

TABLE DES MATIÈRES

	Page
INTRODUCTION GENERALE	1
CHAPITRE 1 INTRODUCTION AUX ARCHITECTURES ORIENTEES SERVICES	4
1.1 Introduction.....	4
1.2 Architecture Orientée Services	4
1.2.1 Le concept de service	8
1.2.2 La notion de couplage.....	8
1.2.3 Le principe d'abstraction	10
1.2.4 La notion de réutilisabilité.....	11
1.2.5 L'autonomie d'un service.....	12
1.2.6 La découvrabilité de services	14
1.2.7 La composabilité de services.....	15
1.3 Avantages de l'architecture SOA.....	16
1.4 Limites de l'architecture SOA	17
1.5 Web services : une réalisation de SOA.....	18
1.6 SOA et la modélisation des processus d'affaires.....	20
1.6.1 Notation de modélisation unifiée (UMN).....	21
1.6.2 Langage d'exécution des processus d'affaires BPEL.....	22
1.7 Conclusion	23
CHAPITRE 2 APPROCHES D'IDENTIFICATION DE SERVICES DANS L'ARCHITECTURE SOA	24
2.1 Introduction.....	24
2.2 Cycle de vie de SOA.....	24
2.2.1 Phase d'analyse.....	25
2.2.2 Phase de conception orientée service	26
2.2.3 Phase de développement.....	26
2.2.4 Phase de test.....	27
2.2.5 Phase de déploiement	27
2.2.6 Phase de gestion des services	28
2.3 Approches d'identification de services.....	28
2.3.1 Approche descendante (<i>Top-Down</i>)	28
2.3.2 Approche ascendante (<i>Bottom up</i>).....	31
2.3.3 Approche mixte	32
2.4 Analyse critique des différentes approches d'identification de services	35
2.5 Conclusion	36
CHAPITRE 3 PROPOSITION D'UNE APPROCHE D'IDENTIFICATION DE SERVICES.....	37
3.1 Introduction.....	37
3.2 Principes de l'approche proposée	38

3.3	Description détaillée de la démarche proposée.....	42
3.3.1	Identification des services entités	43
3.3.1.1	Sélection des activités sujettes d'étude	45
3.3.1.2	Identification des informations clés	46
3.3.1.3	Sélection des informations les plus récurrentes	46
3.3.1.4	Sélection des activités les plus sollicitées et des informations qui leur sont corrélées	48
3.3.1.5	Identification des services entités	49
3.3.2	Identification des opérations des services entités	49
3.3.3	Identification des services d'application	52
3.3.4	Identification des opérations des services d'application	53
3.3.5	Identification des services centrés-tâche	54
3.3.5.1	Identifier les activités candidates à être des services centrés-tâche	54
3.3.5.2	Définir la logique des candidats de service centrés-tâche.....	56
3.3.6	Définir la couche d'orchestration	56
3.3.6.1	Identification des processus à orchestrer	57
3.3.6.2	Orchestration des processus d'affaires.....	58
3.3.6.3	Transformation d'un modèle métier en un modèle intermédiaire	58
3.3.6.4	Génération automatique d'un BPEL à partir d'un modèle intermédiaire	60
3.4	Conclusion	65
CHAPITRE 4 VALIDATION DE L'APPROCHE PROPOSEE		67
4.1	Introduction.....	67
4.2	Identification des services entités	67
4.2.1	Sélection des activités sujettes d'étude.....	67
4.2.2	Identification des informations clés.....	68
4.2.2.1	Sélection des informations les plus récurrentes	68
4.2.2.2	Sélection des activités les plus sollicitées et des informations qui leur sont corrélées	70
4.2.3	Identification des services entités	73
4.3	Identification des opérations des services entités	75
4.4	Identification des services d'application.....	78
4.5	Identification des opérations des services d'application.....	79
4.6	Identification des services centrés-tâche.....	80
4.6.1	Identifier les activités candidates à être des services centrés-tâche.....	80
4.6.2	Définir la logique des candidats de service centrés-tâche	82
4.7	Définir la couche d'orchestration.....	83
4.7.1	Identification des processus à orchestrer	83
4.7.2	Orchestration des processus d'affaires	90
4.7.2.1	Transformation d'un modèle métier en un modèle intermédiaire	90
4.7.2.2	Génération automatique d'un BPEL à partir d'un modèle intermédiaire	92
4.8	Conclusion	96

CONCLUSION GENERALE.....	97
ANNEXE I LISTE DES ACTIVITÉS AUTOMATIQUES ET SEMI-AUTOMATIQUES.....	99
ANNEXE II OCCURRENCES DES INFORMATIONS MÉTIERS	102
ANNEXE III LISTE DES ACTIVITÉS LES PLUS SOLLICITÉES.....	104
ANNEXE IV DESCRIPTION DES OPÉRATIONS BASIQUES DES SERVICES ENTITÉ.....	105
ANNEXE V DESCRIPTION DÉTAILLÉE DES OPÉRATIONS DES SERVICES D'APPLICATION	107
ANNEXE VI DIAGRAMMES DE DEPENDENCE POUR LES SERVICES CENTRÉS-TÂCHE	108
ANNEXE VII DIAGRAMMES DE SÉQUENCE DES SERVICES CENTRÉS-TÂCHE	110
ANNEXE VIII ANALYTICAL HIERARCHY PROCESS (AHP)	113
ANNEXE IX MATRICES DE PONDÉRATION.....	117
BIBLIOGRAPHIE.....	119

LISTE DES TABLEAUX

	Page
Tableau 2.1	Étude comparative des approches d'identification de services.....34
Tableau 4.1	Métriques de sélection des informations récurrentes.....69
Tableau 4.2	Premier lot d'informations clés.....70
Tableau 4.3	Métriques de sélection des activités sollicitées.....71
Tableau 4.4	Prototype de matrice pour l'identification des informations corrélées.....72
Tableau 4.5	Informations corrélées aux activités métiers sollicitées.....73
Tableau 4.6	Entités métiers identifiées.....74
Tableau 4.7	Identification des opérations basiques pour les services entité.....76
Tableau 4.8	Prototype de matrice pour l'identification des informations corrélées.....77
Tableau 4.9	Opérations métiers identifiées pour les services entité.....78
Tableau 4.10	Identification des services d'application.....79
Tableau 4.11	Identification des opérations des services d'application.....79
Tableau 4.12	Candidats de services centrés-tâche.....81
Tableau 4.13	Processus d'affaire à orchestrer.....85
Tableau 4.14	Ordre d'orchestration des processus d'affaires.....89
Tableau 4.15	Liste complète des services identifiés.....95

LISTE DES FIGURES

	Page
Figure 1.1	Les trois couches primaires d'une SOA.....6
Figure 1.2	Les différents types de couplage dans SOA.9
Figure 1.3	Centralisation des contrats de service et de la logique métier.12
Figure 1.4	Cas d'une autonomie partagée.....13
Figure 1.5	Scénario de découverte d'un service.15
Figure 1.6	Les principales limites de SOA.....18
Figure 1.7	Scénario d'utilisation d'un web service.19
Figure 1.8	Exemple d'activité dans UMN.21
Figure 2.1	Cycle de vie SOA.....25
Figure 2.2	Les différentes approches d'identification de services.....34
Figure 3.1	Types de services identifiés.38
Figure 3.2	Les six étapes d'identification des services les plus appropriés.42
Figure 3.3	Diagramme de dépendance entre services.55
Figure 3.4	Processus d'orchestration des services.....58
Figure 3.5	Translation d'un modèle MI en un code BPEL.....61
Figure 3.6	Algorithme de génération d'un BPEL à partir d'un MI.....62
Figure 3.7	Exemples de parcours dans un MI.....64
Figure 3.8	Algorithme résumant l'approche proposée.....65
Figure 4.1	Occurrences des informations métiers.....69
Figure 4.2	Occurrences des activités métiers.71
Figure 4.3	Diagramme de dépendance pour le service « VerificationCompte ».82
Figure 4.4	Diagramme de séquence pour le service « PriseRDV ».82

Figure 4.5	Pondération des critères de sélection.	85
Figure 4.6	Matrice de pondération des alternatives par rapport au critère « Sécurité »	86
Figure 4.7	Vecteur de priorité global des alternatives.....	87
Figure 4.8	Graphe de performance pour l'analyse de sensibilité.	88
Figure 4.9	Modèle intermédiaire de processus de création d'un compte bancaire	91
Figure 4.10	Organigramme du processus de création d'un compte bancaire.	92
Figure 4.11	Arborescence de génération d'un code BPEL	93
Figure 4.12	Orchestration du processus de création d'un compte bancaire.	94

LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

AHP	<i>Analytic Hierarchy Process</i> (Analyse Multicritères)
BPEL	<i>Business Process Execution Language</i> (Langage d'exécution des processus d'affaires)
BPM	<i>Business Process Modeling</i> (Modélisation des processus d'affaires)
BPMN	<i>Business Process Modeling Notation</i> (Notation de modélisation des processus d'affaires)
DSI	Directeurs de Systèmes d'Information
EPC	<i>Event Process Chain</i> (Chaîne de processus événementielle)
HTTP	<i>HyperText Transfer Protocol</i> (Protocole de transfert HyperTexte)
IDC	<i>International Data Group</i>
MI	Modèle Intermédiaire
NC	Nœud Courant
OSSAD	<i>Office Support Systems Analysis and Design</i> (Bureau de support des systèmes d'analyse et de conception)
PDF	<i>Portable Document Format</i> (Format de document portable)
ROI	<i>Return On Investment</i> (Retour sur investissement)

