des données est double :

- Assurer une qualité des données optimales après les échanges entre différents systèmes.
- Faire disparaître les tâches à non-valeur ajoutée induites par l'absence d'interopérabilité.

Or, si l'établissement de l'interopérabilité entraînera automatiquement la suppression des tâches à non-valeur ajoutée, la qualité des données n'en sera pas pour autant totalement assurée. Certes, le seul fait d'éviter la ressaisie des données évitera un grand nombre d'erreurs. Mais si la donnée n'est pas de qualité avant les différents échanges, elle le restera tout au long du processus.

Les extensions de BPMN 2.0 que nous venons de proposer vont nous permettre d'avoir une vue globale de la qualité des données. En effet, les valeurs des indicateurs de qualité et de disponibilité des données qui y sont affichées englobent la totalité des données incluses dans le document échangé. Même si cette vue globale est indispensable, elle ne nous permet d'appréhender uniquement l'existence et la nature du problème de la qualité des données car les valeurs de la qualité des différentes données sont fusionnées dans une valeur unique. De ce fait, notre approche est globale et qualitative. Or, si la perception d'un problème est une condition sine qua non de sa résolution, une connaissance des causes de ce problème en est une autre.

Il nous faut donc compléter notre proposition par un autre outil qui nous offrira la possibilité d'avoir une approche détaillée et quantitative. Il devra nous permettre d'avoir une vision détaillée de chacune des données impliquée dans le processus. On pourra ainsi déterminer les motifs et les conséquences de leur non-qualité dans le but d'élaborer une action corrective.

Afin de pouvoir œuvrer à une meilleure qualité des données, nous nous proposons de définir une méthode inspirée de l'AMDEC.

Nous allons tout d'abord présenter cette méthode de maintenance. Ensuite nous exposerons notre proposition.

3.5.1 AMDEC

L'AMDEC (Analyse des Modes de Défaillance, de leurs Effets et de leur Criticité) est une méthode développée par l'armée américaine dans les années 1950 et adaptée à différents secteurs industriels à partir de la décennie suivante. Son but est d'identifier l'ensemble des

défaillances potentielles d'un système ou d'un processus, d'en évaluer l'importance dans le but d'y amener une action corrective avant qu'elles ne se produisent et ainsi en améliorer la fiabilité. Elle est utilisée dans de nombreux secteurs de l'industrie et des services tels que l'aéronautique, le secteur automobile, la santé ou encore les logiciels. Elle peut être utilisée pour améliorer la fiabilité d'un produit, d'une machine, d'un processus, etc.

Il existe plusieurs types d'AMDEC. On citera par exemple :

- AMDEC fonctionnelle : recherche des modes de défaillance et de ses causes à l'étape de la conception ;
- AMDEC produit : vérification de la viabilité d'un produit en fonction des demandes des clients en termes de fiabilité ;
- AMDEC processus : identification des risques liés aux défaillances d'un produit dues au processus de fabrication ;
- AMDEC moyen de production : analyse les risques liés aux défaillances de la chaîne de production ;
- AMDEC flux : analyses les risques liés aux ruptures de flux : approvisionnement, temps de réaction et de correction et coûts liés à un fonctionnement normal.

Cette méthode s'applique en quatre étapes :

1. initialisation (mise en place de la méthode) : au cours de cette étape, on compose l'équipe qui mènera le projet et on identifie le système et son environnement, ce qui revient à délimiter le périmètre du projet et préciser les objectifs que l'on souhaite atteindre ;
2. analyse fonctionnelle : on y identifie le système, ses fonctions et ses composants (sous-ensembles et éléments) ;
3. analyse qualitative et quantitative : identification des modes de défaillance et détermination de leurs causes et effets pour chacun des composants du système. Évaluation de la criticité et propositions d'actions correctives (ces points seront développés plus bas) ;
4. synthèse : hiérarchisation des défaillances, établissement d'une liste des points critiques puis d'une liste de recommandations.

Lors de la troisième étape, on est amenés à évaluer la criticité du mode de défaillance. Celle-ci est calculée à l'aide de la formule suivante :

$$C = F \times N \times G$$

où :

- F : Fréquence d'apparition de la défaillance : elle doit représenter la probabilité d'apparition du mode de défaillance résultant d'une cause donnée.

- N : Fréquence de non détection de la défaillance (ou détectabilité) : elle doit représenter la probabilité de ne pas détecter la cause ou le mode de défaillance avant que l'effet survienne.
- G : Gravité des effets de la défaillance : la gravité représente la sévérité relative à l'effet de la défaillance.

Ces trois paramètres sont évalués pour chacun des modes de défaillance sur une échelle, par exemple de 1 à 4 ou de 1 à 10 avant d'être multipliés entre eux.

On présente les résultats sous forme d'un tableau (aussi appelé grille AMDEC) comme l'illustre le Tableau 5 dans le cas d'un AMDEC processus.

Étape du processus	Type de défaillance	Causes	Effets	F	N	G	C	Actions correctives

Tableau 5 : Exemple de grille AMDEC

En fonction du résultat de la criticité, on déduit les mesures à prendre, ainsi que leur priorité.

Par la suite, on peut compléter la grille AMDEC afin qu'elle serve de support à une démarche d'amélioration continue. Le Tableau 6 nous montre une telle grille.

Étape du processus	Type de défaillance	Causes	Effets	Évaluation				Actions correctives	Résultat			
				F	N	G	C		F'	N'	G'	C'

Tableau 6 : Exemple de grille AMDEC pour l'amélioration continue

où F', N', G' et C' sont les nouvelles valeurs de F, N, G et C une fois les actions correctives réalisées.

On peut ainsi, grâce aux nouvelles valeurs calculées, vérifier l'efficacité des actions entreprises, affecter de nouvelles priorités et y appliquer les actions correctives adéquates, suivant ainsi les étapes d'un PDCA (Plan, Do, Check, Act).

En résumé, on peut dire que l'AMDEC permet de réaliser la mise en place des plans de maintenance préventive en nous offrant des outils de décision et d'amélioration.

3.5.2 ADAPTATION DE L'AMDEC A LA QUALITE DES DONNEES

Nous venons de présenter l'outil de sûreté de fonctionnement et de gestion de la qualité qu'est l'AMDEC. Nous avons vu que cet outil se déclinait en plusieurs versions pour répondre à des besoins différents : fonctionnelle, produit, processus, etc. Nous proposons de le décliner dans le but de déterminer les priorités d'actions à mener pour améliorer la qualité des données. Nous proposons le nom « AMDEC données » pour cette méthode.

Une grille AMDEC données sera mise en place pour chacun des échanges de données présent dans le processus entre deux étapes de celui-ci. Donc si le même document est échangé plusieurs fois au cours du processus, une grille sera mise en place pour chacun des échanges et chacune des données qui y est comprise sera présente dans la grille. Cette grille sera complétée par une grille globale qui agrègera toutes les données présentes dans le processus étudié.

Nous avons choisi de conserver les étapes de la méthode AMDEC en les modifiant légèrement pour les faire correspondre au domaine des données. Dans un premier temps nous présenterons les quatre étapes de manière synthétique, et nous les détaillerons par la suite.

Présentation des étapes de la méthode AMDEC données :

1. initialisation, mise en place de la méthode : on constitue l'équipe qui mènera la réflexion. On définit le périmètre du projet (entreprise, services) d'amélioration de la qualité des données et on précise les objectifs que l'on souhaite atteindre ;
2. décomposition fonctionnelle : on identifie les documents présentant des problèmes de qualité, ainsi que les données qui y sont présentes ;
3. pour chaque donnée, identification des causes et des effets de la non-qualité. Évaluation de la criticité et propositions d'actions correctives ;
4. synthèse : hiérarchisation des défaillances de qualité, établissement d'une liste des points critiques puis d'une liste de recommandations.

Nous allons maintenant détailler les étapes de la méthode AMDEC données.

3.5.2.1 Étape 1 : initialisation

L'équipe qui mènera la réflexion doit comprendre des acteurs de tous les métiers concernés par la démarche afin que tous les aspects de la problématique soient pris en compte.

Le périmètre du projet correspondra à celui de la résolution des problèmes d'interopérabilité. Comme nous l'avons déjà dit plus haut, corriger des problèmes d'interopérabilité sans s'assurer de la qualité des données ne serait pas cohérent. La mise en place de l'AMDEC données doit donc compléter l'établissement de l'interopérabilité et s'appliquer sur le même périmètre.

Il en va de même en ce qui concerne les objectifs que l'on souhaite atteindre.

Dans le cas de notre exemple illustratif, l'équipe doit inclure entre autres les responsables de l'affinage et de l'expédition. Le périmètre du projet inclut l'ERP (car c'est lui qui calcule les besoins) et l'affinage.

3.5.2.2 Étape 2 : analyse fonctionnelle

Dans la méthode AMDEC, lors de cette étape on identifie le système, ses fonctions et ses composants (sous-ensembles et éléments).

Dans notre cas, le système correspond au(x) système(s) d'information, car nous travaillons dans le domaine des données.

Les composants de ce système sont les documents échangés qui présentent un problème de qualité et chacune des données qui y sont contenues.

Les modélisations réalisées à l'aide des extensions présentées précédemment vont nous aider dans la réalisation de cette étape.

En ce qui concerne l'exemple de l'affinage, les composants du système sont la liste des besoins fournie par l'ERP (et ressaisie par le responsable affinage) et cette même liste une fois les numéros de lots renseignés.

3.5.2.3 Étape 3 : analyse qualitative et quantitative

Identification des modes de défaillance, détermination des causes et des effets de la non-qualité pour chacune des données identifiées. Évaluation de la criticité et propositions d'actions correctives.

Identification des modes de défaillance : avant toute chose, il nous faut connaître le type de la défaillance pour chacun des problèmes de qualité des données. Il n'existe pas de liste exhaustive des composants de la qualité des données, et plus de deux cent dimensions en ont été recensées [WANG 1995]. Il en va donc de même pour leurs modes de défaillance. Dans ses travaux, Laure Berti-Équille nous propose une liste des principales dimensions les plus consensuelles [BERTI-EQUILLE 2004]. Cette liste est visible dans le Tableau 7. La liste des modes de défaillance dépendra évidemment du contexte métier considéré.

Qualité du modèle conceptuel des données	Lisibilité	Caractère qui confère au modèle conceptuel une facilité de lecture par sa clarté et sa minimalité (degré de factorisation)
	Complétude	Caractère qui confère au modèle conceptuel une couverture de l'ensemble des besoins
	Expressivité	Caractère qui confère au modèle conceptuel une richesse descriptive pour représenter naturellement les besoins et la réalité
	Correction	Caractère qui confère au modèle conceptuel une conformité par rapport aux spécifications
	Traçabilité	Documentation détaillée et historique de la conception et de l'évolution du modèle conceptuel des données
	Simplicité	Caractère qui restreint le modèle conceptuel à un ensemble minimal d'éléments nécessaires
Qualité des processus de traitement des données	Sécurité	Ensemble des facteurs portant sur l'aptitude du système à préserver les données de toute manipulation malveillante ou hasardeuse
	Fiabilité	Ensemble des facteurs portant sur l'aptitude du système à maintenir les données dans des conditions précises et pendant une période déterminée (tolérance aux pannes et récupération des données)
	Accessibilité	Ensemble des facteurs sur l'aptitude du système à rendre les données consultables et manipulables dans des temps adéquats
	Disponibilité	Ensemble des facteurs portant sur l'effort nécessaire pour l'utilisation des données et sur l'évaluation individuelle de cette utilisation par un ensemble défini ou implicite d'utilisateurs
	Maintenabilité	Ensemble des facteurs portant sur l'effort nécessaire pour faire des modifications sur les données et sur leur schéma.
	Interopérabilité	Ensemble des facteurs portant sur l'aptitude du système à permettre et faciliter l'échange des données
	Confidentialité	Ensemble des facteurs portant sur l'aptitude du système à assurer que les données ne soient accessibles que par ceux dont l'accès est autorisé
Qualité des instances ou valeurs des données	Complétude	Quantité de valeurs renseignées
	Cohérence	Quantité de valeurs satisfaisant l'ensemble des contraintes ou règles de gestion définies
	Exactitude	Quantité de valeurs correctes et sans erreur
	Fraîcheur	Ensemble des facteurs qui capturent le caractère récent et le caractère d'actualité d'une donnée entre l'instant où elle a été extraite ou créée dans la base et l'instant où elle est présentée à l'utilisateur

Tableau 7 : Principales dimensions de la qualité des données

Nous l'avons dit, notre exemple illustratif est volontairement simple. Nous n'y avons identifié qu'un mode de défaillance : la non-exactitude.

Détermination des causes de la non-qualité : comme dans le cas d'un AMDEC classique, il faut ici sonder les participants au processus qui utilisent les données afin de déterminer le pourquoi de la non-fiabilité de ces données. La consultation des fournisseurs de ces données, qu'ils soient internes ou externes, est aussi indispensable. Il faut aussi envisager les causes de possible non-qualité, c'est-à-dire de défaillance non encore avérée ou détectée.

Dans notre exemple, les participants à sonder sont les membres du service affinage, le responsable des expéditions et le responsable du système d'information. Les causes de la non-qualité sont au nombre de deux : les erreurs de saisie et un mauvais calcul des besoins (ERP mal paramétré).

Détermination des effets de la non-qualité : pour chacune des données identifiées, il faut mesurer les conséquences entraînées par ce manque de qualité. Ces conséquences sont à évaluer par rapport à l'incidence causée sur la suite du processus et donc sur le client, le terme client signifiant l'utilisateur de la suite du processus, celui du processus suivant ou bien le client final. L'effet peut se faire sentir lors de plusieurs étapes du processus pour une même donnée. L'effet initial peut être anodin et l'effet final important. L'effet que l'on prendra en compte sera alors l'effet final, celui qui est le plus grave. On pourra évaluer les conséquences à l'aide d'une grille d'évaluation, d'une échelle, etc.

Dans notre exemple illustratif, les effets de la non-qualité sont soit un manque de fromages au dépôt des expéditions (et donc une impossibilité à honorer des commandes), soit au contraire une trop grande quantité dans un local où l'espace libre est précieux.

Évaluation de la criticité : à ce stade, il faut remplir la grille AMDEC données. Celle-ci est légèrement modifiée par rapport à celle que nous présentait le Tableau 4.

Dans la grille que nous proposons, nous remplaçons la première colonne, correspondant à l'étape du processus, par la colonne correspondant à la donnée considérée.

Les autres colonnes correspondent à celles que nous avons vues plus haut.

Le Tableau 8 nous donne une vue de la grille d'évaluation de la criticité pour la non-qualité des données que nous proposons :

Donnée	Type de défaillance	Causes	Effets	Évaluation				Action corrective	Résultat			
				F	N	G	C		F'	N'	G'	C'

Tableau 8 : Proposition de grille AMDEC qualité

Les définitions des paramètres F, N et G étant les suivantes :

- F : Fréquence de la cause de non-qualité : elle correspond à la fréquence du manque de fiabilité de la donnée considérée. Pour déterminer la fréquence, il est préférable de s'appuyer sur des relevés plutôt que sur le seul ressenti des utilisateurs, car ceux-ci auront tendance à la surestimer, ou au contraire à la sous-estimer, en fonction de certains paramètres. En particulier le désagrément personnel occasionné (par exemple si ce sont eux qui sont chargés de retrouver la donnée en cause ou bien si cette absence de qualité impacte la tâche qu'eux-mêmes accomplissent ou non) jouera un grand rôle dans leur estimation. Encore faut-il que de tels relevés existent. Si ce n'est pas le cas, il est indispensable d'en mettre en place pour la phase d'amélioration continue.

Afin d'attribuer une valeur à cette variable, on peut s'appuyer sur une grille d'évaluation telle celle illustrée par le Tableau 9.

Niveau	Critère	Valeur de la fréquence F
Très rare	1 fois par an	1
Rare	1 fois par mois	2
Peu Fréquent	1 fois par semaine	3
Fréquent	2 à 3 fois par semaine	4
Très fréquent	1 à plusieurs fois par jour	5

Tableau 9 : Grille d'évaluation de la fréquence de non-qualité

Les valeurs des niveaux et des critères doivent être ajustés selon le contexte. On peut aussi définir un plus grand nombre de niveaux et ajuster les valeurs de la fréquence en conséquence.

Nous utiliserons la grille du Tableau 9 dans notre exemple illustratif pour évaluer la fréquence.

- N : Fréquence de non détection (ou détectabilité) de la cause de non-qualité : elle doit représenter la probabilité de ne pas détecter la cause ou le mode de défaillance avant que l'effet ne survienne. Son évaluation n'est pas chose aisée. On pourra s'appuyer sur le calcul suivant : Si P1 est la probabilité de la cause non-qualité de D1 et P2 celle de D2, la probabilité P de non détection de non-qualité de D2 en cas d'absence de D1 est obtenue par la formule :

$$P = P1 \times P2$$

On peut généraliser cette formule à la donnée D_n , $n^{\text{ième}}$ donnée présente dans le processus, et calculer sa probabilité de non détection de la cause de non-qualité P à l'aide de la formule :

$$P = \prod_{i=1}^n P_i$$

On utilisera ensuite une grille d'évaluation de la fréquence de non détection de la cause de non-qualité à l'aide d'une grille du type de celle du Tableau 10.

Probabilité de non détection	N
$P \leq 0,25$	1
$0,25 \leq P \leq 0,5$	2
$0,5 \leq P \leq 0,75$	3
$0,75 \leq P \leq 1$	4

Tableau 10 : Grille d'évaluation de la non-détection

Les valeurs des bornes de la probabilité ainsi que celles de N sont bien évidemment à ajuster en fonction du contexte.

Nous utiliserons la grille du Tableau 10 dans notre exemple illustratif pour évaluer la fréquence de non détection.

- G : Gravité des effets de la non-qualité : la gravité représente la sévérité relative à l'effet de la non-qualité de la donnée. Évaluer la gravité revient à quantifier l'impact occasionné au client ou au processus suivant. Dans certains cas, l'impact sera uniquement interne à l'entreprise. Par exemple si on perd du temps à retrouver la donnée correcte mais que le produit arrive à temps et avec la même qualité au client final, il n'y aura aucun impact pour celui-ci (en revanche, la personne qui a corrigé l'erreur a perdu du temps qu'elle aurait consacré à une autre activité, elle a donc été impactée). Si au contraire la non-qualité entraîne un retard de livraison ou une qualité moindre du produit, le client final sera impacté.

Les critères d'estimation de la gravité doivent être choisis en fonction du contexte de l'entreprise. On pourra y retrouver selon le cas le coût financier engendré, la baisse de qualité du produit, le retard de livraison, les risques éventuels sur la santé, etc.

Il est important de dé-corréler la gravité et la fréquence, car on a souvent tendance à penser qu'une défaillance qui se produit couramment est plus grave qu'une autre qui se produit rarement. Or, le nombre d'occurrence est pris en compte dans l'estimation de la fréquence. Il ne doit donc jouer aucun rôle dans celui de la gravité.

Le critère de Farmer illustré par la Figure 40 nous aide à définir les notions de risque faible (ou acceptable) ou élevé (ou inacceptable) [FARMER 1967] :

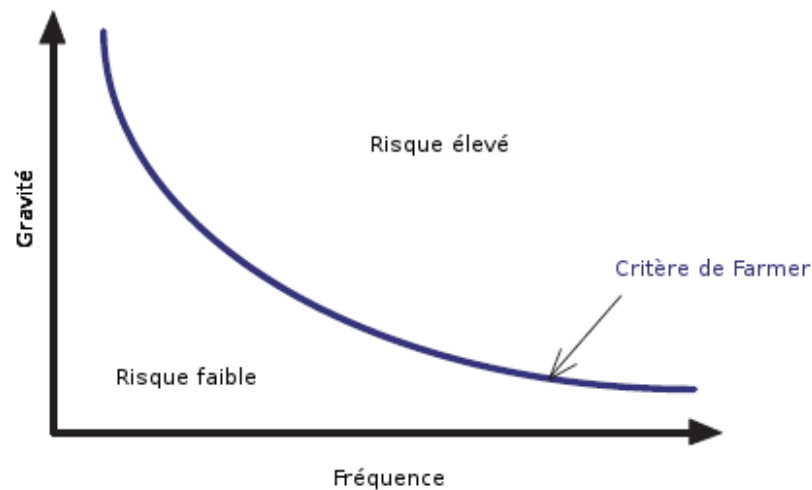


Figure 40 : Critère de Farmer

Le Tableau 11 nous montre un exemple de grille d'évaluation de la gravité.

Niveau	Critère	Valeur de la gravité G
Très faible	Temps peu important passé à retrouver la donnée correcte (moins de 5 minutes)	1
Faible	Temps important passé à retrouver la donnée correcte (plus de 5 minutes)	2
Grave	Donnée impossible à corriger. Légère insatisfaction du client.	3
Très grave	Donnée impossible à corriger. Insatisfaction importante du client.	4

Tableau 11 : Grille d'évaluation de la gravité

Dans ce cas aussi, le nombre de niveaux (et donc les valeurs de la gravité) ainsi que les critères d'évaluation sont à adapter en fonction du contexte.

Nous utiliserons la grille du Tableau 11 dans notre exemple illustratif pour évaluer la gravité.

Les valeurs des variables F, N et G étant déterminées pour chacune des données identifiées, on peut remplir la grille et calculer la criticité qui nous aidera à fixer les priorités dans la résolution de la non-qualité des données dans l'étape suivante. La criticité se calcule à l'aide de la formule suivante :

$$C = F \times N \times G$$

On pourra faire apparaître dans les modèles BPMN 2.0 des liens cliquables qui amènent directement vers les différentes grilles d'évaluation constituées afin de passer de la vue globale à la vue détaillée des données.

Une des deux grilles utilisées dans notre exemple illustratif est présentée dans le Tableau 12.

Donnée	Type de défaillance	Causes	Effets	Évaluation				Action corrective	Résultat			
				F	N	G	C		F'	N'	G'	C'
N° de lot	Donnée inexacte	Erreur de saisie	Problème de traçabilité	3	4	3	36	Sensibiliser les opérateurs	2	4	3	24

Tableau 12 : Grille d'évaluation de la criticité de l'exemple illustratif

On n'y voit qu'une donnée présente, le numéro de lot. Les valeurs de F, N et C ont été évaluées à l'aide des grilles d'évaluation vues précédemment.

3.5.2.4 Étape 4 : synthèse

Une fois les grilles d'évaluation de la criticité renseignées pour chacun des échanges de données, on établira une grille de synthèse regroupant l'ensemble des données échangées tout au long du processus.

Pour celles n'apparaissant qu'une seule fois, il suffira de reporter telle quelle la ligne leur correspondant dans sa grille d'origine vers la grille de synthèse.

Par contre, se pose la question des données échangées à plusieurs reprises au cours du processus. Si les actions correctives préconisées diffèrent, la donnée devra apparaître pour chacune des actions envisagées afin de corriger toutes les sources d'erreur. La valeur de la criticité calculée nous aidera à établir la priorité des actions correctives à mener. S'il n'y a qu'une action corrective recommandée pour plusieurs occurrences de la donnée considérée, mais que des valeurs différentes ont été calculées en ce qui concerne la criticité, il nous faut déterminer la valeur de la criticité qui sera retenue. Nous avons opté pour sélectionner la ligne dont la criticité est la plus haute. En effet, tout autre choix que le maximum affecterait la

valeur de la criticité au point où elle est la plus haute et amènerait ainsi à minimiser la priorité de son action corrective.

Nous avons ajouté une colonne à la grille de synthèse correspondant au document contenant le nom du document dans lequel apparaît la donnée présente. La grille de synthèse est représentée par le Tableau 13.

Document	Donnée	Type de défaillance	Causes	Effets	Évaluation				Action corrective	Résultat			
					F	N	G	C		F'	N'	G'	C'

Tableau 13 : Grille de synthèse

Une fois la grille de synthèse renseignée en totalité, on va pouvoir hiérarchiser les défaillances de qualité. En effet, on ne peut se fier aveuglément à la criticité calculée, elle n'est qu'une aide à la décision. De nombreux autres paramètres que ceux qui sont présents dans le calcul de la criticité entrent en ligne de compte dans ce classement, comme par exemple le volume de commandes concernées ou encore le chiffre d'affaires engendré par tel ou tel produit, etc. On ne mettra pas sur un pied d'égalité la criticité d'une donnée concernant un produit représentant un volume important de la production avec celle concernant un produit fabriqué occasionnellement. Il faut donc ici faire preuve de discernement au moment de classer les défaillances à traiter en priorité, car on peut aussi préférer contenter un client occasionnel dans un but commercial (pour le fidéliser) plutôt qu'un client habituel.

A l'aide de cette hiérarchisation, on établira une liste des points critiques du processus en ce qui concerne la qualité des données. Il est possible pour cela de fixer un seuil de criticité. Les données pour lesquelles la criticité est supérieure à ce seuil devront faire l'objet d'une action corrective, les autres pouvant être négligées. Ce seuil devra être fixé en fonction des échelles de valeurs choisies pour les valeurs F, N et G, sans quoi il serait possible qu'aucune des données ne l'atteigne. Il faut toutefois rester prudent avec un tel critère. En effet, si on fixe la valeur de ce seuil à 50 et que l'on calcule pour les données D1 et D2 les valeurs de la criticité 49 et 51 respectivement, il serait dangereux d'écarter une action corrective pour D1 et d'en recommander une pour D2. Notamment parce qu'une part de subjectivité existant dans l'évaluation des paramètres F, N et G, la valeur de la criticité calculée pourrait varier en fonction de l'utilisateur du processus qui a été sondé par exemple ou de la période sur laquelle ont été relevées ces valeurs, ce qui pourrait inverser les valeurs de la criticité de D1

et D2. Dans tous les cas, il faut faire preuve de souplesse et de discernement si l'on décide de fixer un tel seuil.

On proposera alors une liste de recommandations d'actions correctives pour les points critiques identifiés permettant d'améliorer la fiabilité des données correspondantes. Ces actions correctives seront reportées dans la grille de synthèse présentée dans le Tableau 13.

Un lien cliquable correspondant à cette grille de synthèse pourra aussi être placé au niveau des modèles réalisés à l'aide de BPMN 2.0.

La grille de synthèse de notre exemple illustratif est présentée par le Tableau 14.

Document	Donnée	Type de défaillance	Causes	Effets	Évaluation				Action corrective	Résultat			
					F	N	G	C		F'	N'	G'	C'
Liste des besoins 1	Quantité	Donnée inexacte	Mauvais calcul des besoins	Manque de fromages à l'expédition	2	4	3	24	Revoir les paramètres du calcul des besoins	1	4	3	12
Liste des besoins 2	N° de lot	Donnée inexacte	Erreur de saisie	Problème de traçabilité	3	4	3	36	Sensibiliser les opérateurs	2	4	3	24

Tableau 14 : Grille de synthèse de l'exemple illustratif

On y retrouve la ligne correspondant au document « Liste des besoins 2 » (document « liste des besoins » lors du deuxième échange) ainsi que celle du premier échange (grille non représentée ici).

3.5.2.5 Amélioration continue

Une fois cet outil que représente l'AMDEC données mis en place, la grille de synthèse nous permettra de suivre l'évolution de la non-qualité des données après chaque action corrective effectuée.

On réévaluera pour cela les paramètres F, N et G suite à la mise en place des actions correctives et on obtiendra les valeurs de F', N' et G' qui nous serviront à calculer la nouvelle criticité C' pour chacune des données présentes dans la grille. Cette nouvelle criticité nous aidera à hiérarchiser les défaillances de fiabilité, à établir une nouvelle liste de points critiques et à proposer de nouvelles actions correctives.

3.6 SYNTHÈSE GÉNÉRALE

Nous venons de définir les extensions de BPMN 2.0 *dataInteroperabilityBarrier* et *dataInteroperabilityResolved* qui vont nous offrir la possibilité de représenter et donc de visualiser les échanges de données présentant un problème d'interopérabilité ainsi que ceux pour lesquels l'interopérabilité a été établie. Nous avons aussi défini l'extension *performanceMeasurement* qui permettra d'afficher les mesures de la performance au sein de nos modèles.

Nous avons aussi présenté les méthodes de réduction des processus et d'agrégation de la performance que nous utiliserons pour fusionner les mesures de la performance des tâches appartenant à un sous-processus.

Nous avons enfin proposé la méthode AMDEC données qui permettra de visionner les données individuellement, d'en déterminer les causes et les conséquences et enfin d'en calculer la criticité dans le but de définir des priorités ainsi que des actions correctives pour en améliorer la qualité. Les grilles d'évaluation de la criticité en résultant correspondront à chacun des échanges de données intervenant dans le processus et une grille de synthèse se rapportera à l'ensemble du processus.

La Figure 41 synthétise l'ensemble de ces contributions conceptuelles appliquées à notre exemple illustratif. On peut y voir le modèle TO BE où sont représentés à l'aide de l'extension *dataInteroperabilityResolved* les deux échanges de données pour lesquels l'interopérabilité a été établie. Les deux grilles d'évaluation AMDEC données y sont aussi présentes. Chacune d'elles est reliée par une flèche rouge pointillée au document échangé lui correspondant. Ces deux grilles pointent vers la grille de synthèse à l'intérieur de laquelle leurs données sont fusionnées.

Document	Donnée	Type de défaillance	Cause	Effets	Evaluation				Action corrective	Résultat			
					F	N	G	C		F'	N'	G'	C'
Liste des besoins 1	Quantité	Donnée inexacte	Mauv ais calcul des besoins	Manque de fromages à l'expédition	2	4	3	24	Revoir les paramètres du calcul des besoins	1	4	3	12
Liste des besoins 2	N° de lot	Donnée inexacte	Erreur de saisie	Problème de traçabilité	3	4	3	36	Sensibiliser les opérateurs	2	4	3	24

Donnée	Type de défaillance	Cause	Effets	Evaluation				Action corrective	Résultat			
				F	N	G	C		F'	N'	G'	C'
Quantité	Donnée inexacte	Mauv ais calcul des besoins	Manque de fromages à l'expédition	2	4	3	24	Revoir les paramètres du calcul des besoins	1	4	3	12

Donnée	Type de défaillance	Cause	Effets	Evaluation				Action corrective	Résultat			
				F	N	G	C		F'	N'	G'	C'
N° de lot	Donnée inexacte	Erreur de saisie	Problème de traçabilité	3	4	3	36	Sensibiliser les opérateurs	2	4	3	24

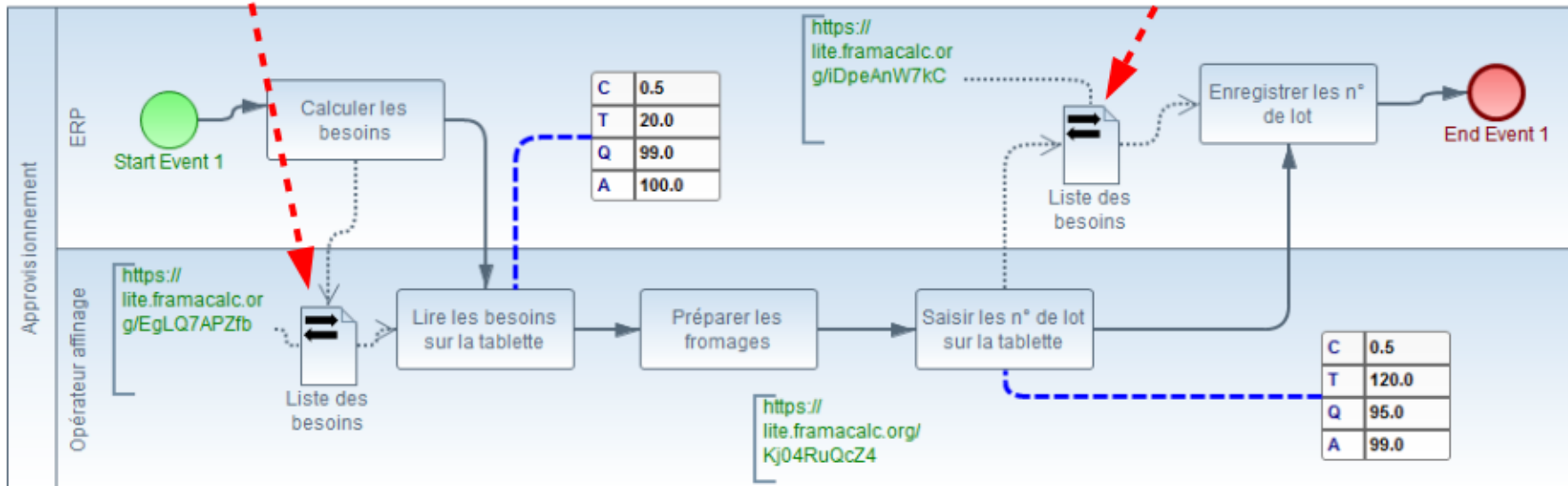


Figure 41 : Schéma de synthèse des contributions conceptuelles de l'exemple illustratif

3.7 CONCLUSION

Au cours de ce chapitre, nous avons proposé la création de deux extensions durables de BPMN 2.0. La première permettant de représenter l'interopérabilité dans les échanges de données entre différents systèmes d'information et la seconde d'afficher la performance au niveau du coût, du délai, de la qualité et de la disponibilité pour une tâche ou un ensemble de tâches. Pour chacune d'entre elles nous avons proposé une représentation graphique. Pour représenter les échanges de données présentant un problème d'interopérabilité, nous proposons de reprendre l'icône représentant les échanges de données et d'y inclure un symbole représentant l'interopérabilité non établie. En ce qui concerne les échanges de données pour lesquels l'interopérabilité a été établie, nous faisons de même avec un symbole représentant l'interopérabilité assurée. Enfin pour afficher les valeurs des mesures de la performance du coût, du délai, de la qualité et de la disponibilité nous proposons un tableau que nous relierons à la tâche ou au sous-processus concerné.

Nous avons aussi présenté le mécanisme d'extension de BPMN sur lequel nous allons nous appuyer pour réaliser ces extensions. Les extensions *dataInteroperabilityBarrier* et *dataInteroperabilityResolved* vont permettre de signaler dans les modèles les points d'interopérabilité avant et après leur résolution. L'extension *performanceMeasurement* quant à elle offre la possibilité d'afficher les mesures de la performance, au niveau du coût, du délai, de la qualité et de la disponibilité des données, des tâches impliquées dans ces échanges de données. Nous serons ainsi en mesure d'apprécier les conséquences concrètes de l'interopérabilité.

Nous avons ensuite introduit la notion de réduction des processus ainsi que les méthodes d'agrégation de la performance. Ces méthodes nous permettront de calculer les valeurs des mesures de la performance d'un sous-processus. Nous en avons également présenté les formules de calcul. Nous pourrons ainsi afficher les valeurs des mesures de la performance d'un ensemble de tâches induites par un échange de données présentant un problème d'interopérabilité agrégées dans un seul tableau. Nous serons ainsi en mesure de faire une comparaison rapide avec celles du même échange de données une fois l'interopérabilité assurée et donc apprécier les bénéfices de la résolution de ce problème.

Ensuite, nous avons évoqué la gestion de la qualité des données. Dans ce but nous avons présenté la méthode AMDEC et en avons proposé une adaptation au cas de la qualité des données. Cette adaptation, que nous avons nommée AMDEC données nous permettra de calculer la criticité de la non-qualité des données. Pour cela nous avons proposé une grille d'évaluation de la criticité, ainsi que des grilles d'évaluation de la fréquence, de la non détectabilité et de la gravité, paramètres qui entrent en compte dans le calcul de la criticité. Nous avons aussi présenté une grille de synthèse à l'intérieur de laquelle les données présentes dans chacune des grilles d'évaluation seront regroupées. Nous avons vu comment cette grille allait nous permettre de hiérarchiser les défaillances de qualité, d'établir une nouvelle liste de points critiques et enfin de proposer de nouvelles actions correctives.