SI	Système d'Information
SOA	<i>Service Oriented Architecture</i> (Architectures orientées services)
SOAP	<i>Simple object access protocol</i> (Simple protocole d'accès objet)
TI	Technologie de l'Information
UML	<i>Unified Modeling Langage</i> (Langage de modélisation unifié)
URI	<i>Uniform Resource Identifier</i> (Identificateur de ressources uniforme)
URL	<i>Uniform Resource Locator</i> (Localisateur de ressources uniforme)
UDDI	<i>Universal Description Discovery and Integration</i> (Description, découverte et intégration universelle)
UMN	<i>Unified Modeling Notation</i> (Notation de modélisation unifiée)
W3C	<i>World Wide Web Consortium</i>
WPM	<i>Weight Product Method</i> (Méthode de produit pondéré)
WSDL	<i>Web Services Description Language</i>
WSM	<i>Weight Sum Method</i> (Méthode de la somme pondérée)
XML	<i>Extensible Markup Language</i> (Langage de balisage extensible)

INTRODUCTION GENERALE

L'industrie informatique ne cesse de produire de nouveaux concepts qui créent un effet de mode chez les professionnels consommateurs de cette industrie. Ces concepts généralement prématurés se transforment, à terme, en réalités incontournables. Le relationnel ainsi que l'objet sont là pour en témoigner.

Cela s'applique bien évidemment au concept d'architectures orientées services (SOA) qui viennent pour contribuer à l'adaptation des systèmes d'information au nouvel environnement assez turbulent dans lequel évoluent les organisations.

Basée sur la notion de distribution de ressources, apparue dans les années 90, l'idée de constituer une architecture orientée services, assurant un partage aux moindres coûts des ressources informatiques de l'entreprise, est devenue la clé de voûte de ce qu'on appelle architecture distribuée. Conscientes des opportunités qu'offrent les systèmes d'informations basés sur ce type d'approches architecturales, les entreprises manifestent un intérêt croissant pour l'adoption de ce concept, afin de pouvoir exploiter les avantages que fédère cette architecture. Selon une étude récente d'*International Data Corporation*¹ (IDC), les dépenses des entreprises sur le SOA vont croître de 25% durant les trois prochaines années. Ceci explique l'intérêt qu'exercent les fournisseurs de service informatiques, les plus reconnus au monde, au principe de SOA en introduisant ce concept dans la plupart de leurs produits, les plus récents.

Malgré la frénésie actuelle autour de SOA et le niveau de maturité atteint par cette approche architecturale, il n'existe pas une procédure standard à suivre pour la mise en place d'une telle architecture. Ceci rend plusieurs Directeurs de Systèmes d'Information (DSI) sceptiques

¹ IDC est un acteur majeur de la recherche, du conseil et de l'événementiel sur les marchés des Technologies de l'Information, des Télécommunications et des Technologies Grand Public.

sur le concept de SOA et jugent que la décision d'adopter ce genre d'architecture est très risquée. En effet, la mise en place d'une architecture orientée services présente de nouveaux défis à l'entreprise, étant donné que le cycle de vie SOA est complètement différent de celui pour les solutions informatiques traditionnelles.

L'identification des services d'une architecture orientée services constitue la première phase du cycle de vie de SOA. Elle est considérée comme la phase la plus importante dans un projet de mise en place d'une SOA (Erl, 2005). Cette phase consiste non seulement à déterminer les services qui doivent être mis en place, mais aussi à définir la logique que doit encapsuler chaque service. Une identification appropriée des services à mettre en place peut éviter des coûts importants à l'entreprise et lui permet de tirer pleinement parti des avantages du concept SOA. L'identification des services peut être réalisée à partir de diverses sources d'information. Cette diversité de sources peut induire en erreur les analystes SOA. Le défi est de mettre en place une approche qui assiste le mieux les analystes à fouiller systématiquement et à analyser l'énorme quantité d'information à laquelle il faut faire face. Cette approche devrait aussi contenir un ensemble clair de directives et de bonnes pratiques à suivre afin de pouvoir automatiser certaines tâches manuelles en vue d'accélérer le processus d'identification de services. Ceci permettra de réduire l'intervention humaine et ainsi minimiser les risques d'erreurs.

C'est dans ce cadre d'approches que ce projet de maîtrise s'inscrit, au cours duquel nous essayons d'élaborer une approche semi-automatique, qui assiste les analystes à identifier les services les plus appropriés à leur entreprise. Notre contribution consiste à fournir un outil de support de décision, basé sur une méthodologie structurée en directives amenant à une identification justifiée des services à mettre en place, ceci en analysant statistiquement les processus d'affaires de l'entreprise. Cet outil va permettre l'automatisation de certaines directives suivies dans le processus d'identification. La méthodologie proposée devait bien évidemment respecter les principes de SOA et s'assurer que la logique métier et applicative seront en parfait alignement. La présente approche sera testée et validée sur un cas d'étude réel.

Ce rapport se compose de quatre chapitres, à savoir :

1. Le premier chapitre « INTRODUCTION AUX ARCHITECTURES ORIENTÉES SERVICES » dans lequel nous introduisons des notions de base sur les concepts utilisés tout au long du rapport.
2. Le deuxième chapitre « APPROCHES D'IDENTIFICATION DE SERVICES » au niveau duquel nous présentons les différentes phases du cycle de vie de SOA et nous faisons une synthèse des approches d'identification de services déjà existantes, notamment les approches de types descendantes, ascendantes et mixtes, pour pouvoir par la suite dégager les limites de chacun de ces types d'approches.
3. Le troisième chapitre « PROPOSITION D'UNE APPROCHE D'IDENTIFICATION DE SERVICES » où nous mettons l'accent sur la description de l'approche proposée en expliquant avec minutie les directives à suivre en vue d'une identification justifiée de services.
4. Le quatrième chapitre « VALIDATION DE L'APPROCHE PROPOSEE » est consacré pour la validation et l'application de la présente approche sur un cas réel d'étude.

Une conclusion résume en fin les principales leçons tirées de ce travail et propose quelques perspectives futures.

CHAPITRE 1

INTRODUCTION AUX ARCHITECTURES ORIENTEES SERVICES

1.1 Introduction

Ce chapitre a pour ambition d'établir une revue de littérature sur les architectures orientées services; sans prétendre à l'exhaustivité, notre objectif est que ce chapitre soit suffisamment complet pour qu'une personne non initiée aux architectures des systèmes d'information (SI) perçoive les motivations de notre travail.

Une première partie de ce chapitre sera consacrée à la présentation des architectures orientées services (SOA) et de leurs principes de base. Nous nous intéresserons ensuite à la présentation des avantages et des limites de SOA. Nous allons exposer par la suite les principaux aspects d'une architecture SOA réalisée à base de web services. Avant de clore ce chapitre, nous montrerons le lien qui relie les architectures orientées services à la modélisation des processus d'affaires.

1.2 Architecture Orientée Services

SOA, ou architecture orientée services, est un concept qui a pris beaucoup d'ampleur cette dernière décennie (à partir des années 2000). Cette architecture est le résultat de l'évolution exponentielle qu'ont connu les systèmes informatiques au cours des quarante dernières années. L'idée de concevoir des architectures orientées services est venue avec l'apparition des solutions Client / Serveur, au début des années 1990.

Les architectures traditionnelles (basées sur l'orienté objet et l'orienté composant) ont atteint leurs limites en termes de capacité, alors que les besoins des organisations demeurent en perpétuelle augmentation. Plusieurs architectures ont été conçues afin d'assurer des traitements distribués. Des langages de programmation destinés à être exécutés sur n'importe quelle plateforme et une myriade de produits de connectivité ont vu le jour pour assurer une

intégration rapide et efficace des applications. Toutefois, les coûts et les délais de cette intégration deviennent de plus en plus inacceptables, surtout que les systèmes d'information deviennent de plus en plus complexes.

SOA est une approche évolutive qui contribue à mettre en place un système d'information flexible, assurant ainsi une interopérabilité intrinsèque et une meilleure agilité pour l'entreprise.

Diverses définitions ont été avancées sur le compte de SOA, chacune a sa propre vision sur ce concept. Malgré cette diversité, ces définitions partagent le même point de vue sur ce que SOA ne peut pas l'être. En effet, SOA ne peut pas être considérée, ni comme un produit, ni une solution, ni une technologie comme le croient certains.

Les définitions trouvées dans la littérature peuvent être classées suivant le niveau d'abstraction qu'elles adoptent pour décrire SOA. Dans un niveau d'abstraction plus haut, Girish et al. (2007) décrivent SOA comme un paradigme permettant l'organisation et l'utilisation de capacités distribuées qui pourraient être sous le contrôle de différents propriétaires de domaines.

Dans un niveau d'abstraction moindre, Sandy (2007) définit SOA comme une approche architecturale orientée métier permettant la mise en place d'un système d'information qui sera alignée avec les objectifs métiers de l'entreprise. La clé de voûte de SOA est le service qui n'est autre qu'un composant ou une tâche métier qui se répète fréquemment dans les processus d'affaires de l'entreprise.