Nous avons enfin vu comment cette même grille allait nous aider à mener une démarche d'amélioration continue en ce qui concerne la qualité des données comprises dans les échanges pour lesquels nous œuvrons à établir l'interopérabilité.

Dans le chapitre suivant, nous allons tout d'abord déterminer l'outil de développement à l'aide duquel nous allons implémenter les différentes extensions de BPMN 2.0 que nous avons proposées.

Nous ferons ensuite une présentation de l'outil choisi.

Nous terminerons par décrire les grandes lignes de l'implémentation des nouvelles extensions.

Chapitre 4 : IMPLEMENTATION DANS BPMN2 MO- DELER

4.1 Introduction	96
4.1 Choix de BPMN2 Modeler	96
4.2 Présentation de BPMN2 Modeler	97
4.2.1 EDI Eclipse	98
4.2.2 EMF	98
4.2.3 Graphiti	98
4.2.4 BPMN 2 Modeler	99
4.3 Implémentation des extensions	102
4.3.1 Spécifications	102
4.3.2 Fonctionnement de BPMN 2 Modeler	103
4.3.3 Implémentation	103
4.3.4 Interface étendue	106
4.4 Conclusion	108

4.1 INTRODUCTION

Ce chapitre fait suite à la contribution conceptuelle, présentée dans la troisième partie de ce manuscrit, et propose l'implémentation des extensions de BPMN 2.0 qui permettent de visualiser les points d'interopérabilité et leurs conséquences dans l'objectif de leur résolution. Les extensions sont implémentées dans une représentation graphique.

Ce chapitre débute par l'argumentaire du choix d'utilisation de BPMN2 Modeler pour implémenter les différentes extensions que nous avons proposées. Ensuite il propose une présentation de l'outil sélectionné et des différentes technologies sur lesquelles il s'appuie. Nous décrivons enfin l'implémentation des extensions proposées dans cette thèse.

4.1 CHOIX DE BPMN2 MODELER

Pour choisir une solution logicielle BPMN support pour le développement de nos extensions, nous avons considéré plusieurs critères et effectué une revue de littérature. Notamment, nous avons trouvé particulièrement pertinente l'étude de Marco Roque [ROQUE 2014]. Lors de la réalisation de sa thèse, l'auteur a été amené récemment à créer des extensions de BPMN 2.0, et donc à sélectionner un outil de développement pour y parvenir. A cette fin, il a réalisé une étude pour rechercher un outil qui répondait aux prérequis suivants :

- fournir de bonnes capacités de modélisation et compatible avec BPMN 2.0 ;
- être développé en Java;
- être un projet open source efficace;
- être compréhensible (dans un souci d'extensibilité).

Pour réaliser cette étude, il a été amené à développer et proposer un Framework d'évaluation d'outils OSS (*Open Source Software*). Ce Framework s'appuie sur un ensemble de cinq critères, chacun d'entre eux composé de différents indicateurs. Les voici résumés :

- Activité du projet :
 - nombre de versions ;
 - délai moyen entre deux sorties d'une nouvelle version ;
 - nombre de membres du projet ;
 - date de création du projet ;
 - date de la dernière mise à jour.
- Évaluation des utilisateurs :
 - nombre de téléchargements ;
 - évaluations des utilisateurs ;
 - nombre d'évaluations.

- Activité du forum :
 - nombre de messages ;
 - nombre de sujets ;
 - nombre d'utilisateurs enregistrés ;
- Nature du projet :
 - type d'outil (plug-in Eclipse, plate-forme web ou plateforme client riche (*Rich Client Application*)).
- Documentation disponible :
 - API ;
 - tutoriels ;
 - guide de l'utilisateur.

Il a ensuite répertorié les projets open source existants dans les dépôts suivants : *Sourceforge*, *GitHub*, *Google code*, *Codeplex*, *Javaforge* et *Gitorious*. Sur les 107 projets trouvés, certains ont été éliminés immédiatement car ils ne correspondaient pas aux quatre prérequis, n'étaient pas actifs ou ne disposaient pas de forum de discussion. Il ne restait plus alors que six projets : *Bonita Open Solution : Open Source BPM*, *jBPM*, *Modelio – Modeling environment (UML)*, *Yaoqiang BPMN Editor*, *Activiti* et *Bpmn2 Modeler*. Ces six projets ont ensuite été évalués à l'aide du Framework, et le choix final s'est porté sur *Bpmn2 Modeler*.

Nous avons choisi de suivre cette préconisation, guidés par les quatre prérequis et les cinq critères retenus pour composer le Framework d'évaluation que nous jugeons tout à fait pertinents. En détail, nous rappelons que nous avons pour cahier des charges que l'extension de BPMN soit réalisée sur un outil open source, gratuit et portable sur tous les OS. Ceci afin que d'autres contributeurs et utilisateurs puissent utiliser librement ce travail et le poursuivre ou s'en inspirer. De plus, cet outil se devait d'être efficace, accessible par un utilisateur non expert et fournir toutes les fonctionnalités de modélisation du standard BPMN 2.0. Le fait qu'il soit développé en Java est un atout supplémentaire du fait de la toujours importante popularité de ce langage de programmation. Les cinq critères du Framework correspondent aussi à nos souhaits, car ils assurent qu'il s'agit d'un projet bien vivant (au sens de la vivacité des communautés de développeurs libres du projet), actuel, documenté (par le guide de l'utilisateur et par le forum d'entraide) et estimé de ses utilisateurs.

4.2 PRESENTATION DE BPMN2 MODELER

BPMN2 Modeler est un outil open source soutenu par Red Hat (une des organisations dédiées aux logiciels open source les plus importantes et les plus reconnues) permettant de créer et d'éditer des modèles BPMN respectant les standards OMG BPMN 2.0. Mais il permet aussi de créer des extensions du méta-modèle.

BPMN2 Modeler est un plug-in de la plateforme de développement Eclipse. Il se base sur les plug-ins Eclipse Graphiti et EMF (*Eclipse Modeling Framework*) que nous allons présenter dans la suite.

4.2.1 EDI ECLIPSE

Eclipse (www.eclipse.org) est un projet open source qui se décompose en de nombreux sous-projets de développement logiciel. Il est développé par la Fondation Eclipse, organisation à but non lucratif rassemblant de nombreuses entreprises (entre autres IBM, Red Hat, Oracle, etc.) et une communauté de participants répartis à travers le monde. Son EDI (environnement de développement intégré) supporte la quasi-totalité des langages de programmation. Pour plus de commodité, dans la suite du document nous entendrons par Eclipse l'EDI et non la totalité du projet.

Il est entièrement composé de plug-ins (ou greffons) ce qui lui donne une grande souplesse quant à son extension ; chacun l'adapte en fonction de ses besoins en installant les plug-ins adéquats. De plus, il fournit la possibilité d'en développer de nouveaux par le biais du PDE (*Plug-in Development Environment*), ce qui facilite grandement son extension.

Il fournit un environnement simplifiant la programmation. Son interface contient plusieurs perspectives offrant un ensemble de fonctionnalités et de ressources différentes selon la nature du projet (ex : la perspective Java). Chaque perspective regroupe des vues (pour visualiser et sélectionner des éléments) et des éditeurs (pour visualiser et modifier ces éléments). Au sein d'une perspective, le développeur est libre d'afficher ou pas ces vues et éditeurs selon ses besoins. Nous pouvons voir dans la Figure 45 l'interface de développement d'un projet de plug-in.

Son importante communauté, et notamment son forum d'entraide, contribue aussi grandement à sa popularité.

4.2.2 EMF

EMF (*Eclipse Modeling Framework*) est un Framework de modélisation et de génération de code pour développer des applications basées sur un modèle de données structuré. Le modèle peut être soit défini graphiquement (diagramme de classes), soit par le biais d'un modèle *ecore*, soit enfin directement écrit en langage XMI. Son générateur de code transforme ensuite le modèle en code, créant les classes directement dans le code du projet.

Il existe un modèle *ecore* correspondant au méta-modèle BPMN 2.0. Il suffit donc au développeur qui souhaite étendre ce méta-modèle de créer la description de son extension sans avoir à réécrire l'ensemble du méta-modèle.

4.2.3 GRAPHITI

Le Framework graphique Graphiti permet le développement d'éditeurs de diagrammes ad-hoc pour la réalisation de diagrammes de modèles.

Le principe est le suivant : Graphiti fournit des fonctions de création d'éléments graphiques correspondant aux différents objets du modèle à représenter. Il faut ensuite lier chacune des représentations à l'objet auquel elle correspond. On aura alors créé un éditeur de modèles, dans

lequel le code généré par EMF sera utilisé pour instancier les objets du domaine de modèle qui pourront ainsi être représentés.

L'architecture de base de Graphiti est visible dans la Figure 42 :

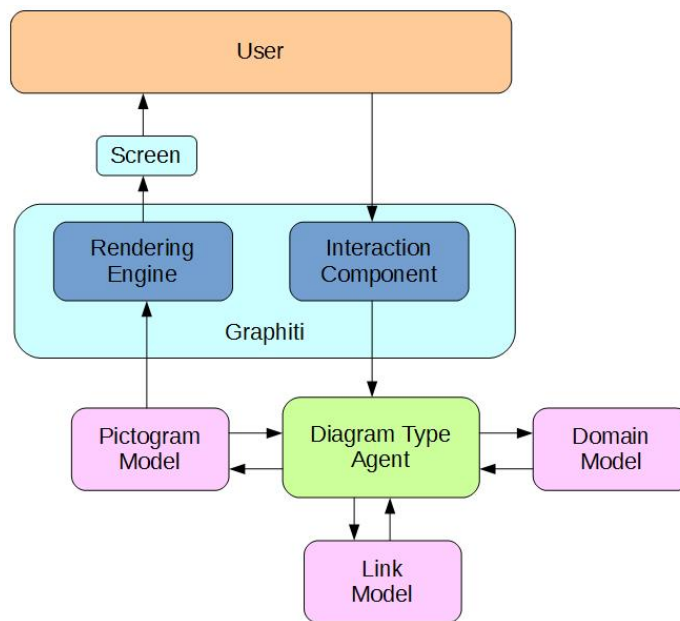


Figure 42 : Architecture de base de Graphiti

On peut y voir que Graphiti sert d'interface entre l'utilisateur et les éléments du modèle représenté. Un *Interaction Component* reçoit les requêtes de l'utilisateur (par exemple cliquer-glisser un élément vers le diagramme) et les transmet au *Diagram Type Agent*, qui est implémenté par le développeur. Celui-ci a pour rôle de :

- Créer les objets du modèle de domaine (*Domain Model*).
- Créer la représentation graphique qui va représenter l'objet créé (*Pictogram Model*).
- Lier l'objet et sa représentation graphique (*Link Model*).

Enfin, un moteur de rendu (*Rendering Engine*) affichera le graphique à l'écran, afin qu'il soit visualisé par l'utilisateur.

La programmation du *Diagram Type Agent* et la définition du modèle de domaine sont les éléments clés car ce sont eux qui vont conditionner le comportement du logiciel final et donc de le faire correspondre au domaine que l'on souhaite représenter.

4.2.4 BPMN 2 MODELER

BPMN 2 Modeler est un éditeur graphique de diagrammes BPMN respectueux des standards OMG BPMN 2.0. Il permet aussi de visualiser des diagrammes BPMN réalisés avec d'autres outils, à la condition qu'ils respectent ces mêmes standards. Son interface intuitive est composée de cinq vues et éditeurs qui sont décrits dans Figure 43.

1. Vue des dossiers : permet de naviguer dans les projets et de sélectionner les modèles.
2. Vue de la structure : nous offre une vue des éléments du diagramme sous forme d'arborescence. Cette vue est synchronisée avec l'éditeur de diagramme : quand on clique sur un élément de l'éditeur, ce même élément est surligné dans cette vue, et inversement.
3. Éditeur de diagramme : c'est ici que l'on place les éléments du diagramme en les faisant glisser depuis la palette d'outils. Les différents diagrammes sont accessibles dans des onglets.
4. Vue des propriétés : on peut y visualiser et modifier les propriétés des éléments du diagramme. A noter que ces mêmes opérations peuvent être effectuées dans une fenêtre séparée qui apparaît lorsque l'on effectue un double clic sur l'élément graphique.
5. Palette d'outils : c'est d'ici que l'on fait glisser les éléments graphiques vers le diagramme.