Dans un niveau d'abstraction plus pragmatique, SOA peut être définie comme étant une architecture d'applications au sein de laquelle toutes les fonctions sont spécifiées en tant que services indépendants, comprenant des interfaces qui peuvent être sollicitées dans des séquences bien déterminées pour créer des processus d'entreprise (Kishore et al., 2004).

Cette multitude de niveaux d'abstraction adoptés pour définir SOA a influencé considérablement les architectures orientées services contemporaines, qui sont très souvent organisées en couches, offrant ainsi plusieurs niveaux d'abstraction. Cette organisation multicouche a pour objectif de :

- lier le niveau métier, qui est plutôt abstrait, au niveau applicatif, qui est très technique,
- réduire la complexité de l'architecture,
- apporter une plus-value en termes de flexibilité et de couplage faible (*loose-coupling*) à l'échelle de l'entreprise,
- permettre une modélisation d'affaires orientée services,
- et assurer une meilleure agilité organisationnelle.

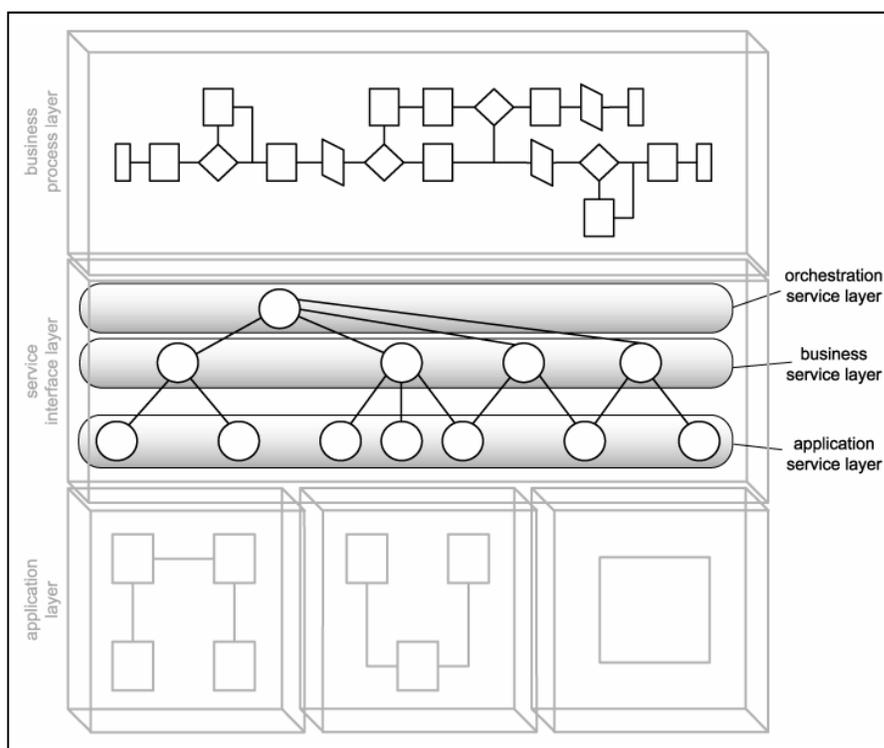


Figure 1.1 Les trois couches primaires d'une SOA.
Tirée de Erl (2005, p. 336)

La Figure 1.1 présente les différentes couches qui peuvent former une architecture orientée services multicouche (Erl, 2005), à savoir :

- **La couche de services d'application** : cette couche établit le niveau d'abstraction le plus bas, qui peut exister dans une SOA. Elle a pour rôle d'exprimer des fonctionnalités spécifiques au système d'information de l'entreprise. Les services qui résident au sein de cette couche peuvent être simplement appelés des services d'application. Le but de ce type de services est de fournir des fonctions réutilisables, liées aux traitements techniques réalisés sur les données, dans les environnements applicatifs.
- **La couche de services d'affaires** : Les services d'affaires sont la pierre angulaire des architectures SOA contemporaines. Leur unique but consiste à représenter la logique d'affaires en sa forme la plus pure. Ces services peuvent englober soit des entités ou des tâches métiers.
- **La couche d'orchestration de service** : elle introduit le niveau d'abstraction le plus élevé dans SOA. Cette couche vient pour répondre aux besoins d'avoir des services, intitulés les services de processus (*process services*), qui gèrent les interactions entre les opérations d'autres services, ceci afin d'assurer l'exécution de ses dernières (les opérations) dans un ordre bien déterminé. Les services de processus sont créés à partir de la composition d'autres services, qui offrent un ensemble spécifique de fonctions, indépendamment des règles d'affaires et de la logique des scénarios, requis pour l'exécution d'un processus.

L'implémentation d'une architecture orientée services multicouche présente un défi pour l'entreprise, surtout qu'il n'existe pas une approche standard pour la mise en place d'une telle architecture. Cependant, il y a un ensemble de principes de base à respecter dans le processus de conception d'une architecture orientée services. Parmi ces principes, nous pouvons citer : le couplage, l'abstraction, la réutilisabilité, l'autonomie, la découvrabilité et la composabilité (Erl, 2008). Dans ce qui suit, nous allons décrire chacun de ces principes, tout en commençant par définir le concept de service.

1.2.1 Le concept de service

Le service est la brique de base de l'architecture orientée services. La granularité d'un service peut varier selon le choix du concepteur du système d'information. Un service peut se réduire à une simple tâche élémentaire comme il peut englober tout un processus métier, d'ordre de complexité plus ou moins importante.

Un service doit disposer d'une interface qui permet de dialoguer avec lui. Par analogie, cette interface joue le rôle d'un contrat entre le fournisseur et le consommateur de service. Le contrat indique notamment la liste des opérations disponibles, la manière de les employer (entrées/sorties attendues), ainsi que la ressource physique (le serveur) sur laquelle se trouve le service (Rouillard, Vantrois et Chevrin, 2007) .

Les caractéristiques d'un service peuvent parfois faire illusion au concept d'« Objet » du paradigme orienté objet, cependant une grande différence existe. Un objet est basé sur le principe d'encapsulation d'un ensemble de données et de méthodes manipulant ces données. Afin d'exploiter l'objet, ce dernier doit être initialement instancié. Un service, quant à lui, traduit une tâche ou un processus d'affaires et tant qu'il est publié, il peut être invoqué et exploité.

1.2.2 La notion de couplage

La définition du terme « couplage » est assez simple dans le vocabulaire informatique : toute entité qui peut se connecter à une autre entité a un couplage. Les entités couplées peuvent former des dépendances entre elles (Erl, 2008). L'ambiguïté dans le terme « couplage » est introduite en ajoutant les termes « lâche » et « fort ». Dans les architectures orientées services, il existe des formes de couplage inévitables et qu'y sont présentes par nature (Figure 1.2). En effet, la logique du service est, par nature, fortement couplée (Heubès, 19 mars 2009) :

- à son implémentation physique (Figure 1.2 lien 1),
- aux technologies sur lesquelles s'appuie son implémentation (Figure 1.2 lien 2),
- aux services qui le composent (dans le cas de services composites) (Figure 1.2 lien 3),
- aux processus qui lui font appel (Figure 1.2 lien 4).

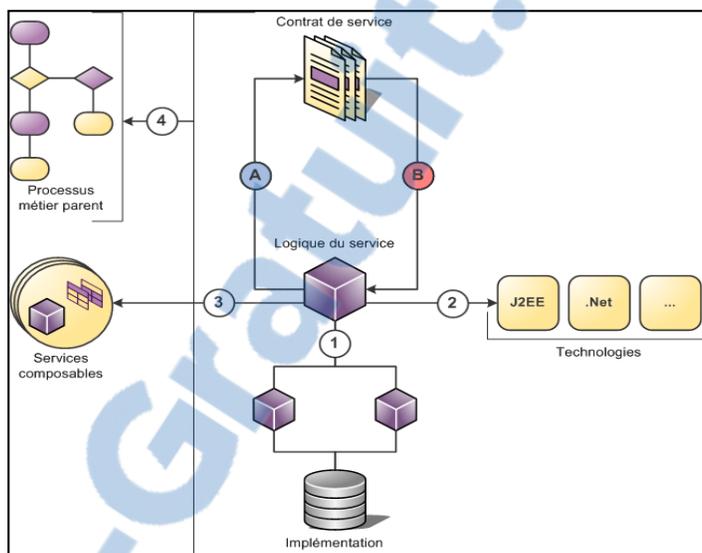


Figure 1.2 Les différents types de couplage dans SOA.
Tirée de Heubès (2009)

Le premier principe de base dans SOA est d'assurer un couplage lâche entre le service et son consommateur ainsi que son environnement. Ce principe permet d'assurer une évolution du service, sans impacts sur les consommateurs de ce dernier.

Le principe de couplage lâche est instauré par le biais du contrat de service, qui est l'élément de base autour duquel tourne la majorité des aspects conceptuels liés au couplage. On distingue deux types de couplage entre le service et son contrat :

- Couplage de la logique du service au contrat (Figure 1.2 lien A) : ce type de couplage illustre une approche recommandée pour la création de services, intitulée « *contract first* ». Cette approche consiste à définir en premier lieu le contrat, et à partir duquel, la logique du service.

- Couplage du contrat à la logique du service (Figure 1.2 lien B) : dans ce type de couplage, le contrat dérive de l'implémentation.

Afin d'assurer un couplage faible entre le service et le consommateur, il faut que ce dernier soit fortement couplé au contrat du service, et à lui seul. Le contrat, à son tour, doit être faiblement couplé à la logique de son service. Sinon, le consommateur du service va être lui aussi, par transitivité, être fortement couplé à la logique et l'environnement du service via le contrat.

1.2.3 Le principe d'abstraction

Le principe d'abstraction consiste à cacher l'information, liée à un service, qui n'est pas nécessairement requise lors de l'utilisation de ce dernier. Le service sera ainsi vu par le consommateur comme étant une boîte noire, cachant toute information d'ordre logique ou technique propre au service. Le degré d'abstraction d'un service doit être ajusté de façon à ne pas mettre en cause le potentiel de ce dernier à être réutilisé.

Il est nécessaire de faire la différence entre le concept d'abstraction et d'encapsulation. En effet, l'encapsulation fait référence à la logique, aux ressources, et aux informations faisant partie du service. Un service, dans son intégralité, est composé d'un contrat et de ce que le service encapsule. L'abstraction, quant à elle, fait référence à tout ce qui est encapsulé par le service et qui doit être accessible ou caché au consommateur externe de ce dernier. Autrement dit, l'abstraction détermine le degré avec lequel on va exposer, en public, ce qui est encapsulé par le service.

Le principe d'abstraction vient pour remédier au partage délibéré des informations qui portent sur les détails d'implémentation et de conception du service. Ainsi, en cas où le propriétaire de service décide de faire des modifications sur la partie encapsulée par ce dernier, le consommateur ne sera pas affecté par ces changements.

Ce principe d'abstraction vient pour confirmer le premier principe de SOA, à savoir la nécessité d'assurer un couplage faible entre le consommateur et le service. Ainsi, le consommateur n'aura accès qu'aux informations contenues dans le contrat du service (Erl, 2008).

1.2.4 La notion de réutilisabilité

La réutilisation est l'une des raisons d'être de SOA, elle constitue la brique de base de l'architecture orientée services. En technologie de l'information (TI), le terme réutilisabilité fait allusion à la capacité d'un logiciel ou d'un programme d'être utile pour plus qu'un seul objectif. Ceci, afin d'assurer un meilleur retour sur l'investissement (ROI) et augmenter l'agilité d'affaires de l'entreprise, assurant ainsi une réponse rapide aux futures exigences métiers de cette dernière. Un service est d'autant plus réutilisable qu'il est agnostique à son environnement. Un service est d'autant plus agnostique que sa logique est indépendante des processus d'affaires qui l'utilisent, de la technologie et de la plateforme applicative sur laquelle il est implémenté.

Un service peut être réutilisé de deux façons différentes (Erl, 2008). Il peut être invoqué, à plusieurs reprises, par le même consommateur, afin d'automatiser à chaque fois la même tâche métier; ou il peut être invoqué par différents consommateurs pour répondre à différentes tâches métiers. Bien que ces deux scénarios puissent être considérés comme des cas de réutilisation de service, la deuxième forme de réutilisation est la forme la plus recherchée par une architecture orientée services. Pour assurer cette deuxième forme de réutilisabilité, il faut mettre en application deux concepts de base : la centralisation de la logique métier et des contrats de services.

La centralisation de la logique métier vient pour assurer que le personnel TI n'ait qu'un seul point d'accès pour chaque ressource ou fonctionnalité offerte par le SI, à savoir le service responsable sur ces ressources (Figure 1.3, lien 1). Ceci peut être assuré par l'instauration d'un inventaire de services. Ainsi, dans le cas où une équipe de projet veut automatiser une

tâche métier, la première chose qu'elle doit faire est de parcourir cet inventaire et voir si cette tâche métier implique un service déjà existant. Dans ce cas, l'équipe de projet doit impérativement utiliser ce service. Dans le cas échéant, cette dernière (l'équipe de projet) doit créer un service qui répond aux exigences de la tâche métier qu'elle veut automatiser, tout en sachant que ce service peut être réutilisé par autrui.

La centralisation des contrats de services, quant à elle, vient pour assurer que les consommateurs de service n'accèdent aux services que par le biais de son contrat publié (Figure 1.3, lien 2). La combinaison de ces deux principes donne comme résultat un inventaire de services hautement standardisés et normalisés, qui vient à l'appui d'une réutilisabilité maximale et un faible couplage entre le service et ses consommateurs.

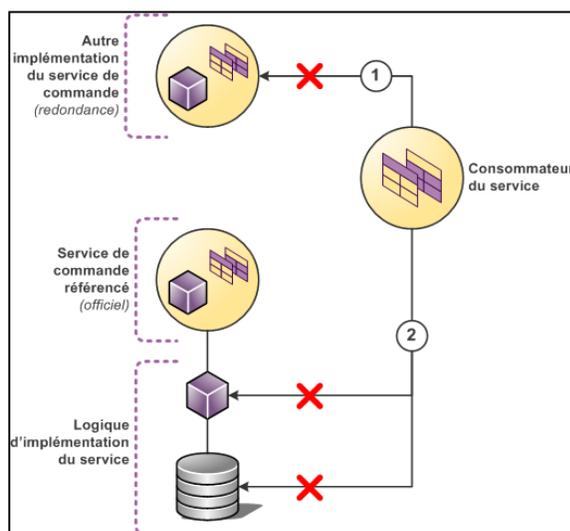


Figure 1.3 Centralisation des contrats de service et de la logique métier.
Tirée de Heubès (2009)

1.2.5 L'autonomie d'un service

L'autonomie d'un service représente sa capacité de s'autogouverner. Un service est considéré comme autonome s'il a la liberté et le contrôle de prendre ses propres décisions, sans avoir besoin d'une approbation ou intervention externe (Erl, 2008). L'autonomie d'un service

permet d'assurer une meilleure fiabilité, performance et prédictibilité dans l'exécution d'un service, surtout dans le cas de réutilisation ou composition de ce dernier. Erl (2008) définit quatre niveaux d'autonomie d'un service :

1. Autonomie de contrat de service : à ce niveau, les contrats de services sont alignés les uns avec les autres afin d'éviter un chevauchement des fonctionnalités offertes par chacun. Dans ce type d'autonomie, même si les fonctionnalités exprimées ne se chevauchent pas, les implémentations de service sous-jacentes peuvent encore se chevaucher.
2. Autonomie partagée : dans ce cas d'autonomie, la logique et les ressources qui forment le service sont partagées avec d'autrui. En effet, plusieurs entreprises trouvent inacceptable de réinventer la roue et créer tous les services à partir de zéro. Les entreprises ont souvent tendance à créer des services en encapsulant des systèmes ou des applications enracinées dans leur SI et qui sont accessibles par d'autres systèmes (Figure 1.4).

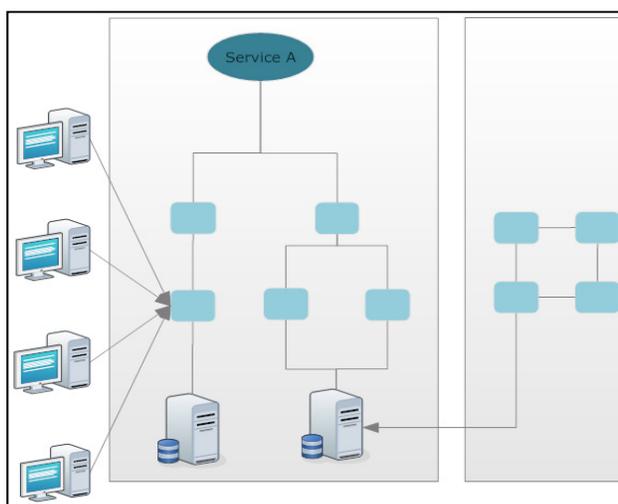


Figure 1.4 Cas d'une autonomie partagée.
Adaptée de Erl (2008, p. 306)

3. Autonomie de la logique du service : ce type d'autonomie illustre le cas où la logique métier du service est isolée, mais par contre les ressources utilisées par le service sont partagées par d'autrui.
4. Autonomie pure : c'est le niveau d'autonomie le plus élevé qu'on peut atteindre dans une architecture orientée services. Dans ce cas d'autonomie, les ressources ainsi que la

logique sous-jacente sont isolées et toutes les deux dédiées au service. On trouve, généralement, ce cas d'autonomie lors de la création de nouveaux services sans avoir recours à l'encapsulation des systèmes déjà existants.

L'autonomie est un facteur très important dans SOA. Cependant, l'application de ce principe, dans un environnement réel de l'entreprise, pose un grand défi. En effet, lors de la création de services, on n'est pas toujours épargné de l'encapsulation des anciens systèmes (*legacy systems*). En plus, fournir un environnement isolé pour chaque service peut engendrer des coûts énormes pour l'entreprise, surtout que dans certains cas, le service ne requiert pas vraiment un environnement isolé pour bien achever ses tâches.