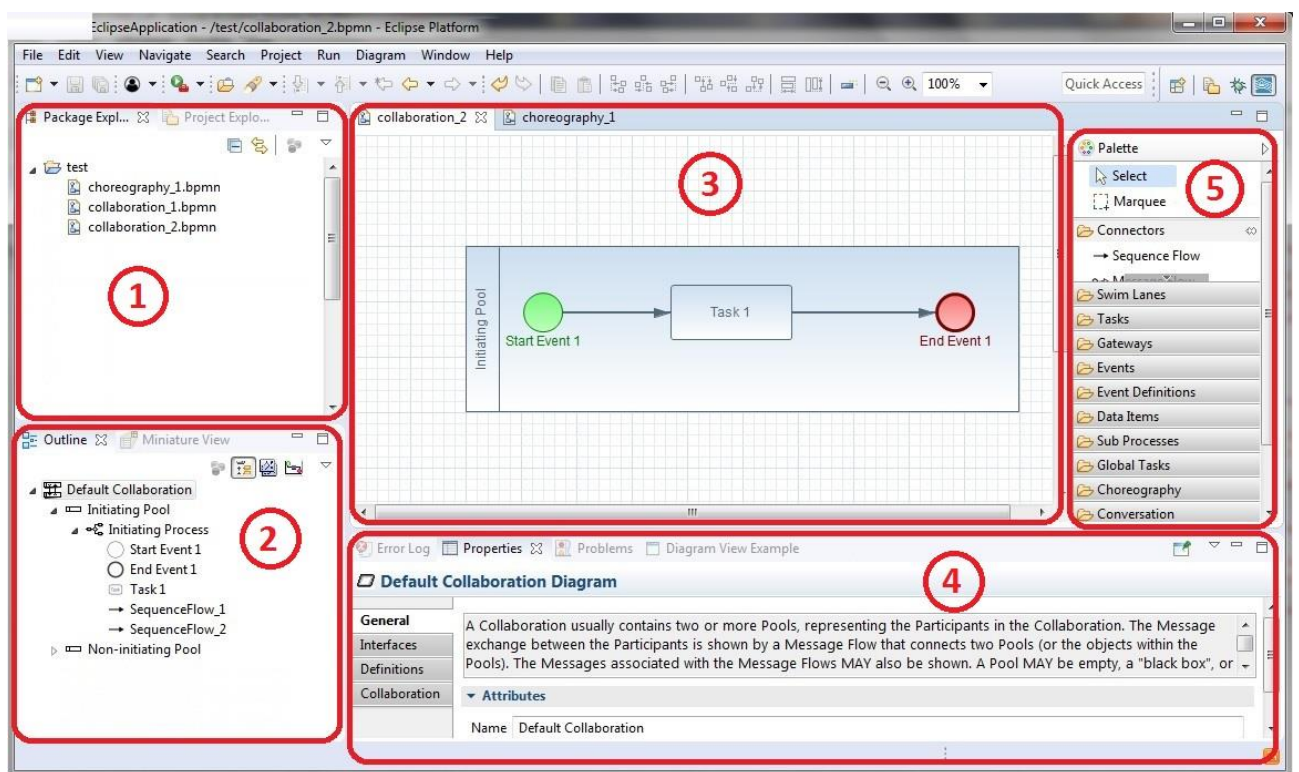


Figure 43 : Interface de création de modèles BPMN de BPMN 2 Modeler

L'éditeur nous offre aussi la possibilité de visualiser le code source du diagramme en langage XML comme présenté dans la Figure 44.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- origin at X=0.0 Y=0.0 -->
<bpnm2:definitions xmlns:t1="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:bpnm2="http://www.omg.org/s
  <bpnm2:collaboration id="ColLaboration_1" name="Default Collaboration">
    <bpnm2:participant id="Participant_1" name="Initiating Pool" processRef="Process_1"/>
    <bpnm2:participant id="Participant_2" name="Non-initiating Pool" processRef="Process_2"/>
  </bpnm2:collaboration>
  <bpnm2:process id="Process_1" name="Initiating Process" definitionalCollaborationRef="Collaboration_1" isExecutable="false">
    <bpnm2:startEvent id="StartEvent_1" name="Start Event 1">
      <bpnm2:outgoing>SequenceFlow_1</bpnm2:outgoing>
    </bpnm2:startEvent>
    <bpnm2:endEvent id="EndEvent_1" name="End Event 1">
      <bpnm2:incoming>SequenceFlow_2</bpnm2:incoming>
    </bpnm2:endEvent>
    <bpnm2:task id="Task_1" name="Task 1">
      <bpnm2:incoming>SequenceFlow_1</bpnm2:incoming>
      <bpnm2:outgoing>SequenceFlow_2</bpnm2:outgoing>
    </bpnm2:task>
    <bpnm2:sequenceFlow id="SequenceFlow_1" sourceRef="StartEvent_1" targetRef="Task_1"/>
    <bpnm2:sequenceFlow id="SequenceFlow_2" sourceRef="Task_1" targetRef="EndEvent_1"/>
  </bpnm2:process>
  <bpnm2:process id="Process_2" name="Non-initiating Process" definitionalCollaborationRef="Collaboration_1" isExecutable="false"/>
</bpnm2:definitions>
<bpnm2:BPMDiagram id="BPMDiagram_1" name="Default Collaboration Diagram">
```

Figure 44 : Code source XML du diagramme BPMN

L'autre fonctionnalité importante de BPMN 2 Modeler est la possibilité d'étendre le méta-modèle de BPMN, ce qui a suscité la création de ce projet.

Les extensions sont codées en Java dans l'interface du *Plug-in Development Environment* présenté ci-après et dans la Figure 45 :

1. Explorateur de projets : on peut y sélectionner les projets et ouvrir les fichiers contenant le code.
2. Éditeur de code : c'est ici que s'affiche et que l'on peut modifier le code. Les différents fichiers ouverts sont accessibles dans des onglets
3. Vue console : où s'affichent les sorties du programme. D'autres onglets permettent d'afficher les problèmes d'exécution, les logs d'erreur, les propriétés de la sélection courante, etc.
4. Vue d'une classe : affiche sous forme d'arborescence les différentes classes et méthodes du fichier affiché. Cette vue est synchronisée avec l'éditeur de code : quand on clique dans une classe ou méthode dans l'éditeur, ce même élément est surligné dans cette vue, et inversement.

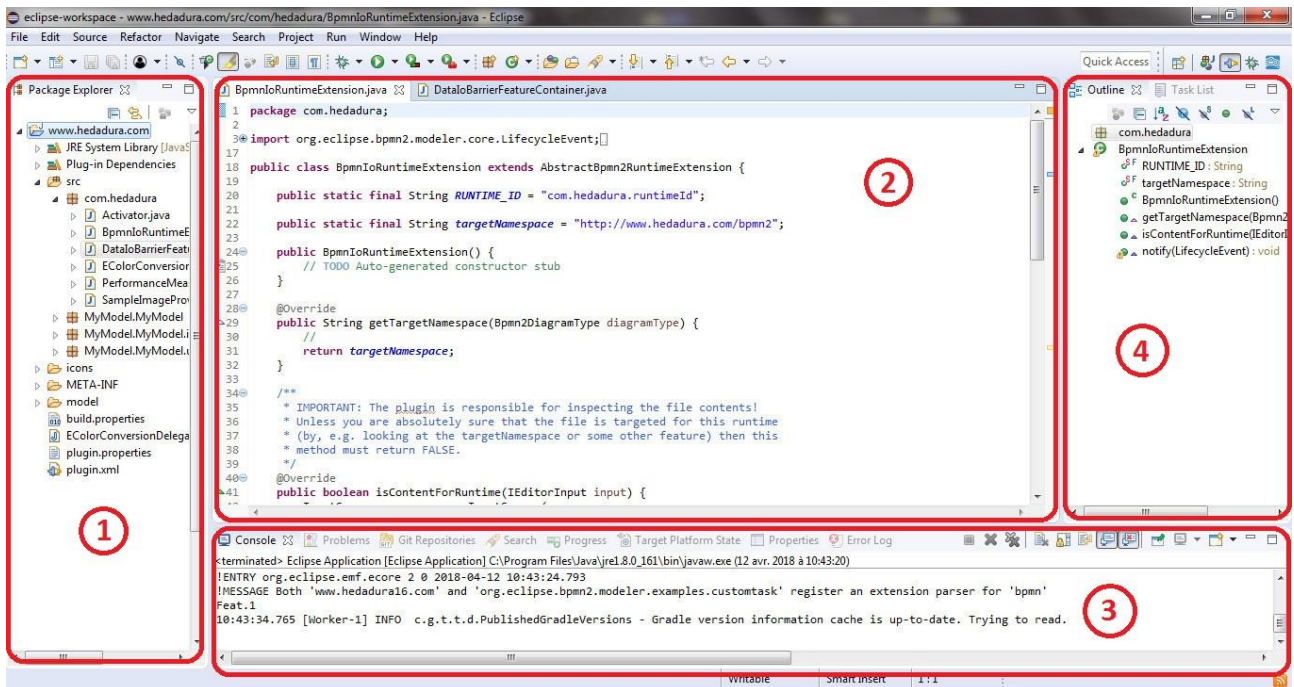


Figure 45 : Interface de développement des plug-ins

Nous aborderons d'autres éléments de ce plug-in dans le paragraphe suivant.

4.3 IMPLEMENTATION DES EXTENSIONS

4.3.1 SPECIFICATIONS

Dans le chapitre 4.2.2, nous avons présenté les extensions que nous souhaitons créer. Nous devons à présent en établir les spécifications.

4.3.1.1 Extensions dataInteroperabilityBarrier et dataInteroperabilityResolved

Le but de ces extensions est de représenter une donnée échangée présentant un problème d'interopérabilité ou bien pour laquelle ce problème a été résolu. L'unique différence avec la représentation d'un Data Object est l'apparence visuelle, et aucun attribut supplémentaire n'est défini.

4.3.1.2 Extension performanceMeasurement

Cette extension a pour but d'afficher les mesures de la performance des quatre dimensions définies (coût, délai, qualité et fiabilité). On va donc lui définir un attribut pour chacune des dimensions et créer un onglet spécifique dans sa fenêtre de propriétés permettant de modifier leur valeur. Il faudra ensuite créer les zones d'affichage de ces valeurs.

4.3.1.3 Extension performanceAssociation

Afin de restreindre la possibilité d'association de l'extension performanceMeasurement aux seules tâches et sous-processus, nous devons étendre le connecteur de type *Association*. En effet, on définit au niveau des connecteurs les objets qui peuvent être reliés entre eux. Cette association

spécifique ne devra permettre que les liaisons entre la source qu'est *performanceMeasurement* et deux différentes cibles que sont les tâches et les sous-processus.

4.3.2 FONCTIONNEMENT DE BPMN 2 MODELER

Comme nous l'avons rappelé précédemment, BPMN 2 Modeler se fonde sur EMF pour définir les objets à représenter. Il existe donc un modèle *ecore* de BPMN 2.0 dans lequel sont décrits tous les éléments de BPMN. Pour la réalisation de l'éditeur graphique, c'est le plug-in Graphiti qui lui sert de base. Des classes de représentation et de liaison des éléments graphiques avec ceux du modèle ont donc été écrites.

En ce qui concerne le mécanisme d'extension, le fichier *plugin.xml* présent dans tout projet de plug-in permet de définir les différentes extensions qui seront ajoutées. On y définit un certain nombre de propriétés, en particulier la classe java qui sera utilisée pour implémenter cette extension. Le terme extension est ici plus vaste que celui employé dans ce qui a précédé et qui signifiait « nouvel élément de BPMN ». Il s'agit des extensions de l'éditeur BPMN, et différents éléments peuvent être étendus : le modèle de domaine, l'apparence du nouvel élément, la fenêtre de propriété de cet élément, etc.

Un élément en particulier doit être étendu : le « Target Runtime » ou environnement d'exécution cible. Cet élément encapsule les extensions du modèle et fournit un environnement de représentation (formes graphiques, fenêtres de propriétés, etc.) pour le moteur d'exécution spécifique de BPMN2.

Des classes d'implémentation spécifiques ont été écrites pour servir d'interface avec les classes des éléments de BPMN2. Plutôt que d'étendre la classe correspondant à l'élément que l'on doit étendre, on utilise ces classes. Par exemple, la classe de représentation de forme graphique *CustomShapeFeatureContainer* doit être étendue pour créer la classe de représentation des éléments de BPMN2.

4.3.3 IMPLEMENTATION

4.3.3.1 Modèle

Nous avons tout d'abord créé le modèle de nos extensions. Comme nous l'avons indiqué plus avant, il ne faut représenter dans ce modèle que les extensions créées. Nous avons donc représenté les éléments *dataInteroperabilityBarrier*, *dataInteroperabilityResolved* et *performanceMeasurement*. La Figure 46 présente le diagramme de classe de ces éléments.

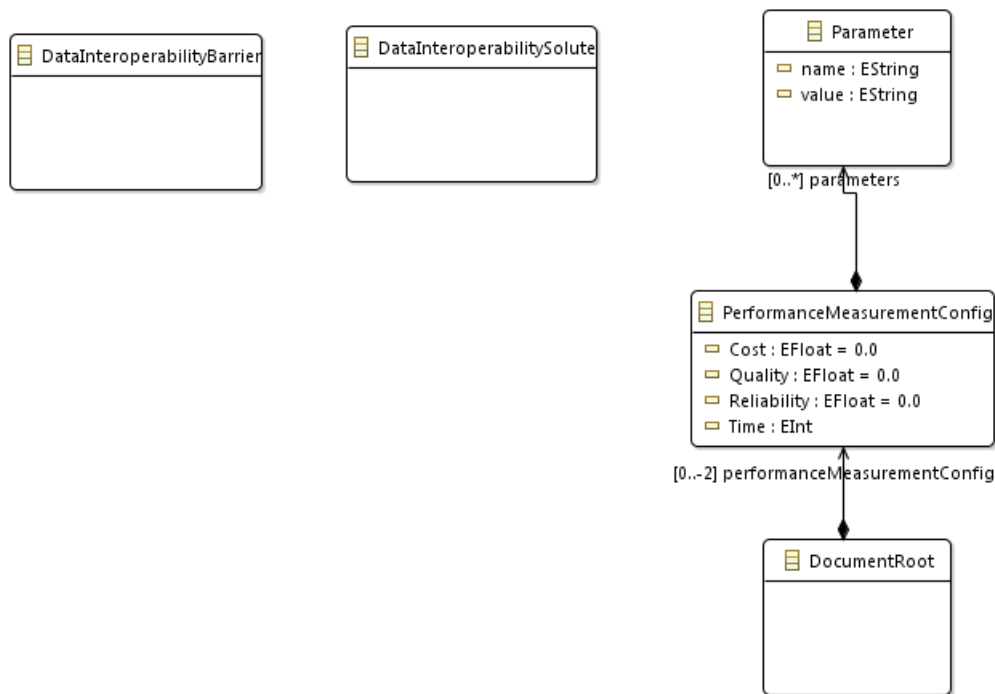


Figure 46 : Classes du modèle ecore

A noter que l'extension *performanceAssociation* ne figure pas sur ce modèle. En effet, si une extension est juste une « décoration » d'un élément existant, il n'est pas nécessaire de la faire apparaître dans le modèle. Les seules modifications que nous voulons apporter à l'élément *Association* étant son aspect et la restriction des éléments qu'elle peut relier, il n'est pas utile de la modéliser.

4.3.3.2 Target Runtime

Nous avons vu qu'il fallait créer un environnement d'exécution pour notre plug-in. Celui-ci est défini dans le fichier *plugin.xml* à l'aide d'un outil graphique.

4.3.3.3 dataInteroperabilityBarrier et dataInteroperabilityResolved

Nous avons défini nos deux extensions sur le même modèle dans le fichier *plugin.xml*. Ceci se fait en définissant un *customTask* dans lequel on renseigne un certain nombre de paramètres. La dénomination *customTask* est trompeuse car elle pourrait faire croire que cette section ne sert qu'à l'extension de l'élément *Task*, alors qu'elle doit être utilisée pour des extensions d'autres types, tels *TextAnnotation*, *EventDefinition*, *Gateway*, etc. Dans notre cas, nous avons étendu l'élément *DataObject*.

Nous avons ensuite étendu la classe *CustomShapeFeatureContainer*. Cette classe est celle qui doit être étendue par toute classe qui permet la représentation graphique des éléments BPMN2.

Nos deux extensions ne possédant pas d'attributs spécifiques, nous n'avons pas configuré de fenêtre de propriétés spécifiques.

Nous avons créé les icônes identifiant les extensions *dataInteroperabilityBarrier* et *dataInteroperabilityResolved*. Ces icônes apparaissent dans la palette graphique afin de permettre de faire glisser les extensions dans les diagrammes. Elles sont aussi incorporées dans la représentation de ces extensions. L'icône de *dataInteroperabilityBarrier* est présentée dans la Figure 47.



Figure 47 : Icône de *dataInteroperabilityBarrier*

L'icône de *dataInteroperabilityResolved* est présentée dans la Figure 48.



Figure 48 : Icône de *dataInteroperabilityResolved*

4.3.3.4 performanceMeasurement

Pour cette extension, nous avons étendu l'élément BPMN2 *TextAnnotation*.

Dans ce but nous avons créé un *customTask* au sein duquel nous avons déclaré les attributs *cost*, *quality*, *availability* et *time*.

Là aussi nous avons étendu la classe *CustomShapeFeatureContainer* dans la classe *performanceMeasurementFeatureContainer* pour représenter notre extension.

Nous avons défini un nouvel onglet dans les propriétés de cet élément afin de pouvoir renseigner les valeurs des différentes mesures de la performance.

Nous avons dû adapter le cas du rafraîchissement de l'affichage lorsqu'on modifie la valeur d'un ou plusieurs attributs, car ce n'est pas automatique.

Il nous a fallu distinguer trois situations :

- Lorsque l'extension est reliée à une tâche ne faisant pas partie d'un sous-processus (cas du TO BE) : dans ce cas, lorsqu'on modifie la valeur des attributs, cette valeur doit être mise à jour dans sa représentation graphique.
- Lorsque l'extension est reliée à une tâche faisant partie d'un sous-processus (cas du AS IS) : dans ce cas, lorsqu'on modifie la valeur des attributs, cette valeur doit être mise à jour dans sa représentation graphique et envoyée vers la représentation de celle qui est reliée au sous-processus pour qu'elle aussi soit mise à jour.

- Lorsque l'extension est reliée à un sous-processus : la valeur affichée doit être calculée sur la base des mesures de performance des tâches prenant part au sous-processus.

Nous avons aussi créé l'icône qui sera affichée dans la palette d'outils. Elle est représentée dans la Figure 49.

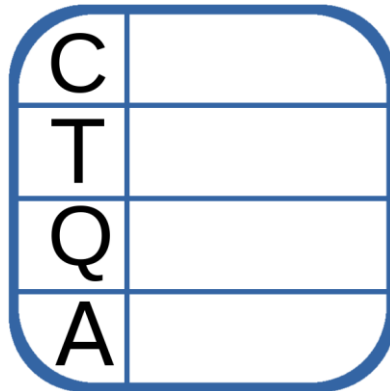


Figure 49 : Icône de performanceMeasurement

4.3.3.5 performanceAssociation

Les extensions des connecteurs se font en étendant la classe *CustomConnectionFeatureContainer*. En effet, les représentations graphiques des connecteurs sont radicalement différentes de celles de tout autre élément de BPMN2. De plus, elles ont la caractéristique de relier des éléments entre eux, ce qui induit des méthodes et des propriétés uniques, telles les restrictions quant aux éléments qu'elles doivent relier.

Nous avons fait le choix de n'accepter comme source de connexion que les éléments *performanceMeasurement* et comme cible que les éléments *Activity* (tâches) et *SubProcess*.

4.3.4 INTERFACE ETENDUE

L'interface de BPMN2 Modeler étendue de nos fonctionnalités est présentée dans la Figure 50. Les numérotations permettent de faire le lien avec la Figure 41 conceptuelle présentée dans le chapitre précédent.

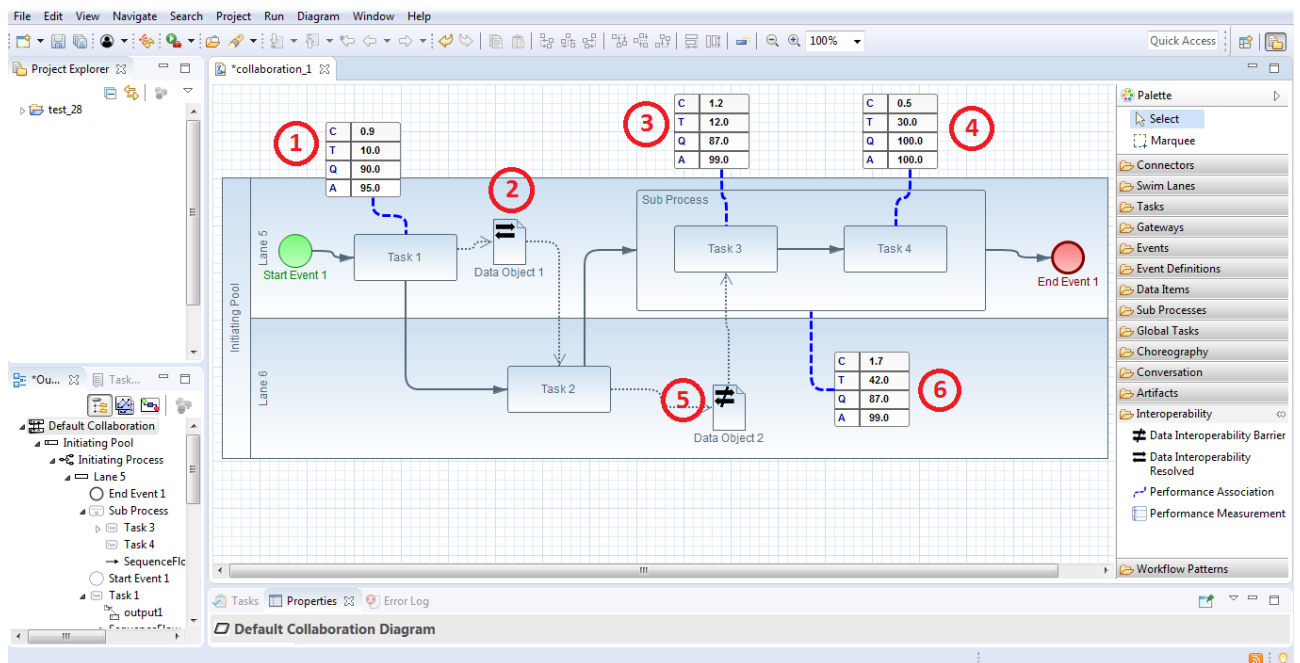


Figure 50 : Interface étendue de BPMN Modeler

L'interface basée Eclipse retrouve les fonctionnalités classiques de BPMN2 Modeler. La différence réside dans la présence dans la palette d'outils de la catégorie « *Interoperability* » dans laquelle on retrouve les extensions que nous avons développées. Un zoom sur cette catégorie est visible dans la Figure 51.

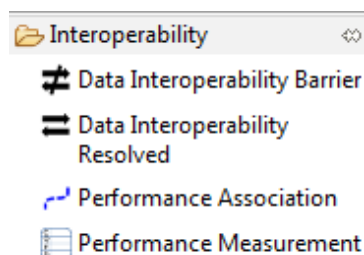


Figure 51 : Catégorie "Interoperability"

Ce menu propose quatre lignes correspondant aux quatre extensions que nous avons implémentées. Sur chacune d'entre elles est présentée l'icône et le nom d'une de ces extensions.

C'est à partir de cette palette que l'utilisateur fera glisser les extensions vers les modèles.

Cette interface étendue permet de réaliser les modèles décrits dans le chapitre précédent. Ces modèles nous serviront, en association avec l'AMDEC données, à identifier et accompagner la résolution de problèmes d'interopérabilité. Nous avons réalisé dans la Figure 50 un modèle d'étude pour illustrer l'utilisation des extensions. Il détaille :

- Un échange de donnée pour lequel l'interopérabilité a été établie représenté par l'extension *dataInteroperabilityResolved* (2).

- Un échange de données présentant un problème d'interopérabilité représenté par l'extension *dataInteroperabilityBarrier* (5).
- Des tableaux de performance, associés à des tâches, représentés par l'extension *performanceMeasurement* (1, 3 et 4). Ces tableaux sont reliés à leurs tâches respectives à l'aide de l'extension *performanceAssociation*.
- Un tableau de performance associé à un sous-processus représenté par l'extension *performanceMeasurement* (6) et relié par l'extension *performanceAssociation*.

Les valeurs des mesures de la performance sont renseignées par le biais d'une fenêtre de propriété correspondant à chacun des tableaux affichés. On fait apparaître cette fenêtre en effectuant un double-clic sur le tableau. La fenêtre de propriété est illustrée par la Figure 52.

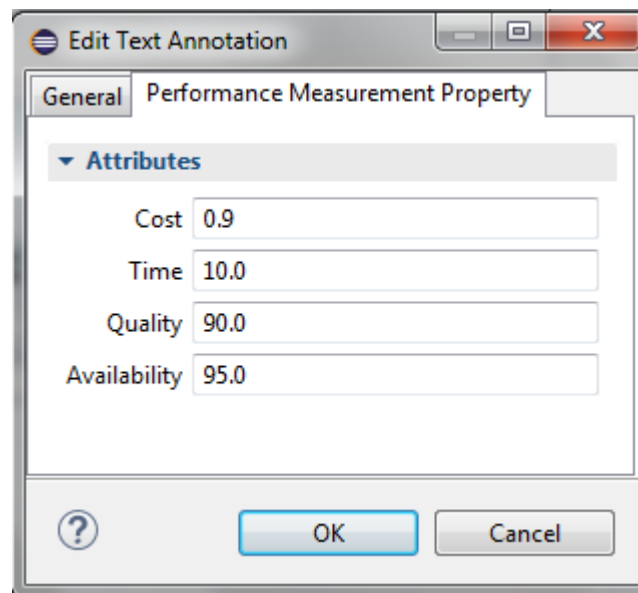


Figure 52 : Fenêtre de propriétés de performanceMeasurement

On renseigne les valeurs de mesures de la performance dans les champs prévus à cet effet.

4.4 CONCLUSION

Dans ce chapitre nous avons décrit l'implémentation des différentes extensions de BPMN 2.0 que nous avons proposées au cours du chapitre 3.

Nous avons commencé par choisir un outil de modélisation BPMN 2.0. Pour cela, nous avons effectué un état de l'art et avons en particulier suivi les recommandations de l'étude réalisée par Marco Roque. Les prérequis d'ergonomie de modélisation pour les utilisateurs, de compatibilité avec le standard BPMN 2.0, de développement en Java, de projet open source efficace et enfin d'évolutivité pour des développements ultérieurs ont guidé le choix vers BPMN2 Modeler.

Après une brève introduction de l'EDI Eclipse, nous avons présenté les plug-ins sur lesquels s'appuie BPMN2 Modeler pour permettre l'implémentation d'extensions : EMF pour définir le

modèle des extensions et générer le code correspondant et Graphiti pour la représentation graphique.