1.2.6 La découvrabilité de services

Le concept de découvrabilité consiste à enrichir les services par des métadonnées afin de permettre à ces derniers d'être découverts et interprétés d'une manière plus efficace. Ce concept vient pour confirmer le principe de réutilisabilité défini précédemment. En effet, pour assurer qu'un service soit réutilisable il faut qu'il soit facilement découvert et interprété, tant par des humains que par des logiciels. L'atteinte de cet objectif nécessite une prévoyance et une compréhension solide du service et des fonctionnalités qu'il offre.

Il existe deux formes de découverte de services : découverte automatique (par un programme) ou manuelle par un acteur humain (un concepteur de systèmes ou développeur). La deuxième forme constitue la forme la plus commode de découverte de services (Figure 1.5). Dans le processus de découverte manuelle, l'acteur commence d'abord par parcourir une sorte de référentiel (*repository*) de services qui assure la localisation, la récupération et l'interprétation des métadonnées d'un service. L'acteur sélectionne, ensuite le service qui lui convient le mieux, et récupère son contrat à partir de l'inventaire de services. Le service sélectionné peut être pris tel qu'il l'est, ou étendu par les fonctionnalités requises par l'acteur.

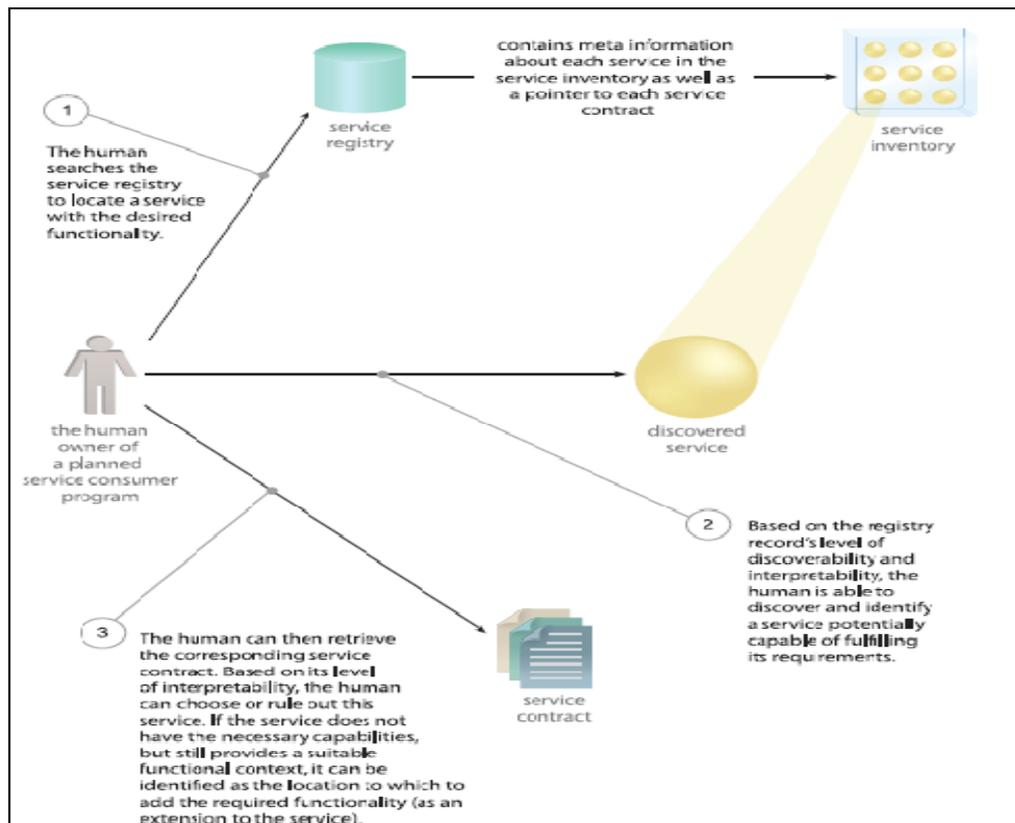


Figure 1.5 Scénario de découverte d'un service.
Tirée de Erl (2008, p. 366)

1.2.7 La composabilité de services

La composabilité est un concept fondamental dans la conception de logiciels. Ce concept permet de réduire la complexité de la logique des solutions informatiques et de les décomposer en modules plus simples. Ces modules peuvent être recomposés suivant d'autres configurations pour résoudre d'autres problèmes. Dans SOA, un service est considéré comme composable, s'il a l'aptitude d'être un membre effectif dans une composition d'autres services ou composants logiciel, peu importe la nature ou la complexité de cette composition. Le principe de composabilité vient pour soutenir le principe phare de SOA, à savoir la réutilisabilité, étant donné que la composabilité peut être vue comme une forme de réutilisation. Cependant, la présence de l'un de ces deux concepts n'induit pas la présence de l'autre. En effet d'une part, un service réutilisable n'est pas forcément composable : on ne

peut pas considérer une composition Point-à-Point entre deux services comme un vrai cas de composition, bien qu'il s'agisse d'un vrai cas de réutilisation. D'autre part, un service composable n'est pas forcément un service réutilisable : on peut avoir des services qui seront délivrés pour faire partie d'une composition bien spécifique, sans pour autant être réutilisés par d'autres compositions.

1.3 Avantages de l'architecture SOA

L'application des concepts expliqués précédemment permet de profiter de tous les avantages qu'on peut identifier autour d'une architecture orientée services (Erl, 2008), à savoir :

- **Assurer une interopérabilité intrinsèque** : l'adoption du concept de service permet aux différentes solutions du système TI d'échanger des données et des fonctionnalités entre elles, même s'ils sont développés en différents langages.
- **Assurer un alignement entre le métier et le TI** : SOA permet d'introduire plusieurs niveaux d'abstraction. En effet, elle permet d'introduire une nouvelle couche intitulée service qui encapsule les fonctionnalités techniques et permet une meilleure communication entre le métier et le système TI. Une telle approche facilite la traduction des représentations logiques du métier (processus et entités métiers) en représentation physique (les services).
- **Minimiser l'investissement initial et permettre la réutilisation** : dans une SOA on peut utiliser des composants déjà existants, quel que soit le langage de développement utilisé et quel que soit la plateforme sur laquelle ils tournent. Ceci permet un développement rapide de nouveaux services métier et un gain en termes de temps et d'argent.
- **Accroître l'agilité de l'organisation** : cette agilité est mesurée par l'efficacité par laquelle une entreprise peut répondre au changement (réactivité).
- **Minimiser les risques de défaillance** : étant donné qu'on réutilise des services pour en créer d'autres.

Les avantages cités ci-dessus font de SOA l'approche par excellence pour réduire les coûts TI du système actuel et avoir un important retour sur investissement (ROI). Malgré que les avantages de SOA, en terme d'investissement, sont bien connus et peuvent être facilement cernés (permettre l'ouverture à plusieurs marchés, réduire les coûts de maintenance, etc.), son ROI est un paramètre difficile à évaluer. Ceci peut être expliqué par le fait que SOA ne peut pas se réduire à un simple produit ou solution dont la valeur monétaire peut être appréciée facilement. En effet, SOA est considéré comme une approche qui fait partie d'un large processus transverse.

1.4 Limites de l'architecture SOA

Malgré les avantages cités ci-dessus, l'adoption d'une architecture orientée services peut poser de sérieux problèmes et entraîner de nouveaux défis, auxquels il faut faire attention. Ces problèmes ou limites seront détaillés dans la section actuelle.

Un sondage réalisé en 2006 sur 85 entreprises (Plouin et al., 2007) (Figure 1.6), montre que les principales limites d'une architecture orientée services sont :

- L'aspect sécurité qui est à surveiller surtout dans le cas où les services sont exposés via le Web.
- Les outils de développement utilisés pour la mise en œuvre d'une SOA expriment souvent un manque de productivité.
- Les plateformes d'intégration SOA souffrent d'un manque considérable de robustesse.

Comme toute technologie, SOA a ses avantages et ses limites, l'important est de bien profiter de ses apports et savoir pallier ses limites, pour cela il faut adopter une stratégie bien déterminée et savoir choisir la manière avec laquelle on va mettre en place cette architecture.

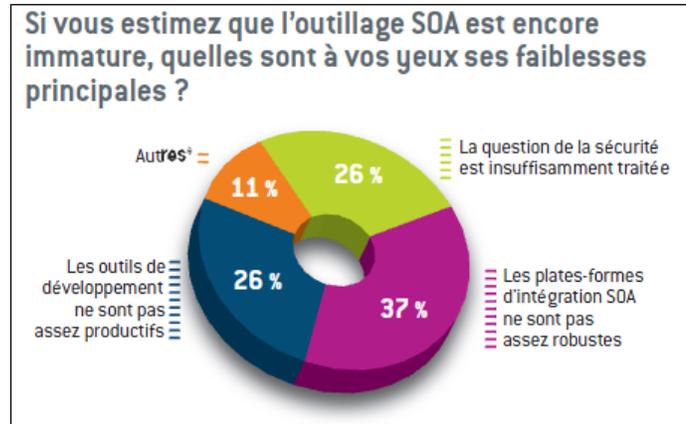


Figure 1.6 Les principales limites de SOA.
Tirée de Plouin (2007)

1.5 Web services : une réalisation de SOA

Il est important de souligner que SOA est une approche architecturale qui est totalement indépendante de n'importe quelle technologie ou plateforme. Les Web services constituent la plateforme technologique la plus commode pour la réalisation de SOA, ce qui a induit une confusion entre SOA et Web service.

Comme son nom l'indique, un web service est basé sur la technologie Web, qui contient d'ores et déjà les facilités nécessaires à la mise en place d'une SOA (routage, accès aux ressources réparties, sécurité, etc.). Ceci rend les Web services le choix par excellence pour la réalisation de SOA.

Plusieurs définitions sont avancées sur le compte d'un web service, le W3C (*World Wide Web Consortium*) définit les web services comme suit :

« Un système logiciel conçu pour supporter une interaction machine à machine interopérable sur un réseau. Un web service a une interface décrite en un format spécifique (WSDL). Les autres systèmes interagissent avec le web service conformément à sa description via des messages SOAP en utilisant le protocole HTTP. »

À partir de cette définition, nous pouvons décrire les différents protocoles et standards sur lesquels repose une SOA à base de Web services, à savoir :

- SOAP (*Simple Object Access Protocol*) pour l'échange de messages,
- WSDL (*Web Service Description Language*) pour la description des services Web, de leur localisation au sens internet (URI / URL), de leurs opérations, des messages, des types de données échangés et des protocoles utilisés,
- les annuaires UDDI (*Universal Description Discovery and Integration*) qui peuvent référencer des services Web.

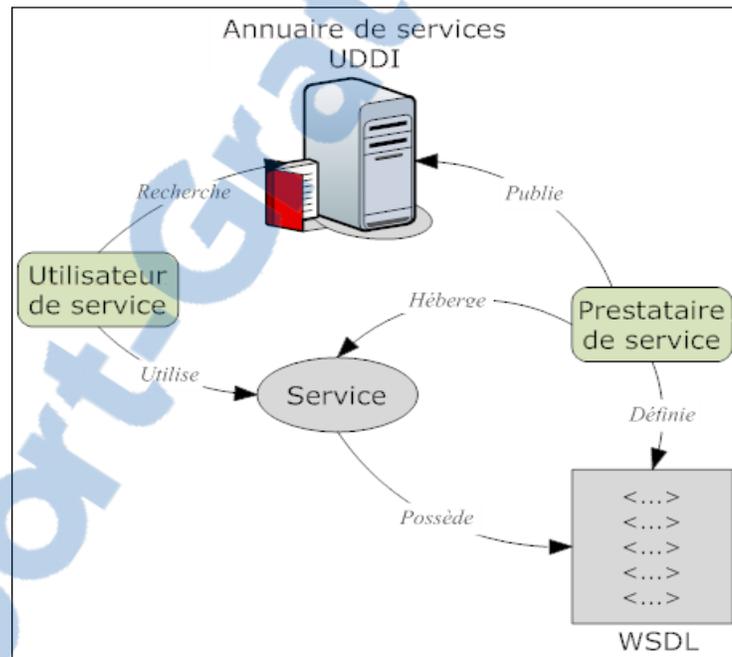


Figure 1.7 Scénario d'utilisation d'un web service.

La figure ci-dessus décrit le scénario de base qu'on trouve dans une SOA à base de Web service. Le fournisseur de service développe son web service qui est composé :

- d'une logique métier,
- et d'un contrat rédigé avec le langage WSDL qui décrit les fonctionnalités offertes par ce service.

Afin de faciliter la découverte de ce service, le fournisseur doit mettre le contrat développé dans un annuaire de service nommé UDDI. D'un autre côté, le client parcourt l'annuaire à la recherche d'un service qui lui convient, une fois qu'il le trouve, il se procure le contrat qui décrit la manière avec laquelle il doit interagir avec le service. La communication entre le client et le service est assurée via le protocole SOAP.

1.6 SOA et la modélisation des processus d'affaires

Un processus d'affaires d'une entreprise peut être défini comme un ensemble d'activités décrivant des tâches métiers qui doivent être menées selon un ordre bien déterminé et qui doivent respecter des règles bien précises. L'objectif majeur de SOA est d'assurer que le système d'information et toutes les ressources informatiques d'une entreprise sont en parfait alignement avec ses processus d'affaires ainsi que ses objectifs, à moyens et à long terme.

Pour atteindre cette fin d'alignement, il y a au moins deux défis à relever (Kapil et Matjaz, 2008) :

- Chaque entreprise a ses propres processus d'affaires et les systèmes d'information doivent être conçus autour de ces processus métiers, et non vice versa.
- Les processus d'affaires ne sont pas constants, ils changent avec le temps. Tout changement dans les processus de l'entreprise doit se refléter dans son SI (Système d'information). Cela nécessite des SI flexibles, qui peuvent être modifiés rapidement et efficacement.

SOA, avec tous les avantages qu'elle offre (flexibilité, agilité, interopérabilité, etc.), ne pourrait pas, toute seule, surmonter les défis mentionnés ci-dessus. L'atteinte de l'objectif d'alignement passe par l'assurance d'une bonne communication entre les analystes, responsables sur le métier de l'entreprise, et les développeurs, qui veillent sur l'infrastructure informatique de l'entreprise.

Afin d'assurer une vision commune des processus d'affaires, il est nécessaire de disposer d'un langage standard de communication entre les différents acteurs de l'entreprise, qui doit

être aussi bien compréhensible par un simple technicien que pour un responsable métier. La représentation graphique des processus d'affaires de l'entreprise peut s'avérer une solution à ce problème, à condition que cette représentation reflète bien le monde réel de l'entreprise. La qualité d'une représentation graphique d'un processus d'affaires est reflétée par la qualité de la notation utilisée pour modéliser ce processus. Il existe plusieurs notations standards pour la modélisation des processus, parmi lesquels on peut citer la notation *Unified Modeling Notation* (UMN).

1.6.1 Notation de modélisation unifiée (UMN)

Dans la modélisation des processus d'affaires, la notation la plus idéale et celle qui permet de présenter, dans une seule vue, tous les aspects pertinents d'un processus d'affaires (Ramudhin et al., 2007). UMN est conçue principalement pour atteindre cet objectif (Figure 1.8).

UMN permet de modéliser les processus d'affaires en suivant la trajectoire des différentes ressources impliquées dans ce processus. Ces dernières peuvent être classées en 4 catégories :

- **Le client** : présente l'entité qui va profiter du service fourni par le processus métier.
- **Les fournisseurs de services ou *providers*** : cette catégorie de ressources fait allusion aux membres du personnel de l'entreprise qui interviennent dans le processus.
- **Les informations** : sous forme électronique ou papier.
- **Le matériel** : logiciel, ordinateur, etc.

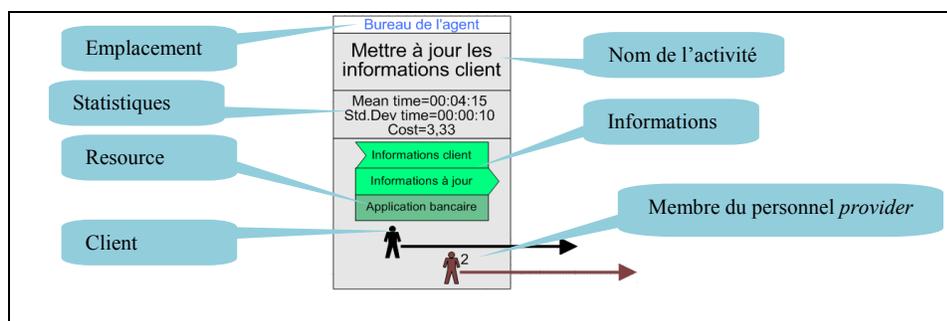


Figure 1.8 Exemple d'activité dans UMN.

1.6.2 Langage d'exécution des processus d'affaires BPEL

Certes, il est important pour une entreprise d'avoir une représentation graphique de ses processus métiers, ceci afin d'avoir une vue commune, que ce soit par les analystes métiers ou par le personnel TI. Toutefois, ces processus métiers doivent être, à un certain niveau, automatisés de façon à orchestrer les différentes ressources (services, systèmes informatiques, etc.) nécessaires à leur bon déroulement. Dans cette perspective on trouve le langage d'exécution des processus d'affaires BPEL (*Business Process Execution Language*).

L'objectif principal de BPEL est de normaliser les processus d'automatisation entre les services Web. En effet, BPEL est considéré comme la technologie clé dans les environnements où les fonctionnalités sont déjà ou seront exposées via des services Web (Matjaz, Benny et Poornachandra, 2006). Parmi les facilités que BPEL peut offrir, nous pouvons citer :

- Décrire la logique des processus d'affaires à travers la composition de services.
- Composer des processus larges et complexes en services et processus moins complexes.
- Corréler les échanges dans et à travers des processus d'affaires.

BPEL permet de décrire les processus d'affaires de deux façons différentes :

- **BPEL exécutable** : dans ce type de BPEL, on spécifie tous les détails nécessaires à l'exécution du processus d'affaires. Le BPEL exécutable peut être directement exécuté par un moteur d'orchestration.
- **BPEL abstrait (*Abstract BPEL*)** : dans cette variante de BPEL, on prend en considération juste l'échange de messages publics réalisé entre les différentes parties participantes au processus d'affaires. Ce type de BPEL ne comprend pas les détails internes du processus et ne peut pas être exécuté. Un BPEL de type abstrait permet principalement de décrire le comportement d'un processus, vu de l'extérieur.