Nous avons enfin succinctement expliqué le fonctionnement de BPMN2 Modeler pour finir par un survol de l'implémentation des extensions que nous avons réalisée : *dataInteroperabilityBarrier*, *dataInteroperabilityResolved*, *performanceMeasurement* et *performanceAssociation*.

Dans le prochain chapitre nous allons relater l'application que nous avons faite des différents outils que nous avons proposés au cours de ce chapitre 3 à travers un cas d'étude réel issue de l'industrie agroalimentaire. Après avoir présenté l'entreprise Onetik, nous parlerons de la problématique à laquelle elle était confrontée, puis présenterons le projet d'établissement de l'interopérabilité que nous y avons réalisé. Nous y verrons les modèles qui y ont été réalisés ainsi que la mise en pratique de la méthode AMDEC données.

Chapitre 5 : CAS D'ETUDE CHEZ ONETIK

<u>5.1 Introduction</u>	111
<u>5.2 Présentation de l'entreprise Onetik</u>	111
<u>5.3 Problématique et solution</u>	112
<u>5.3.1 ERP et fonctionnement</u>	112
<u>5.3.2 Problème d'interopérabilité</u>	112
<u>5.3.3 Solution envisagée</u>	112
<u>5.4 Modèles de résolution</u>	113
<u>5.4.1 Données utilisées</u>	113
<u>5.4.2 Modèle AS IS</u>	116
<u>5.4.3 Modèle TO BE</u>	119
<u>5.4.4 Comparaison des modèles</u>	121
<u>5.4.5 Modèle global</u>	121
<u>5.5 Elaboration de l'AMDEC données</u>	124
<u>5.5.1 Étape 1 : initialisation</u>	124
<u>5.5.2 Étape 2 : analyse fonctionnelle</u>	124
<u>5.5.3 Étape 3 : analyse qualitative et quantitative</u>	126
<u>5.5.4 Étape 4 : synthèse</u>	128
<u>5.5.5 Phase d'amélioration continue</u>	129
<u>5.6 Conclusion</u>	130

5.1 INTRODUCTION

Dans le chapitre 3 nous avons fait des propositions d'outils permettant de progresser dans la résolution des problèmes d'interopérabilité dans les échanges de données entre différents systèmes d'information. Nous avons tout d'abord défini des extensions du langage BPMN 2.0 dans le but de nous donner les moyens de réaliser des modèles qui reflètent la problématique de l'interopérabilité. Nous avons défini les extensions *dataInteroperabilityBarrier* pour illustrer l'absence d'interopérabilité et *dataInteroperabilityResolved* pour représenter les échanges de données pour lesquels ce problème a été résolu. Nous avons ensuite déterminé quels étaient les domaines de performance pertinents pour quantifier les conséquences de l'absence d'interopérabilité et avons défini l'extension *performanceMeasurement* dont le but est d'afficher dans les modèles BPMN 2.0 ces mesures de performance. Nous avons enfin élaboré une méthode de gestion de la qualité des données. Nous avons nommé cette méthode AMDEC données.

Au cours du chapitre 4, nous avons présenté l'implémentation de ces extensions que nous avons réalisées à l'aide de l'outil BPMN2 Modeler. Cette implémentation permet la réalisation de tels modèles.

Dans ce chapitre, nous allons illustrer, par un cas d'étude réel, l'utilisation des extensions de BPMN 2.0 que nous avons proposées et implémentées. Dans un premier temps nous allons présenter l'entreprise Onetik au sein de laquelle nous avons étrenné les différents outils que nous avons proposés.

Ce chapitre se décompose en trois parties. Tout d'abord nous allons présenter l'entreprise et son contexte commercial. Nous exposerons ensuite la problématique à laquelle elle était confrontée et la solution qui a été envisagée pour la résoudre. Nous présenterons les modèles que nous avons produits afin de résoudre la problématique de l'interopérabilité dans la mise en place de la solution. Nous terminerons par décrire la mise en place de la méthode AMDEC données dans l'optique de continuer à améliorer la qualité des données.

5.2 PRESENTATION DE L'ENTREPRISE ONETIK

Onetik est une PME du Pays Basque qui fabrique et commercialise des fromages. Elle emploie environ 85 salariés pour un chiffre d'affaire de près de 35 millions d'euros en 2016.

En 1982, la coopérative Berria est créée de l'union de trois unités de transformation de lait (la SICA Esnea, Baiguramendi et Baigorry) dans le but d'optimiser la collecte des laits. L'année suivante, la fromagerie Onetik est créée. Elle collecte la production laitière de plus de 150 éleveurs de brebis et 200 producteurs de lait de vache. En 2003, elle implante en son sein l'ERP Nodhos afin d'optimiser sa productivité dans tous les domaines.

La production représente un volume avoisinant les 3000 tonnes de fromage par année pour lesquels elle doit trouver des débouchés commerciaux.

On peut classer ses clients en trois catégories : les « petits » clients (magasins, restaurants, etc.), les grossistes et la grande distribution.

5.3 PROBLEMATIQUE ET SOLUTION

5.3.1 ERP ET FONCTIONNEMENT

Onetik utilise l'ERP Nodhos qui couvre la totalité du système d'information de l'entreprise. Toutefois, le module de gestion des expéditions de cet ERP ne lui donne pas entière satisfaction. En effet, la version de cet ERP utilisée chez Onetik est assez ancienne et la gestion des expéditions y est relativement sommaire et plutôt adaptée aux articles vendus à la pièce qu'à ceux vendus au poids (ce qui est le cas des fromages). Nous illustrons par ce cas une des situations de non interopérabilité rencontrée par l'entreprise. A titre d'étude, nous en présentons ci-après un descriptif de fonctionnement : une fois la commande saisie et confirmée, un Bon de Préparation (BP) est généré. Après la préparation des colis et leur pesée, le poids et le numéro de lot des fromages doivent y être renseignés. Ensuite, ce BP servira à générer une facture qui sera expédiée au client. Ce fonctionnement est représenté par le modèle AS IS illustré par la Figure 53.

5.3.2 PROBLEME D'INTEROPERABILITE

Actuellement, il n'existe aucun interfaçage avec les balances sur lesquelles sont pesés les colis. Le BP est donc imprimé puis transmis physiquement aux opérateurs de pesée. Ceux-ci doivent saisir sur la balance un ensemble de données : identifiant produit, identifiant client, date limite d'utilisation optimale (DLUO), tare, nombre de colis de l'article concerné et numéro de lot. A noter que la plupart de ces informations ne sont pas présentes sur le BP et qu'il faut donc les récupérer depuis une autre source (affichage, cahier ou de manière orale). Une fois la pesée effectuée, l'opérateur de pesée note sur le BP le poids et le numéro de lot avant que le document ne soit transmis à la personne qui saisira ces informations dans l'ERP Nodhos.

Cette absence d'interfaçage représente un problème d'interopérabilité puisqu'il impose des tâches de ressaisie des données avant que celles-ci ne puissent être utilisées dans la suite du processus. Ces tâches de ressaisie sont des tâches à non-valeur ajoutée qui représentent une importante source d'erreurs, certaines données étant même ressaisies plusieurs fois (poids et numéro de lot). Les erreurs les plus pénalisantes sont les erreurs de poids (qui se transforment en erreurs de prix), les erreurs de DLUO et les erreurs de clients (une commande étant expédiée vers un client qui ne l'avait pas commandée). Les conséquences en sont une insatisfaction des clients ainsi que de lourdes pénalités financières. En effet, la grande distribution est impitoyable quant à ce type d'erreurs, et elle n'hésite pas à appliquer des sanctions financières forfaitaires même dans le cas d'erreurs minimales (et même dans le cas où une erreur de poids est en leur faveur).

5.3.3 SOLUTION ENVISAGEE

Toutefois, le remplacement de Nodhos n'est pas envisagé à court terme. En effet, le remplacement d'un ERP est un projet important et lourd à mettre en place. Il concerne la totalité de l'entreprise, et Nodhos n'est pas remis en cause pour des secteurs autres que celui des expéditions. De plus, il s'agit d'un chantier dont la durée peut se mesurer en années pour une PME qui n'a pas les finances pour s'offrir les services d'un intégrateur de premier plan, et l'amélioration

des expéditions présentait un caractère d'urgence. Enfin, le coût financier d'un tel projet est loin d'être négligeable pour une PME, et les bénéfices qui vont en découler ne seront visibles qu'à moyen terme.

La direction d'Onetik a donc décidé de greffer le module de gestion des expéditions d'un autre ERP à celui déjà en place. Nous ne relaterons pas ici la procédure qui a été mise en place pour déterminer le choix du nouveau progiciel car elle nous éloignerait trop de notre sujet. L'ERP sélectionné, Integraal Agro, a été développé à destination des entreprises agro-alimentaires, et son module de gestion est particulièrement adapté aux pratiques qui ont cours à Onetik, notamment son interfaçage avec les balances de pesée. Ne restait donc qu'à établir la jonction entre Nodhos et le module de gestion des expéditions d'Integraal Agro.

Nous avons naturellement utilisé les outils que nous proposons dans le cadre de cette thèse pour résoudre les problèmes d'interopérabilité.

5.4 MODELES DE RESOLUTION

Dans le cadre de ce projet, deux types de modélisations ont été réalisés à l'aide de BPMN 2.0 : l'un correspondant au système existant (AS IS) et l'autre au système souhaité (TO BE). Bien évidemment, la mise en service du nouveau module de gestion des expéditions a nécessité la remise en cause du fonctionnement de plusieurs processus de ce service, entraînant autant de modélisations de chacun des deux types.

Les fromages sont conditionnés de différentes manières : entiers dans des cartons pesés en vrac, entiers mais pesés individuellement, en bacs, en portions, etc. Afin de ne pas alourdir notre démonstration, nous ne représenterons ici que le processus simplifié de préparation des commandes de fromages entiers dans des cartons pesés en vrac.

5.4.1 DONNEES UTILISEES

Les chiffres des ventes sont un sujet sensible sur lequel la direction d'Onetik ne souhaite pas communiquer. Les chiffres que nous utiliserons dans ce cas d'étude ont par conséquent été modifiés tout en gardant un caractère réaliste.

Nous avons choisi de réaliser des modèles représentant une journée de travail du service des expéditions. Tous les chiffres qui apparaissent dans les modèles que nous présentons dans la suite ont donc été rapportés à cette période et à l'ensemble du service des expéditions.

Afin de remplir les tableaux de mesure de la performance, il nous fallait estimer les valeurs pour chacune des dimensions que nous avons définies, c'est-à-dire le coût, le délai, la qualité et la disponibilité.

Dans le cas des modèles TO BE, la tâche a été plus aisée car il s'agit du système souhaité. Or, dans un système « idéal » les problèmes d'interopérabilité ayant été éliminés, la qualité et la disponibilité ne sont plus à calculer mais correspondent à l'objectif que l'entreprise se fixe. Ne restent alors à apprécier que les valeurs du coût et du délai pour les seules tâches à valeur ajoutée.

Par contre dans le cas de la modélisation de l'existant (AS IS) la tâche s'est avérée plus complexe comme nous allons le voir dans la suite.

5.4.1.1 Estimation de la qualité et de la disponibilité

Comme nous venons de le voir, l'estimation de la qualité et de la disponibilité ne concerne que les modèles de type AS IS.

La première difficulté à laquelle nous ayons été confrontés a été la collecte de chiffres au niveau de la qualité et de la disponibilité concernant les tâches impactées par des échanges de données présentant un problème d'interopérabilité. En effet, ces données n'étaient pas systématiquement répertoriées. La raison principale en est que les erreurs entraînées par les problèmes d'interopérabilité sont pour la plupart des erreurs commises lors de la ressaisie des données. Or, ces erreurs sont bien évidemment involontaires et donc difficiles, voire impossible pour certaines, à déceler. Un contrôle a posteriori aurait permis d'en repérer une partie, par exemple les erreurs dans la notation du poids si les colis avaient été repesés avant le départ de la marchandise. Mais les contraintes de l'expédition, notamment l'horaire de départ des camions, ne permettent pas un tel contrôle. Dans d'autres cas, déceler de telles erreurs s'avère impossible. Comment savoir si un fromage correspond à tel ou tel lot quand, dans la même journée, plusieurs lots ont été utilisés pour ce même produit ?

Afin d'obtenir des données significatives il nous a donc fallu utiliser les données du service après-ventes. Or, les données que recense ce service ne correspondent qu'aux doléances des clients. Mais tous les types de clients ne contrôlent pas de la même manière l'exactitude des données présentes sur les étiquettes, sur les bons de livraisons ou sur les factures. Par exemple, il est rare qu'un « petit » client repese les colis de fromage qui lui sont livrés. Ce type de client ne pourra donc déceler que des erreurs grossières, telle par exemple une adresse de livraison erronée qui entraîne une livraison hors délais. Par contre, les usages de la grande distribution font que la plupart des erreurs commises sont détectées et signalées (du fait des pénalités qu'ils appliquent aux fournisseurs). Les colis sont repesés, la concordance des numéros de lot entre le bon de livraison et les étiquettes sur les colis vérifiés, etc.

Pour obtenir des données correspondant le plus possible à la réalité, nous avons donc choisi de ne prendre en compte que les réclamations des clients de la grande distribution et de reporter les chiffres obtenus à la totalité des expéditions effectuées. Il a aussi fallu écarter les réclamations non imputables aux problèmes d'interopérabilité (problèmes de qualité du produit, problèmes de livraison dus au transporteur, etc.).

Une autre difficulté a été de différencier l'échange de donnée en cause dans les cas où une donnée était ressaisie deux fois au cours du même processus. Nous avons choisi de diviser par deux les valeurs recensées et d'affecter la valeur obtenue à chacune des tâches de ressaisie intervenant dans le processus et concernant l'échange de cette donnée.

Enfin, la cause exacte de la non-qualité de la donnée incriminée n'est pas toujours claire dans les archives du service après-ventes et on ne peut pas forcément savoir si elle est due à un manque de disponibilité. Dans d'autres cas, la non-disponibilité de la donnée a été compensée par une

recherche à la source et n'a pas entraîné une réclamation mais une grande perte de temps en amont. En conséquence, pour la valeur de la disponibilité nous nous sommes appuyés sur les valeurs estimées dans le cadre de l'AMDEC données.

Nous avons ainsi pu faire une estimation de la qualité et de la disponibilité des données pour les modèles représentant le système existant. En ce qui concerne le système futur, comme nous l'avons déjà dit, il ne s'agit pas d'une estimation mais d'un objectif.