1.7 Conclusion

Dans ce chapitre nous avons pu faire le tour des principaux aspects de base d'une architecture orientée services, tels que : la réutilisabilité, le couplage, l'abstraction, etc. Nous avons aussi essayé de décrire les différents aspects liés à SOA à savoir : les langages de modélisation et d'exécution des processus d'affaires.

L'architecture orientée services, devenue ces dernières années une architecture très répondeuse, nécessite plusieurs outils et méthodes pour sa mise en place. Aujourd'hui, plusieurs travaux de recherche se font dans le domaine des architectures orientées services. Ces travaux concernent principalement : la gouvernance, la qualité de service et la mise en place d'une SOA. Dans le chapitre suivant, nous allons nous intéresser principalement au problème de mise en place des architectures orientées services et particulièrement au problème de l'identification des services.

CHAPITRE 2

APPROCHES D'IDENTIFICATION DE SERVICES DANS L'ARCHITECTURE SOA

2.1 Introduction

Plusieurs travaux de recherche ont porté sur les techniques d'identification des services d'une architecture SOA. L'attention portée à cette problématique est due à la complexité de mise en place d'une telle architecture. Les principes de base de l'architecture SOA ainsi que les concepts correspondants ont été soulignés dans le chapitre « INTRODUCTION AUX ARCHITECTURES ORIENTÉES SERVICES ». Dans ce chapitre, nous allons essayer de mettre l'accent sur le cycle de vie général de SOA et montrer à quel niveau et selon quels critères peut se faire l'identification des services d'une architecture SOA.

2.2 Cycle de vie de SOA

Le cycle de vie de SOA est composé par une série de phases, similaires dans la forme et pas dans le fond, aux phases menées dans les projets de développement basés sur d'autres architectures. Le cycle de vie d'une architecture SOA est indifférent de l'approche suivie pour sa mise en place.

Un projet SOA doit toujours commencer par une phase d'analyse et de conception orientée services. Les services conçus, artefacts de ces deux premières phases, seront par la suite développés, testés et enfin déployés et gérés (Figure 2.1).

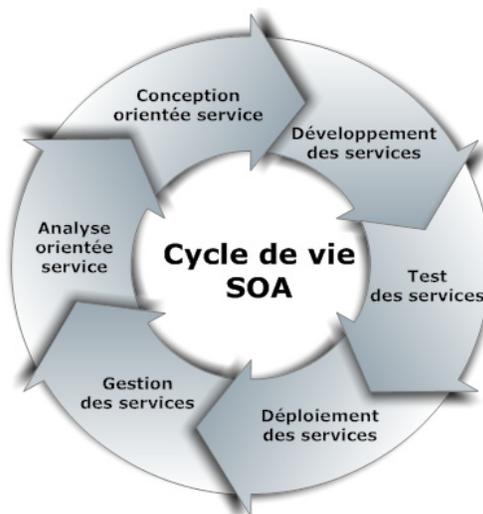


Figure 2.1 Cycle de vie SOA.
Adaptée de Erl (2005, p. 359)

Dans ce qui suit, nous allons décrire avec minutie les différentes phases du cycle de vie de SOA.

2.2.1 Phase d'analyse

Le principal objectif de la phase d'analyse est d'identifier les candidats de services et définir la logique encapsulée par chacun. C'est au niveau de cette phase que se situe, principalement, notre contribution. Il est à noter qu'à ce stade nous parlons de candidats de services et non pas de service proprement dit. En effet, à ce niveau nous ne pouvons pas être sûrs de la faisabilité des services identifiés, ceci ne peut être confirmé qu'au cours des prochaines étapes (Erl, 2005).

En plus de l'identification des services, la phase d'analyse a pour objectif aussi de spécifier les différentes couches de services qui vont former la future architecture. Cette phase s'avère très critique, étant donné qu'elle permet de donner une idée claire sur les traits de base de l'architecture à mettre en place. Face à ce problème, plusieurs auteurs ont mené des travaux afin d'assurer une identification jugée des services et des couches de services qui

conviennent le plus aux besoins de l'entreprise. Ces approches seront abordées avec plus de détail dans la section « 2.3 Approches d'identification de services ».

2.2.2 Phase de conception orientée service

Une fois que les candidats de services ont été bien identifiés, il est nécessaire de spécifier comment ces derniers vont être construits. La phase de conception orientée service permet de définir le processus par lequel on dérive la conception physique des services à partir des candidats de services et comment composer ces derniers pour implémenter les processus métiers (Erl, 2005). Parmi les principales activités menées dans la phase de conception orientée service phase, on peut citer :

- Définition des différentes composantes technologiques et standards industriels qui forment l'architecture orientée services à mettre en place. Les composantes fondamentales qu'on peut trouver dans une SOA sont : une représentation de données basée sur XML (*Extensible Markup Language*), des Web services basés sur des normes industrielles et une plateforme capable d'accueillir et traiter les données en XML ainsi que les services Web.
- Conception des interfaces des services identifiés, en spécifiant pour chacun, ses opérations, les messages échangés par ces opérations et le type de ces messages.
- Conception de la couche d'orchestration de services.

Étant donné les artefacts, cités ci-dessus, livrés lors de la phase de conception, cette phase s'avère aussi importante que la phase d'analyse dans la détermination de la qualité et de la longévité de l'architecture orientée services à mettre en place.

2.2.3 Phase de développement

La phase de développement a pour objectif de faire la construction, proprement dite, de l'architecture orientée services, tout en suivant les directives mises en place dans la phase

d'analyse et de conception. C'est au cours de cette phase que se fait le choix du langage de programmation, la plateforme SOA ainsi que l'environnement de développement.

2.2.4 Phase de test

Comme tout projet informatique, les services d'une SOA doivent être rigoureusement testés avant d'être déployés dans un environnement de production. En plus des défis ordinaires qu'on peut rencontrer, dans la phase de test, dans le cadre d'un projet informatique classique, SOA apporte ses propres défis, parmi lesquels nous pouvons citer :

- Tester des services qui ne possèdent pas d'interfaces utilisateurs.
- Pouvoir tester des services externes à l'organisation.
- S'assurer que le service ait un certain niveau de qualité pour qu'il puisse être réutilisé sans problèmes.
- Surveiller l'aspect sécurité dans le développement d'une stratégie de test.

La phase de test nécessite beaucoup d'efforts et implique plusieurs ressources. Pour cela, cette phase doit être menée par des personnes confirmées, qui maîtrisent autant l'environnement TI que le métier de l'entreprise. Il existe dans le marché des solutions qui viennent pour assister les testeurs SOA, parmi lesquelles, nous pouvons citer : Green Hat GH Tester, Mercury, Parasoft SOAtest, etc. (Harris, 2007).

2.2.5 Phase de déploiement

La phase de déploiement consiste à installer et configurer les composants distribués, les interfaces de services ainsi que les produits intergiciel (middleware) associés, sur des serveurs de production (Erl, 2005). Cette phase vient pour répondre à des questions, tel que : Comment les services vont être distribués? Comment les applications et les services existants vont être affectés par l'introduction des nouveaux services? Quelles sont les configurations requises pour assurer un certain niveau de sécurité?

2.2.6 Phase de gestion des services

Afin de profiter du plein potentiel de SOA, il est nécessaire d'assurer une bonne gestion des services formant le SI de l'entreprise. La phase de gestion des services a pour objectif de :

- Contrôler les versions des services et gérer les documents de description des fonctionnalités offertes par ces derniers.
- Gérer et garder la trace des messages échangés.
- Détecter les goulots d'étranglement.

2.3 Approches d'identification de services

Le service est considéré comme étant la clé de voûte d'une SOA. Partant de ce principe, la réussite d'un projet SOA passe par une identification précise des services les plus appropriés à l'entreprise et qui respecte les principes de base discutés au chapitre précédent (la réutilisabilité, la composabilité, etc.). Malgré qu'il existe un consensus sur ces principes, il n'existe pas une norme ou une méthodologie standard pour atteindre cette fin d'identification de service. Plusieurs approches ont été avancées pour guider le processus d'identification de services. Ces approches peuvent être comparées selon des critères techniques, économiques, métiers, etc (Boerner et Goeken, 2009; Klose et al., 2007), ou encore selon les intrants d'analyses utilisés et les différents types de services identifiés. En nous basant sur la nature des intrants d'analyses utilisés, nous pouvons distinguer trois types d'approches d'identification de services, à savoir : l'approche descendante (*Top-Down*), l'approche ascendante (*Bottom-up*) et l'approche mixte.

2.3.1 Approche descendante (*Top-Down*)

Étant donné que l'objectif principal d'une SOA, est de pouvoir aligner le métier de l'entreprise avec son SI, les partisans de l'approche *Top-Down* se basent sur l'analyse des processus d'affaires et des objectifs métiers afin d'identifier les services les plus appropriés à l'entreprise.