Voici la méthode que nous avons suivie pour estimer la valeur globale de la qualité et de la disponibilité pour une journée de travail : si, pour un échange de données, au moins une donnée présente un problème de qualité, la valeur de la qualité est de 0. S'il n'y a aucun défaut de qualité, elle est de 1. Pour calculer le taux de la journée, nous calculons le pourcentage d'échanges de données de qualité par rapport au nombre total d'échanges de données (donc de commandes préparées). De même pour la disponibilité.

5.4.1.2 Estimation du coût

Nous avons vu au cours du chapitre 2 qu'il existait différents types de coûts. Dans le cadre de cette étude de cas nous en avons retenu trois.

Tout d'abord, le coût opératoire. Il concerne le coût de réalisations des différentes tâches, c'est-à-dire les tâches à valeur ajoutée et celles à non-valeur ajoutée qui sont induites par le manque d'interopérabilité (dans le modèle AS IS). Il cumule le salaire de l'opérateur et, le cas échéant, le coût d'exploitation.

Ensuite nous retiendrons le coût de mise en place de la solution qui permet la résolution du problème d'interopérabilité (dans le modèle TO BE). En effet, pour être objectifs il nous faut calculer le coût de la mise en place de la solution préconisée.

Nous retiendrons enfin le coût des conséquences des problèmes d'interopérabilité. Comme nous l'avons vu, ces problèmes entraînent des répercussions en terme financier (pénalités, coûts de réexpédition dans le cas d'une adresse erronée, etc.). Ce coût sera donc pris en compte dans le cadre des modèles AS IS.

Nous présentons ci-après les étapes réalisées pour chacun des types de modèles.

5.4.1.2.1 Estimation du coût dans les modèles AS IS

Pour les modélisations de l'existant, nous n'avons évidemment pas retenu le coût de la mise en place de la solution.

En ce qui concerne le coût opérationnel, nous avons ajouté le coût salarial de l'employé pour la durée de la réalisation de la tâche au coût d'exploitation (par exemple le coût de l'impression des BP). Nous avons en revanche écarté de nos calculs les coûts qui restaient stables après la mise en place de la solution (par exemple le coût de fonctionnement et de maintenance de la balance).

Dans le cas des coûts des conséquences de l'interopérabilité, nous avons procédé de la manière suivante : nous avons tout d'abord calculé la somme des coûts engendrés par le manque d'interopérabilité en aval du processus au cours des 12 derniers mois précédant le départ du projet.

Ces coûts sont essentiellement composés des pénalités financières, des frais de transport dans le cas de réexpédition de la marchandise et enfin des gestes commerciaux effectués dans certains cas en compensation de l'erreur commise. La valeur obtenue pour chacune des tâches a ensuite été divisée par le nombre de jours de travail par an du service des expéditions.

Le coût affiché dans les modèle AS IS correspond à la somme des deux coûts présentés ci-avant.

5.4.1.2.2 Estimation du coût dans les modèles TO BE

En ce qui concerne les modélisations du projet futur, nous avons retenu le coût opérationnel, le coût de la mise en place de la solution mais aussi le coût des conséquences de la non-qualité des données.

Pour le coût opérationnel nous avons procédé de la même manière que dans le cas de la modélisation de l'existant.

Le coût de la mise en place a été calculé en définissant une période d'amortissement. La somme globale a donc été divisée par le nombre de jours de travail du service des expéditions pendant cette période, puis par le nombre de tâches impliquées dans les échanges de données dans le modèle réalisé.

Enfin, nous avons choisi de conserver le coût des conséquences de la non-qualité des données car il est illusoire de penser que la seule résolution des problèmes d'interopérabilité éradiquera totalement les problèmes de qualité des données. Nous avons donc conservé ce coût en proportion de l'objectif de qualité qui a été fixé.

5.4.1.3 Estimation du délai

Dans le cas qui nous préoccupe, il s'agira du temps de réalisation des tâches impliquées dans les échanges de données (qu'ils aient été résolus ou non). Ce temps doit évidemment être mesuré dans le cas de la modélisation de l'existant et estimé dans celui de la modélisation du projet futur.

5.4.2 MODELE AS IS

Nous allons donc commencer par représenter le système existant dans le modèle AS IS.

Dans ce modèle, nous n'avons utilisé que l'extension *dataInteroperabilityBarrier* pour représenter les échanges de données. En effet, à ce stade les problèmes d'interopérabilité n'étaient pas encore résolus, et il n'y avait aucun échange de données pour lesquels l'interopérabilité était assurée.

En ce qui concerne l'extension *performanceMeasurement*, nous l'avons affectée à chacune des tâches concernées par les échanges de données, qu'elles soient des tâches à valeur ajoutée ou non. Nous l'avons aussi affectée aux différents sous-processus regroupant les tâches impliquées dans chacun des échanges de données afin de pouvoir comparer les mesures qui y sont affichées avec celles correspondant aux mêmes échanges de données dans le modèle TO BE.

Nous avons aussi associé une annotation textuelle à chacun des échanges de données, ainsi qu'au modèle (située en haut à gauche et non associée). Ces annotations contiennent un lien vers la grille d'évaluation AMDEC données correspondant au document échangé. Celle qui est associée au document correspond à la grille de synthèse. Il s'agit ici de tableurs en ligne, mais il pourrait bien entendu s'agir de liens locaux ou vers le réseau d'entreprise.

Le modèle AS IS est présenté dans la Figure 53 et la grille AMDEC correspondant au premier point d'interopérabilité dans la Figure 54

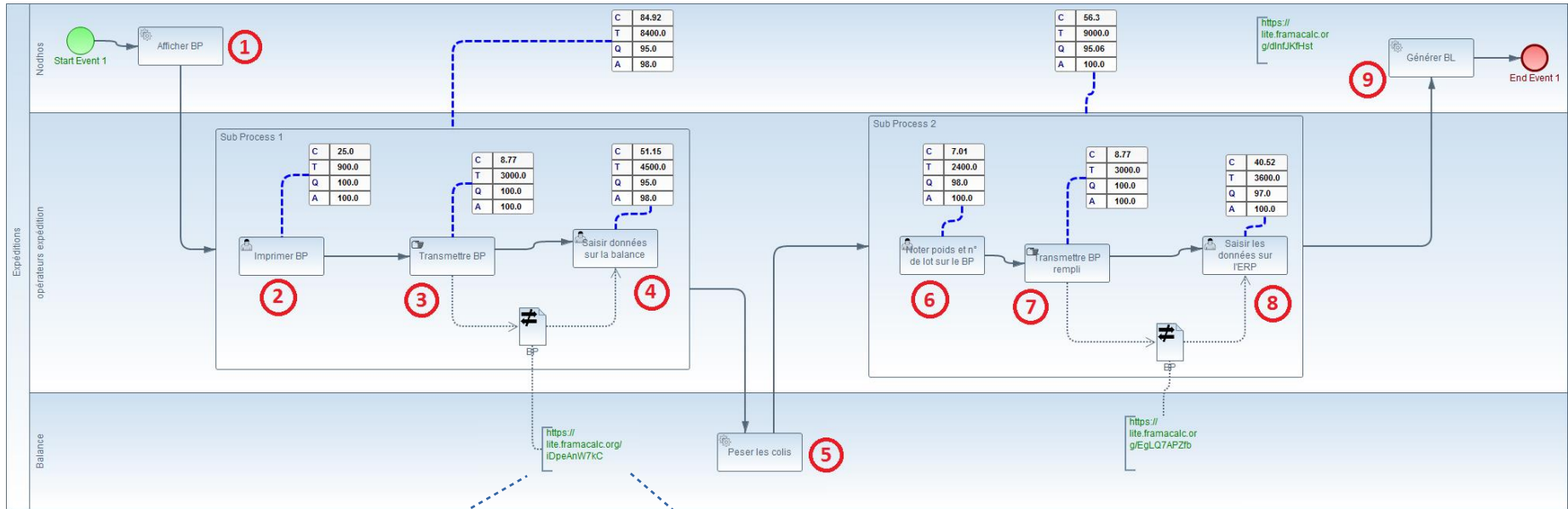


Figure 53 : Modèle AS IS

	A	B	C	D	E F G H				I	J K L M			
1	Donnée	Type de défaillance	Causes	Effets	Évaluation				Action corrective	Résultat			
2					F	N	G	C		F'	N'	G'	C'
3	Identifiant client	Donnée fausse	Erreur de saisie	Commande réexpédiée	3	1	4	12	Projet IO				
4	N° de lot	Non disponible	Non saisi	Pénalité financière	4	1	4	16	Projet IO				
5	N° de lot	Non disponible	Non saisi	Réclamation client	4	1	3	12	Projet IO				
6													

Figure 54 : Grille d'évaluation AMDEC données

Le déroulement du processus est le suivant :

1. le bon de préparation (BP) est affiché à l'écran par l'ERP ;
2. un opérateur imprime le BP ;
3. l'opérateur apporte le BP papier à l'opérateur de pesée ;
4. l'opérateur de pesée saisit les données suivantes sur la balance : identifiant produit, identifiant client, tare, nombre de colis de ce produit, DLUO (date limite d'utilisation optimale) et numéro de lot ;
5. l'opérateur de pesée met le colis sur la balance qui pèse le colis ;
6. l'opérateur de pesée note le poids et le numéro de lot sur le BP papier ;
7. l'opérateur de pesée transmet le BP ;
8. un opérateur saisit le poids et le numéro de lot sur le BP dans l'ERP ;
9. l'ERP génère un bon de livraison (BL) à partir du BP.

Le processus est ici simplifié, mais les tâches non modélisées ne sont pas directement impactées par les problèmes d'interopérabilité et il ne nous a pas paru nécessaire, dans un souci de clarté, de les faire apparaître sur le modèle.

On constate ici deux points d'interopérabilité, représentés par l'extension *dataInteroperabilityBarrier*, qui nécessitent une ressaisie des données :

- Le premier est constitué par le transfert des données contenues dans le BP vers la balance de pesée. Ceci engendre les tâches à non-valeur ajoutée 2, 3 et 4.
- Le second est constitué par le transfert du BP complété par le poids et le numéro de lot depuis la balance vers l'ERP. Le problème d'interopérabilité entraîne les tâches à non-valeur ajoutée 6, 7 et 8.

Les valeurs affichées dans les différents tableaux de type *performanceMeasurement* correspondant à des tâches ont été calculées selon la démarche décrite au cours du paragraphe 6.4.2. Elles ont ensuite été saisies dans les champs leur correspondant dans les propriétés de cette extension. Les valeurs des tableaux correspondant aux sous-processus ont été calculées automatiquement par l'extension *performanceMeasurement*.

Nous faisons apparaître dans la Figure 54 la grille d'évaluation dont nous parlerons plus bas hébergée sur le site framacalc correspondant au premier document échangé pour illustrer le lien entre les modèles réalisés avec BPMN 2.0 et la méthode AMDEC données.

5.4.3 MODELE TO BE

Nous avons ensuite réalisé le modèle correspondant au système souhaité. Puisque dans ce système les problèmes d'interopérabilité ont été résolus, les échanges de données sont représentés par l'extension *dataInteroperabilityResolved*. Ce modèle a été élaboré avec la direction d'Onetik et le chef de projet de la société AGI qui développe l'ERP Integraal Agro. Les responsables du service des expéditions ont participé à sa validation. Ce modèle est présenté dans la Figure 55.

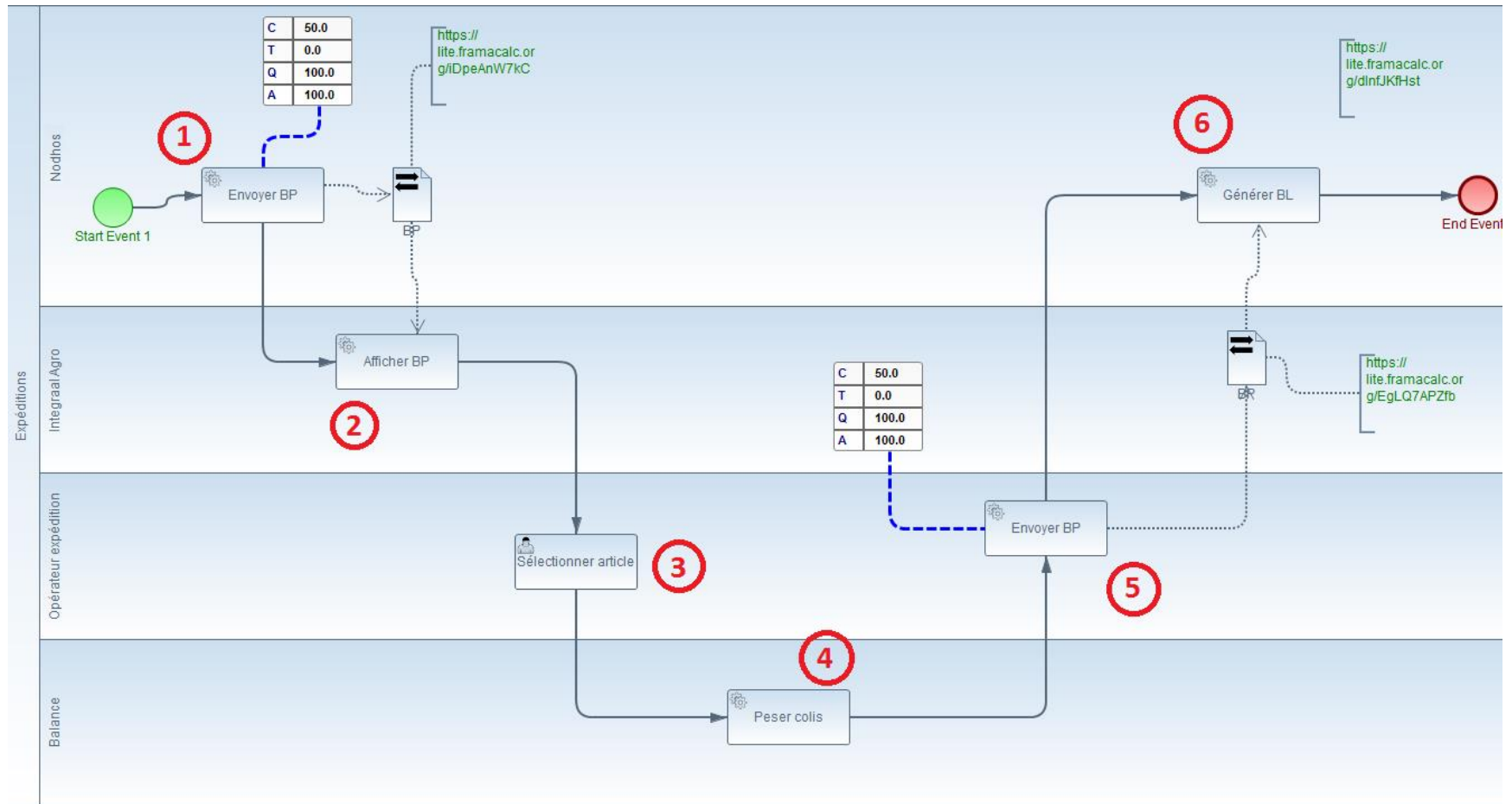


Figure 55 : Modèle TO BE

Le nouveau déroulement du processus est le suivant :

1. le bon de préparation (BP) est envoyé par l'ERP vers le nouveau module de gestion des expéditions ;
2. Le module de gestion des expéditions affiche le BP sur la balance ;
3. l'opérateur de pesée sélectionne la ligne à peser et le numéro de lot à utiliser ;
4. l'opérateur de pesée met le colis sur la balance qui pèse le colis ;
5. à la fin de la pesée de tous les produits de la commande, le module de gestion des expéditions retourne le BP vers l'ERP ;
6. l'ERP génère un bon de livraison à partir du BP.

Les valeurs affichées dans les différents tableaux de type *performanceMeasurement* correspondent toutes à des tâches à valeur ajoutée, car celles à non-valeur ajoutée ont été supprimées. Elles ont été calculées selon le raisonnement vu plus haut.

5.4.4 COMPARAISON DES MODELES

Une fois les modèles réalisés, nous avons pu les présenter à l'équipe projet afin de procéder à leur comparaison.

La première chose qui a marqué les acteurs a été la réduction du nombre de tâches. Le modèle AS IS comporte 9 tâches tandis qu'il n'y en a que 6 dans le modèle TO BE. L'élimination des tâches à non-valeur ajoutée est ce qui marque au premier coup d'œil.

Nous avons ensuite expliqué la signification des nouvelles extensions. Nous avons d'abord rapidement décrit les extensions *dataInteroperabilityBarrier* et *dataInteroperabilityResolved*, donnant ainsi la possibilité de visualiser les points d'interopérabilité posant problème dans un cas et résolu dans l'autre.

Enfin, nous avons introduit l'extension *performanceMeasurement* et détaillé la nature des différents indicateurs affichés. Nous avons précisé qu'ils n'étaient reliés qu'à des tâches impliquées dans les échanges de données impactées par l'interopérabilité. Nous avons ensuite expliqué que les regroupements en sous-processus dans le modèle AS IS correspondaient aux tâches impliquées dans un même échange de données, et indiqué que le tableau relié au sous-processus affichait les mesures de la performance de l'ensemble de ce sous-processus. Nous leur avons finalement dit que les valeurs affichées dans ces tableaux reliés à un sous-processus devaient être comparées à celles du modèle TO BE.

L'effet ne s'est pas fait attendre et l'ensemble des participants a immédiatement pris la mesure de l'enjeu que représentait l'élimination des problèmes d'interopérabilité, ce qui était l'effet escompté.

5.4.5 MODELE GLOBAL

Nous n'avons vu dans les exemples ci-dessus que des processus simplifiés. Certaines tâches n'y sont pas représentées car elles n'impactent pas la qualité des données : mise des fromages dans les cartons, collage de l'étiquette provenant de la balance sur le carton, etc. De plus, ces modèles ne représentent que le processus de préparation des commandes de fromages entiers dans des cartons pesés en vrac.

Dans la Figure 56 ci-dessous, nous avons représenté le modèle AS IS global de la préparation des commandes, abstraction faite des spécificités les moins courantes.

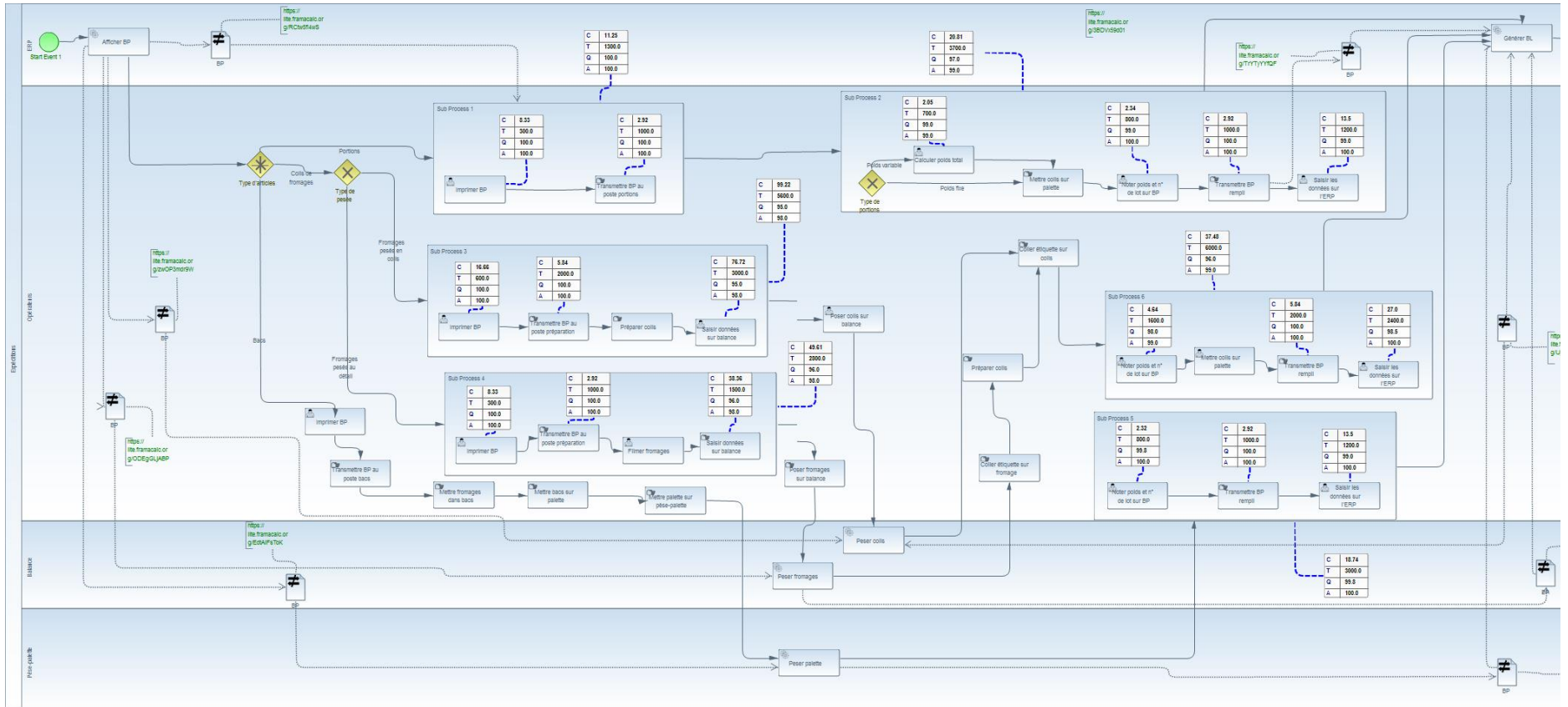


Figure 56 : Modèle global de la préparation des commandes

On peut y voir que tous les types de préparation étaient impactés par des problèmes d'interopérabilité. Ceci dit, le processus de préparation des commandes de fromages en bacs n'a pas été modifié à l'issue de la mise en place du projet pour deux raisons. Tout d'abord, la balance qui sert à peser les palettes de bacs n'est pas interfaçable avec un système informatique. Ensuite, ce processus compte peu de tâches à non-valeur ajoutée, et le taux de non-qualité des données y est faible.

On voit aussi que la lecture de ce modèle n'est pas des plus aisées pour un public non averti. C'est la raison pour laquelle il a été choisi de réaliser des modèles simplifiés représentant chacun un type de commande particulier.

5.5 ELABORATION DE L'AMDEC DONNEES

Parallèlement à ces modèles, nous avons mis en place notre méthode de gestion de la qualité des données. Voici les étapes que nous avons suivies.

5.5.1 ÉTAPE 1 : INITIALISATION

Comme nous l'avons défini lors de la présentation de la méthode, le périmètre du projet correspond à celui de la résolution des problèmes d'interopérabilité, c'est-à-dire ici le service des préparations de commandes des expéditions.

Les objectifs en termes de qualité sont définis pour chacune des données intervenant dans le processus.

L'équipe projet comprenait le directeur d'usine, les deux responsables du service expéditions et le responsable du système d'information.

5.5.2 ÉTAPE 2 : ANALYSE FONCTIONNELLE

Nous avons commencé par l'identification du système, de ses fonctions et de ses composants.

Dans notre cas, le système correspond au système d'information d'Onetik, l'ERP Nodhos, ou plus précisément à son module de gestion des expéditions.

Les composants de ce système sont les documents échangés qui présentent un problème d'interopérabilité, c'est-à-dire les bons de préparation qui transitent dans un premier temps depuis Nodhos vers les balances de pesée puis en sens inverse dans un second temps, ainsi que les données qui y sont contenues.

Un bon de préparation est visible dans la Figure 57.

ONETIK
 64240 MACAYE
 Tel : 05.59.70.27.28
 Fax : 05.59.29.49.74

Destinataire	
Client	Page: 1/ 1

BON DE PREPARATION 499697 du 05/10/18

Transporteur : STEF1
Expédition le : 05/10/18

COMMANDE

[Redacted]

[Redacted]

[Redacted]

[Redacted]

LIVRAISON

[Redacted]

[Redacted]

[Redacted]

[Redacted]

V/Réf.:02437405
 N/Réf.:533961/4948

Code article	Désignation	Nb colis à livrer	Qté. à livrer	Poids théorique	Poids pesé	DLC Quantième
32160	BLEU DES BASQUES 3KG (x2)	4.00	8.00	26.8		19/12/18
41150	TOMME VACHE ET BREBIS (x2)	7.00	14.00	58.4		02/02/19
13150	COEUR BASQUE PRIMEUR (x2)	7.00	14.00	60.8		02/02/19
2215PIMENT	PUR BREBIS PAYS BASQUE AU PIMENT D ESPELETTE	4.00	8.00	36.9		14/12/18
27150	PORTION AOP PUR BREBIS OSSAU IRATY POIDS FIXE	1.00	15.00	3.0		09/11/18
TOTAL		23.00	59.00	185.8		

Mode Expédition: Transport par route

Figure 57 : Exemple de bon de préparation (BP)

Il s'agit d'un cas réel, dans lequel les données du client ont été masquées. On peut y voir pour chaque référence commandée :

- Le nombre de colis à préparer. Il doit être saisi sur la balance.
- Le nombre de fromages nécessaires à la préparation de la ligne de commande.
- Le poids théorique. Si le poids réel est trop éloigné du poids théorique, on peut penser que les fromages utilisés ne sont pas ceux qui correspondent à la commande ou que leur nombre n'est pas correct.
- Le poids pesé. Il doit être noté par l'opérateur à la fin de la pesée.
- La colonne « DLC Quantième » fait double emploi :
 - La DLC qui y est présente doit être saisie sur la balance.
 - A côté, on notera le numéro du lot utilisé pour préparer la commande.

5.5.3 ÉTAPE 3 : ANALYSE QUALITATIVE ET QUANTITATIVE

Identification des modes de défaillance : nous avons dressé la liste des défaillances déjà constatées ou envisageables quant à la qualité des données. Cette liste est la suivante :

- donnée fausse : la valeur de la donnée ne correspond pas à la réalité ;
- donnée incomplète : la donnée n'est pas présente dans sa totalité ;
- donnée incohérente : la valeur de la donnée n'est pas réaliste ;
- donnée non disponible : la donnée est absente.

Détermination des causes de la non-qualité : nous avons passé en revue les différentes données pour lesquelles des cas de non-qualité avaient été constatés et avons tenté d'en trouver les causes. Nous avons fait ceci en nous basant sur le fichier du service après-ventes et en recoupant les données qui y étaient contenues avec le témoignage des différents opérateurs du service des expéditions. Ces causes sont :

- erreur de saisie lors de la prise de commande (en amont du processus d'expédition) ;
- erreur de saisie, lors de la saisie des lots par exemple (en amont du processus d'expédition) ;
- erreur de ressaisie lors du processus d'expédition ;
- donnée non saisie ;

Détermination des effets de la non-qualité : les effets décelés sont de deux types : les effets internes au service des expéditions et les effets externes. Ces effets sont :

- Les effets internes au service des expéditions : si la non-qualité est repérée pendant le processus de pesée, les opérateurs vont aller rechercher l'information correcte et donc perdre du temps.
 - Temps peu important passé à retrouver la donnée correcte (moins de 5 minutes)
 - Temps important passé à retrouver la donnée correcte (plus de 5 minutes)
- Les effets externes : ce sont les conséquences sur les clients qui entraînent la plupart du temps des surcoûts (pénalités financières, frais de réexpédition, etc.) :
 - Donnée impossible à corriger. Légère insatisfaction du client. Faible coût engendré.
 - Donnée impossible à corriger. Insatisfaction importante du client. Coût important engendré.

Évaluation de la criticité : pour cette phase, nous avons établi et complété les différents tableaux que nous avons définis au cours du paragraphe 3.4.2.3 afin de déterminer les valeurs de la fréquence de non-qualité, de la fréquence de non détection de la non-qualité et de la gravité de ses effets. Nous présentons dans la suite ces différents tableaux.

1. **Fréquence de non-disponibilité** : après avoir examiné les documents du service après-ventes et après les avoir recoupés avec le témoignage des opérateurs du service des expéditions,

nous avons établi la grille d'évaluation de la fréquence présentée dans le Tableau 15. Pour tenir compte du fait que certaines données étaient absentes plusieurs fois par jour, nous avons tenu à leur attribuer le niveau « Très fréquent ». Les valeurs des autres critères en ont découlé.

Niveau	Critère	Valeur de la fréquence F
Très rare	1 fois par mois	1
Rare	1 fois par semaine	2
Peu Fréquent	2 à 3 fois par semaine	3
Fréquent	1 fois par jour	4
Très fréquent	Plusieurs fois par jour	5

Tableau 15 : Grille d'évaluation de la fréquence utilisée chez Onetik

2. **Fréquence de non détection de la non-qualité** : les valeurs de la grille d'évaluation que nous avons définies apparaissent dans le Tableau 16.

Probabilité de non détection	N
$P \leq 0,25$	1
$0,25 \leq P \leq 0,5$	2
$0,5 \leq P \leq 0,75$	3
$0,75 \leq P \leq 1$	4

Tableau 16 : Grille d'évaluation de la non détection utilisée chez Onetik

3. **Gravité des effets de la non-qualité** : en ce qui concerne la gravité, nous avons repris telle quelle la grille d'évaluation que nous avons définie dans le paragraphe 3.4.2.3. La différenciation entre les valeurs « Grave » et « Très grave » étant faite en fonction de l'importance de l'éventuel surcoût occasionné. La perte d'un client entre également en ligne de compte quant à cette différenciation. Cette grille d'évaluation est représentée dans le Tableau 17.

Niveau	Critère	Valeur de la gravité G
Très faible	Temps peu important passé à retrouver la donnée correcte (moins de 5 minutes)	1
Faible	Temps important passé à retrouver la donnée correcte (plus de 5 minutes)	2
Grave	Donnée impossible à corriger. Légère insatisfaction du client. Faible coût supplémentaire.	3
Très grave	Donnée impossible à corriger. Insatisfaction importante du client. Coût supplémentaire important.	4

Tableau 17 : Grille d'évaluation de la gravité utilisée chez Onetik

Une fois ces grilles d'évaluation déterminées, nous avons pu remplir les grilles AMDEC données correspondant aux deux échanges de données présents dans le processus avec les différentes données identifiées précédemment. Une vue partielle de cette grille est présentée par le Tableau 18.

Donnée	Type de défaillance	Causes	Effets	Évaluation				Action corrective	Résultat			
				F	N	G	C		F'	N'	G'	C'
Identifiant client	Donnée fausse	Erreur de ressaisie	Commande réexpédiée	3	1	4	12	Projet IO				
N° de lot	Non disponible	Non saisi	Pénalité financière	4	1	4	16	Projet IO				
Poids	Donnée fausse	Erreur de ressaisie	Réclamation client	4	1	3	12	Projet IO				

Tableau 18 : Grille AMDEC données utilisée chez Onetik

5.5.4 ÉTAPE 4 : SYNTHÈSE

Une fois les deux grilles d'évaluation complétées, nous avons fusionné leurs données dans le tableau de synthèse représenté dans le Tableau 19. Nous y avons nommé les documents ainsi :

- BP 1 : BP au stade du premier échange de données.
- BP 2 : BP au stade du deuxième échange de données.

Document	Donnée	Type de défaillance	Causes	Effets	Évaluation				Action corrective	Résultat			
					F	N	G	C		F'	N'	G'	C'
BP 1	Identifiant client	Donnée fausse	Erreur de ressaisie	Commande réexpédiée	3	1	4	12	Projet IO				
BP 2	N° de lot	Non disponible	Non saisi	Pénalité financière	4	1	4	16	Projet IO				
BP 2	Poids	Donnée fausse	Erreur de ressaisie	Réclamation client	4	1	3	12	Projet IO				

Tableau 19 : Grille de synthèse utilisée chez Onetik

Avant tout, il faut souligner que la première phase de l'AMDEC données réalisée dans le cadre de l'établissement de l'interopérabilité des données est spécifique. En effet, la première action corrective est déjà connue : il s'agit de l'établissement de l'interopérabilité des données. Elle concerne l'ensemble des données échangées tout au long du processus en question et est censée avoir un effet général significatif quant à leur qualité. Toutefois, les valeurs de la criticité calculées à ce stade vont servir de base à la phase d'amélioration continue qui démarrera une fois l'interopérabilité établie. On pourra ainsi apprécier l'augmentation de la qualité engendrée par cette action, et ce pour chacune des données impliquées par le processus.

Par la suite, les défaillances relevées et leur criticité donneront lieu à une hiérarchisation ainsi qu'à des recommandations d'actions correctives adaptées, comme dans le cas d'un AMDEC classique.

5.5.5 PHASE D'AMELIORATION CONTINUE

La grille de synthèse nous a servi, après l'établissement de l'interopérabilité, à gérer la qualité des données. Le Tableau 20 nous présente une partie de la grille d'évaluation mise à jour après la mise en place de ce projet.

Document	Donnée	Type de défaillance	Causes	Effets	Évaluation				Action corrective	Résultat			
					F	N	G	C		F'	N'	G'	C'
BP 1	Identifiant client	Donnée fausse	Erreur de ressaisie	Commande réexpédiée	3	1	4	12	Surveillance	1	1	4	4
BP 2	N° de lot	Non disponible	Non saisi	Pénalité financière	4	1	4	16	Sensibilisation du personnel	2	1	4	8
BP 2	Poids	Donnée fausse	Erreur de ressaisie	Réclamation client	4	1	3	12	Surveillance	1	1	3	3

Tableau 20 : Phase d'amélioration continue

On remarque au premier coup d'œil que la criticité de la non-qualité a chuté spectaculairement. Ce résultat était escompté, mais il subsiste encore des défaillances dans la qualité des données. Des actions correctives sont désormais prises concernant les données présentant la criticité la plus importante. Dans certains cas, la criticité est très faible. Ceci est dû principalement au fait que les données sont désormais transmises directement d'un système à l'autre, sans intervention humaine, c'est-à-dire sans ressaisie. Il a été choisi comme action corrective une surveillance de cette criticité, car il serait illusoire de penser que la qualité de ces données est établie ad vitam æternam.

La phase d'amélioration continue a pris le relai du projet d'établissement de l'interopérabilité dans l'optique de l'amélioration de la qualité des données. La perpétuation de l'application de cette méthode entraînera une amélioration et aidera à maintenir une qualité des données optimale dans le processus d'expédition des commandes.

5.6 CONCLUSION

Au cours de ce chapitre nous avons présenté un cas d'étude réalisé à partir de nos travaux au sein de la PME Onetik.

Nous avons commencé par présenter l'entreprise et le problème d'interopérabilité auquel était confronté son département des expéditions. Nous avons aussi évoqué le fait que la décision de greffer le module de gestion des expéditions d'un autre ERP avait été adoptée.

Ensuite, nous avons exposé les travaux que nous avons effectués. En détails, ce chapitre présente des modèles que nous avons réalisés à l'aide des extensions de BPMN 2.0 proposées : le modèle AS IS permet de dresser un portrait de l'existant, puis le modèle TO BE dépeint le fonctionnement de ce même service une fois l'interopérabilité établie. Lors de la phase concertation en entreprise, la comparaison de ces deux modèles a permis à l'ensemble des participants au projet d'identifier et d'appréhender l'intérêt que représente une telle transformation vers l'interopérabilité.

Suite à cela, nous avons décrit la mise en place de la méthode AMDEC données que nous avons proposée lors du chapitre 3. Nous en avons suivi toutes les étapes : initialisation, analyse fonctionnelle, analyse qualitative et quantitative et enfin synthèse. Nous avons notamment établi puis renseigné les grilles d'évaluation correspondant aux différents échanges de données, et en avons déduit la grille de synthèse. Nous avons ainsi vu que les calculs de la criticité de chacune des données échangées nous serviraient par la suite pour apprécier l'amélioration de la qualité des données et déterminer les actions correctives à entreprendre dans l'optique de l'amélioration continue de la qualité des données.

CONCLUSION GENERALE

Dans ce mémoire de thèse de doctorat, nous avons défini des outils permettant de contribuer à une méthode de résolution de l'interopérabilité des données et plus généralement d'amélioration continue de leur qualité.

Nous avons commencé par présenter la problématique de l'interopérabilité, les niveaux de décision auxquels elle se rencontre et les différentes approches envisageables quant à sa résolution. Nous avons aussi établi le fait que cette problématique souffrait d'une carence de connaissance auprès de nombreux publics présents dans les entreprises. Nous avons présenté la nécessité d'utiliser des modèles représentant le système existant et celui à mettre en place pour établir l'interopérabilité, notamment des modèles de haut niveau offrant la possibilité de communiquer sur ce sujet avec différents acteurs. Nous avons aussi défini les indicateurs de performance comme éléments nécessaires à la visualisation plus formalisée des conséquences d'un manque d'interopérabilité. Constatant que l'établissement de l'interopérabilité était une condition nécessaire mais non suffisante pour une qualité des données optimale, nous avons déduit le besoin d'une méthode de gestion de la qualité des données. Nous avons dit que cette méthode devait permettre d'avoir une approche détaillée et quantitative des données échangées.

L'état de l'art réalisé a recensé une palette de langages de modélisation et de méthodes de mesure de la performance les plus appropriés. Il nous a permis de constater plusieurs états de fait. Tout d'abord, qu'aucun des langages de modélisation que nous avons étudiés ne permettait de représenter les points d'interopérabilité, qui cependant constituent une exigence majeure de notre problématique. Ensuite, que les méthodes de mesure de la performance évoquées ne répondaient pas à notre besoin d'action locale de par leur nature globale et la lourdeur de leur mise en place. Sans avoir trouvé de langage directement adapté à la problématique, nous en avons conclu que BPMN 2.0 constituait le langage de modélisation le plus proche de nos besoins de modélisation. Les raisons en sont que c'est un langage de haut niveau qui permet de visualiser les échanges de données de manière évidente, qu'il offre la possibilité de réaliser des vues comportementales et enfin que son mécanisme d'extension nous autorise à développer les éléments de modélisation qui répondent aux exigences que nous avons formulées. Sa très grande popularité dans les entreprises a aussi contribué à faire pencher la balance de son côté.

Nous avons donc opté pour le développement d'extensions de BPMN 2.0 permettant :

- de faire apparaître dans les modèles les échanges de données présentant un problème d'interopérabilité ;
- de faire apparaître dans les modèles les échanges de données pour lesquels les problèmes d'interopérabilité ont été résolus ;
- d'afficher au sein de ces mêmes modèles les valeurs des mesures de la performance du coût, de la durée, de la qualité des données ainsi que de leur disponibilité pour les tâches directement impactées par les échanges de données concernées par l'interopérabilité.

Nous avons présenté des méthodes de réduction de processus et d'agrégation de la performance dont nous avons besoin pour calculer les valeurs de la performance d'un ensemble de tâches. Ce calcul étant nécessaire, dans les modélisations de l'existant, pour apprécier la performance globale de l'ensemble des tâches impactées par un échange de données pour lequel l'interopérabilité est défaillante, c'est-à-dire les tâches à valeur ajoutée et celles à non-valeur ajoutée. Ceci permet de comparer cette performance avec celle du modèle du même processus une fois l'interopérabilité établie.

Constatant que l'établissement de l'interopérabilité ne suffisait pas à lui seul à garantir une qualité optimale des données, nous avons conçu la méthode « AMDEC données », dérivée de l'AMDEC, pour compléter notre contribution conceptuelle. Son but est double : dans un premier temps, visualiser et analyser les données de manière individuelle, ce qui permet de passer d'une approche globale et qualitative que constituent les modèles réalisés à l'aide des nouvelles extensions à une approche détaillée et quantitative. Cette méthode permet dans un second temps de mettre en place un processus d'amélioration continue ayant pour objectif d'aider à assurer et maintenir dans le temps une qualité des données optimale.

Nous avons ensuite justifié le choix de BPMN2 Modeler en tant qu'outil de développement des extensions proposées. Nous avons présenté ce projet et les différentes technologies sur lesquelles il s'appuie (EDI Eclipse, EMF et Graphiti).

Enfin, nous avons présenté le cas d'étude qui nous a donné l'occasion de mettre en pratique les différents outils que nous avons développés au sein de la PME Onetik. Nous avons ainsi pu démontrer la pertinence des outils que nous avons proposés.

L'ensemble de ces travaux a été diffusé dans 4 conférences internationales listées dans la section intitulée diffusion de résultats.

PERSPECTIVES

Notre proposition d'outils constitue une contribution pour l'identification, l'établissement et la quantification de l'interopérabilité ainsi que pour l'amélioration de la qualité des données. Ces premiers résultats ouvrent des pistes de travail qui restent à explorer.

En premier lieu, l'utilisation des modèles BPMN 2.0 pourrait être complétée par la mise en place de simulations. Ces simulations auraient pour but d'exécuter les modèles du projet futur afin d'en prédire un comportement, ceci afin de vérifier que la dynamique de ces processus correspond bien aux attentes de l'entreprise.

Ensuite, des améliorations restent à apporter au niveau de l'implémentation des extensions dans BPMN2 Modeler ou dans un autre outil. La plateforme développée est une maquette de validation des concepts. Elle n'a pas vocation ni la prétention d'être un outil informatique complètement abouti. En particulier au niveau de la relation entre les modèles et les grilles AMDEC données. A titre d'exemple, il serait souhaitable qu'un lien cliquable sur les modèles BPMN 2.0 amène l'utilisateur directement à la grille d'évaluation correspondant à chacun des échanges de données ou à la grille de synthèse. Or, ce comportement n'est pas possible nativement dans BPMN2 Modeler. Nous n'avons pu qu'afficher ce lien dans une annotation textuelle. Cette amélioration renforcerait l'ergonomie de la liaison entre les modèles et les grilles AMDEC données. Enfin les grilles sont stockées pour le moment sur un réseau interne ou dans un cloud de l'entreprise, leur interconnexion est maintenue par le processus BPMN. Leur administration directe pourrait aussi être le sujet d'une modélisation avec une vue donnée unifiée des différentes grilles. Des outils d'analyse statistique pourraient ainsi être utilisés afin de remonter directement des vues agrégées depuis les grilles AMDEC données.

DIFFUSION DES RÉSULTATS

Xabier Heguy, Gregory Zacharewicz, Yves Ducq, Said Tazi; Interoperability Markers for BPMN 2.0 - Making Interoperability Issues Explicit, in Proceedings of the 2017 2nd International Conference on Electrical, Automation and Mechanical Engineering (EAME 2017), Shanghai, Advances in Engineering Research May 2017, ISBN, 978-94-6252-332-6, doi:10.2991/eame-17.2017.78

Xabier Heguy, Gregory Zacharewicz, Yves Ducq, Said Tazi, A Performance Measurement Extension for BPMN: One Step Further Quantifying Interoperability in Process Model, NICST 2017, Clermont https://www.isima.fr/wp-content/uploads/2017/06/NICST2017_Program_VF.pdf

Xabier Heguy, Gregory Zacharewicz, Yves Ducq, Said Tazi, Une extension du BPMN pour modéliser la mesure de la performance, JIAE 2017, Tarbes <http://jiae2017.enit.fr/#programme>

Xabier Heguy, Gregory Zacharewicz, Yves Ducq, Said Tazi, A Performance Measurement Extension for BPMN - One Step Further Quantifying Interoperability in Process Model, Workshop IESA, Berlin To appear in ISTE editions, http://iesa2018.ipk.fraunhofer.de/fileadmin/user_upload/IESA2018/Documents/I-ESA_2018_booklet.pdf

BIBLIOGRAPHIE

- [AFNOR 1988] AFNOR, N. (Octobre 1988). Terminologie relative à la fiabilité – Maintenabilité – Disponibilité, NF X 60-500, 1988.
- [ANDREWS 2003] - T. Andrews, F. Curbera, H. Dholakia, Y. Golland, J. Klein, F. Leymann, K. Liu, D. Roller, D. Smith, S. Thatte, I. Trickovic and S. Weerawarana. Business Process Execution Language for Web Services. Version 1.1. BEA Systems, International Business Machines Corporation, Microsoft Corporation, SAP AG and Siebel Systems available at: <http://xml.coverpages.org/BPELv11-May052003Final.pdf>, 2003.
- [ATHENA 2005] - “Project A4” (Slide presentation), ATHENA Intermediate Audit 29.- 30., C. Guglielmina and A. Berre, September 2005, Athens, Greece, 2005.
- [AUDIBERT 2013] – L. Audibert. Introduction à la modélisation objet, <http://laurent-audibert.developpez.com/Cours-UML/?page=introduction-modelisation-objet>.
- [BANDARA 2005] – Bandara, W., G. G. Gable, and M. Rosemann M. - Factors and Measures of Business Process Modelling: Model Building Through a Multiple Case Study, European Journal of Information Systems (14)4, pp. 347-360, 2005.
- [BATINI 2009] – C. Batini, C. Cappiello, C. Francalanci and A. Maurino. Methodologies for data quality assessment and improvement. ACM Comput. Surv. 41, 3, Article 16 (July 2009), 52 pages, 2009.
- [BENAMA 2016] – Y. Benama. Formalisation de la démarche de conception d’un système de production mobile : intégration des concepts de mobilité et de reconfigurabilité. Mécanique des matériaux [physics.classph]. Université de Bordeaux, 2016.
- [BERTI-EQUILLE 2004] – L. Berti-Equille, Qualité des données. Ingénierie des systèmes d’information 9:117-143, 2004.
- [BEZIVIN 2001] – J. Bézivin and O. Gerbé. Towards a Precise Definition of the OMG/MDA Framework. Proceedings of the 16th IEEE international conference on Automated software engineering (ASE 2001), San Diego, USA, 273-280, 2001.
- [BOCCIARELLI 2011] – P. Bocciarelli and A. D’Ambrogio. A BPMN extension for modeling non functional properties of business processes. In Proc. of TMS-DEVS, pages 160–168. SCS, 2011.
- [BOUQUIN 2008] – H. Bouquin. Le Contrôle de gestion, Paris : Presses Universitaires de France, 8^e édition, 2008.
- [BOURGUIGNON 1997] – A. Bourguignon. Sous les pavés la plage... ou les multiples fonctions du vocabulaire comptable : l’exemple de la performance. Comptabilité Contrôle Audit 3 (1): 89-101, 1997.
- [BOURGUIGNON 2001] – A. Bourguignon, V. Malleret, H. Norreklit. Balanced Scorecard Versus French Tableau de Bord: Beyond Dispute, A Cultural and Ideological Perspective, 2001.
- [BRAUN 2014] - R. Braun and W. Esswein. Classification of domain-specific bpmn extensions, in: The Practice of Enterprise Modeling, Springer, 2014.
- [CARDOSO 2002] – J. Cardoso. Stochastic Workflow Reduction Algorithm, LSDIS Lab, Department of Computer Science, University of Georgia, 2002.

- [CHEN 2006] – D. CHEN. Framework for Enterprise Interoperability, 2006.
- [CLIVILLE 2004] - V. Clivillé. Approche systémique et méthode multicritère pour la définition d'un système d'indicateurs de performance, Thèse de Doctorat de l'Université de Savoie en Génie Industriel, 2004.
- [Cooper 1988] - R. Cooper, R. S. Kaplan. How cost accounting distorts product costs. Management Accounting, vol. 69, no. 10, pp. 20-27, 1988.
- [CORTES-CORNAX 2014] - M. Cortes-Cornax, S. Dupuy-Chessa, D. Rieu and M. Dumas. Evaluating Choreographies in BPMN 2.0 Using an Extended Quality Framework, 2014.
- [CPC 1997] - Club "Production et Compétitivité" - Coordination P.M. Gallois - "De la pierre à la cathédrale : Les indicateurs de performance" - Ministère de l'Industrie de la Poste et des Télécommunications - Editions Londez Conseil, 1997.
- [DAUM 2005] – J. Daum. French Tableau de Bord: Better than the Balanced Scorecard?, Der Controlling Berater, No. 7, pp. 459-502, 2005.
- [DOUMEINGTS 1984] - G. Doumeingts, Méthode GRAI : méthode de conception des systèmes en productive. Université de Bordeaux 1, 1984.
- [DOUMEINGTS 1998] - G. Doumeingts, B. Vallespir, S. Kleinhaus, Y. Ducq. Performance Indicators and the GRAI Approach, Conference on Performance Measurement, Theory and Practice - Cambridge - UK - 15-17 juillet 1998 - Centre for Business Performance - A.D Neely & D.B. Waggoner ed. - Vol1 pp311-318, 1998.
- [DOUMEINGTS 2005] - G. Doumeingts, Y. Ducq. La Méthodologie GRAI - LAPs/GRAI - Avril 2005.
- [DUCQ 1999] - Y. Ducq. Contribution à une méthodologie d'analyse de la cohérence des Systèmes de Production dans le cadre du Modèle GRAI, 1999.
- [FARMER 1967] – F. R. Farmer. Siting criteria - a new approach, Symposium Containment and Siting of Nuclear Power Plants, International Atomic Energy Agency, Wieden, 1967.
- [FERNANDEZ 1999] - A. Fernandez, Les nouveaux tableaux de bord pour piloter l'entreprise : systèmes d'information, nouvelles technologies et mesure de la performance, Editions d'Organisation, Paris, 1999.
- [FIESCHI 2009] – M. Fieschi. La gouvernance de l'interopérabilité sémantique est au cœur du développement des systèmes d'information en santé. Rapport à la ministre de la santé et des sports, 2009.
- [GERACI 1991] – A. Geraci, F. Katki, L. McMonegal, B. Meyer, J. Lane, P. Wilson, J. Radatz, M. Yee, H. Porteous, F. and Springsteel. IEEE standard computer dictionary : Compilation of IEEE standard computer glossaries, 1991.
- [HARMON 2016] – P. Harmon. The State of Business Process Management 2016. Business Process Trends, 2016.
- [HILL 2000] – D. Hill. Contribution à la modélisation des systèmes complexes : application à la simulation d'écosystèmes., Habilitation à Diriger des Recherches de l'Université Blaise Pascal, 123 p., 2000.
- [KAPLAN 1996] – R.S. Kaplan and D.P. Norton. The Balanced Scorecard: Translating Strategy into Action, Harvard Business Press, 1996.
- [LEGLISE 2009] - M. Léglise, B. Ferriès, Évaluation des coûts des défauts d'interopérabilité supportés par entreprises, maîtres d'ouvrage et exploitants, Déc. 2009.

- [LORINO 2001] – P. Lorino. Les indicateurs de performance dans le pilotage de l'entreprise. Chapitre 3, 49 - 64, dans l'ouvrage Indicateurs de performance, coordonné par C. Bonnefous, A. Courtois, Hermès, 285, 2001.
- [MARMUSE 1997] - C. Marmuse. Performance dans Encyclopédie de gestion, P. Joffre, et Y. Simon (eds), pp. 2194-2208, 1997.
- [NEELY 2002] - A. Neely, C. Adams, M. Kennerley, The Performance Prism: The Scorecard for Measuring and Managing Business Success, Financial Times Prentice Hall, 2002.
- [NEELY 2007] - A. Neely. Business performance measurement unifying theory and integrating practice, 2007.
- [OMG 2003] - OMG: MDA Guide Version 1.0.1, 2003.
- [RAVELOMANANTSOA 2009] - M.S. Ravelomanantsoa. Contribution à la définition d'un cadre générique pour la définition, l'implantation et l'exploitation de la performance : Application à la méthode ECOGRAI, 2009.
- [ROQUE 2014] – M. ROQUE, Marco André Marques. Animation of BPMN business processes models. Lisboa: ISCTE. Dissertação de mestrado , 2014.
- [ROTHENBERG 1989] – J. Rothenberg. The Nature of Modeling, Artificial Intelligence, Simulation and Modeling, John Wiley & Sons, Inc., New York, pp 75-92, 1989.
- [SCHEER 1994] – A.W. Scheer, J. Galler, C. Kruse. Workflow Management within the ARIS framework. In European Workshop on integrated manufacturing systems engineering Chapman & Hall, Grenoble, France, 1994.
- [THE-HIEN 2016] – D. The-Hien. Conciliation médicamenteuse assistée par électronique : rôle des acteurs de santé et moyens de mise en œuvre. Sciences pharmaceutiques, 2016.
- [THYSSEN 2006] – J. Thyssen, P. Israelsen, B. Jørgensen. Activity-based costing as a method for assessing the economics of modularization - A case study and beyond, Int. J. Production Economics. vol. 103, pp. 252-270, 2006.
- [TRUPTIL 2010] - S. Truptil, F. Bénabenand H. Pingaud. A Mediation Information System to Help to Coordinate the Response to a Crisis. Luis M. Camarinha-Matos; Xavier Boucher; Hamideh Afsarmanesh. Collaborative Networks for a Sustainable World, 336, Springer, pp.173-180, IFIP Advances in Information and Communication Technology, 978-3-642-15960-2, 2010.
- [VERNADAT 1998] – F. Vernadat. La modélisation d'entreprise par la méthodologie CIMOSA, actes du Séminaire Modélisation d'Entreprise, Programme de recherche du CNRS PROSPER Systèmes de Production, Roissy-en-France, 1998.
- [VERNADAT 1999] – F. Vernadat. Techniques de Modélisation en Entreprise : Applications aux Processus Opérationnels, p. 129, Edition Economica, France, 1999.
- [WFMC 1999] Workflow Management Coalition. 1999. Terminology & Glossary. WfMC-TC-1011, 3.0, Feb.
- [WANG 1995] - R. Wang, V. Storey et C. Firth : A framework for analysis of data quality research; IEEE Transactions on Knowledge and Data Engineering, 7(4), pp. 670-677, 1995.
- [ZACHAREWICZ 2008] Zacharewicz G., Frydman C.S., Giambiasi N., G-DEVS/HLA Environment for Distributed Simulations of Workflows. Simulation 84(5): 197-213 (2008)
- [ZEIGLER 2000] - B. P. Zeigler, H. Praehofer and T. G. Kim. Theory of modeling and simulation: integrating discrete event and continuous complex dynamic systems, Academic press, 2000.