L'approche *Top-Down* consiste principalement à décomposer les processus d'affaires en tâches métier plus fines, et ce, pour permettre l'identification des services de granularités différentes. Ce type d'approches est mené généralement par des analystes d'affaires, qui doivent dans le cadre d'une approche *Top-Down* (Srikanth Inaganti et Gopala Krishna, 2007)

:

- **Activité 1 :** Comprendre et capturer les domaines fonctionnels et voir comment ces domaines interagissent entre eux.
- **Activité 2 :** Détailler les processus d'affaires ainsi que les points d'intersection entre eux (modélisation des processus d'affaires). Cette activité est assurée par la réalisation d'une cartographie complète de tous les processus métiers de l'entreprise.
- **Activité 3 :** Définir une hiérarchie des processus d'affaires en détaillant les sous-processus.
- **Activité 4 :** Faire la correspondance entre les processus actuels de l'entreprise ainsi que les systèmes informatiques et cerner les frontières de ces derniers. Ceci va permettre l'identification des applications affectées par SOA et la définition d'une feuille de route à suivre dans le processus de mise en place de cette architecture.
- **Activité 5 :** Identifier les candidats de service, de plus haut niveau, à partir des activités des processus d'affaires.
- **Activité 6 :** Définir la granularité des services, qui peut aller de la granularité la plus fine (le cas des services les plus élémentaires et les plus réutilisables), à la granularité la plus élevée (le cas des services d'orchestration).

La majorité des approches *Top-Down* trouvées dans la littérature s'articulent autour de la séquence d'activités mentionnée ci-dessus. Cependant, chaque approche a sa propre manière de mener ces activités et de les regrouper. Par exemple, Srikanth et Gopala Krishna (2007) définissent trois phases, qui conduisent à l'identification des services, à savoir : la phase de préparation, la phase d'analyse de services et la phase de catégorisation de services.

Dans la phase de préparation, les auteurs de cette approche ne font que l'analyse des domaines fonctionnels les plus liés au cœur du métier de l'entreprise, au niveau desquelles

les services identifiés auront un grand potentiel de réutilisabilité et leur intégration va apporter une grande valeur ajoutée. Dans la même phase de préparation, les auteurs complètent leur analyse par l'étude des processus d'affaires, en spécifiant :

- le degré de participation des systèmes informatiques dans les activités métiers (automatique, semi-automatique ou complètement manuelle),
- les unités organisationnelles,
- et les parties prenantes impliquées dans ces processus d'affaires, telles que les associés externes (par exemple les clients, les fournisseurs ou les prestataires de services) et les associés internes (par exemple les filiales, les sièges sociaux, etc.).

Srikanth et Gopala Krishna (2007), dans leur approche d'identification de services, ne font pas la correspondance entre les processus métiers et les éléments du SI de l'entreprise (activité 4) impliqués, à l'encontre de Suntae et al., (2008) qui proposent une approche, qui consiste principalement, à identifier les services à partir des objectifs métiers et desceller les différents scénarios suivis pour les atteindre. Ceci, tout en gardant la traçabilité entre les objectifs d'affaires, les changements possibles du métier et les différentes parties du système informatique y impliquées. Ainsi, on pourrait prédire les changements potentiels qui peuvent impacter le système informatique et lui assurer une meilleure agilité.

L'une des activités clés dans une approche *Top-Down* est la hiérarchisation des processus métiers. Différentes méthodologies peuvent être utilisées afin de réaliser cette hiérarchisation, parmi lesquelles nous pouvons citer : UML (*Unified Modeling Language*) qui offre une panoplie de modèles permettant d'organiser les processus métiers, ou encore OSSAD (*Office Support Systems Analysis and Design*) qui présente une méthode d'analyse des organisations par le biais des processus, en prenant appui sur les technologies de l'information et de la communication. L'approche OSSAD offre trois types de modèles assurant trois niveaux d'abstraction, à savoir : 1) le modèle abstrait (ou Modélisation graphique des Processus), 2) le modèle descriptif (ou Modélisation graphique des Procédures), et 3) le modèle prescriptif (ou Modélisation graphique des Instructions de travail) qui est un modèle facultatif (Dumas et Charbonnel, 1990).

L'adoption d'une approche descendante, pour l'identification des services, permet d'avoir non seulement une architecture orientée services complètement alignée aux besoins métiers de l'entreprise, mais aussi d'éviter la redondance de services. Cependant, cette démarche n'est pas épargnée de certains travers que certains essayent de les esquiver en complétant les approches *Top-Down* par d'autres techniques. (Senthil et al., 2008), par exemple utilisent les interfaces graphiques et les informations qu'elles contiennent pour optimiser le processus d'identification de services.

Malgré les avantages qu'elle apporte, la démarche descendante reste souvent très coûteuse et trop risquée, puisqu'elle nécessite une refonte complète ou partielle du SI de l'entreprise qui l'adopte.

2.3.2 Approche ascendante (*Bottom up*)

L'approche ascendante est réalisée principalement par des analystes TI, qui ont l'expertise et les compétences nécessaires qui leurs permettent de :

- analyser la plateforme applicative du SI actuel,
- desceller la logique métier et les fonctionnalités redondantes dans le SI,
- et les exposer sous forme de services, qui peuvent ou non être composés afin de participer à la construction des processus d'affaires.

Ainsi, la migration vers SOA sera réalisée sans avoir ignoré le vaste héritage logiciel et applicatif acquis par l'entreprise depuis sa création, ce qui réduit énormément le coût de mise en place d'une telle architecture.

Une approche ascendante commence par une évaluation rigoureuse de la plateforme TI de l'entreprise (Reddy et al., 2009) et la réalisation d'une cartographie complète de l'architecture applicative, qui décrit les différentes fonctions offertes par le SI actuel. Dans cette cartographie, les logiciels et les programmes du SI peuvent être classés selon plusieurs critères. Sneed (2006), dans son approche ascendante d'identification de service, classe les

actifs du SI, selon leur degré de dépendance vis-à-vis de l'environnement TI. Il définit ainsi trois catégories de programmes :

- les programmes qui sont complètement indépendants de leur environnement,
- les programmes partiellement dépendants de leur environnement,
- et les programmes qui sont totalement dépendants de leur environnement.

Sneed (2006) propose aussi trois alternatives pour procéder à la migration du SI actuel vers une architecture orientée services : La première consiste à analyser le code par ingénierie inverse (*reverse engineer*) et re-implémenter le code dans un autre langage, la deuxième consiste à envelopper le code exécutable et y accéder via une interface, la troisième et dernière alternative consiste à transformer le code source pour qu'il puisse être traduit en service. Cette dernière alternative est recommandée quand le code est suffisamment indépendant de son environnement et le coût de re-implémentation est très élevé.

L'adoption d'une approche ascendante dans la mise en place d'une SOA n'assure pas systématiquement l'alignement du métier avec le SI. En effet, cette approche présuppose que le SI actuel est, d'ores et déjà, aligné avec le métier de l'entreprise et répond aux besoins de cette dernière, ce qui n'est pas toujours le cas. En plus, en suivant cette approche, les bénéfices de mise en place d'une SOA ne seront perçus, qu'une fois les services auront été rendus exploitables au sein des processus métiers, ce qui rend très difficile la justification de l'investissement auprès des décideurs (Heubès, 16 août 2007).

2.3.3 Approche mixte

Les approches mentionnées précédemment (*Top-down* et *Bottom-up*) peuvent être combinées dans une approche intitulée « mixte ». Cette approche préconise de mener à la fois une analyse ascendante et une analyse descendante et faire une corrélation entre les services identifiés par chacune de ces approches.

Une approche mixte d'identification de service peut être résumée en quatre étapes (Srikanth et al., 2007). Dans une première étape, les analystes du métier commencent par analyser les processus d'affaires de l'entreprise et y recenser les activités. Cette étape nécessite d'avoir des processus d'affaires documentés, offrant un certain niveau de détail. Dans une deuxième étape, menée en parallèle avec la première, les analystes TI commencent à analyser les applications au niveau du SI et identifier les fonctionnalités qui peuvent être transformées en services. Dans une troisième étape, les activités identifiées au niveau métier et les services recensés au niveau TI, vont être confrontés afin de pouvoir trouver un lien de corrélation entre eux. Ce lien peut être un lien direct ou un lien de composition, c'est-à-dire, un service identifié à partir du *legacy-system* peut être directement associé à une activité métier ou dans le cas contraire, combiné avec un ou plusieurs autres services, recensé à partir de la couche TI, afin de pouvoir bien mener l'activité du processus d'affaires. La quatrième et dernière étape consiste à raffiner les processus métier et définir plus de sous-processus, ceci afin d'essayer d'atteindre, idéalement, un niveau de corrélation un à un entre les services TI et les activités métiers.

Certes, l'adoption d'une approche mixte permet de combiner les avantages de l'approche ascendante et descendante, cependant, cela ne veut pas dire que cette combinaison va éviter les travers de ces deux approches. En effet, la recherche d'un lien de corrélation entre les activités métiers et les services identifiés au niveau SI présente une tâche très critique. Elle nécessite non seulement la compréhension du métier de l'entreprise, mais aussi la maîtrise de son SI. Dans une approche mixte, la réussite ou non de la mise en œuvre d'une SOA dépend principalement de cette étape d'accostage. La Figure 2.2 schématise les différentes approches d'identifications de services présentées dans cette partie du rapport.

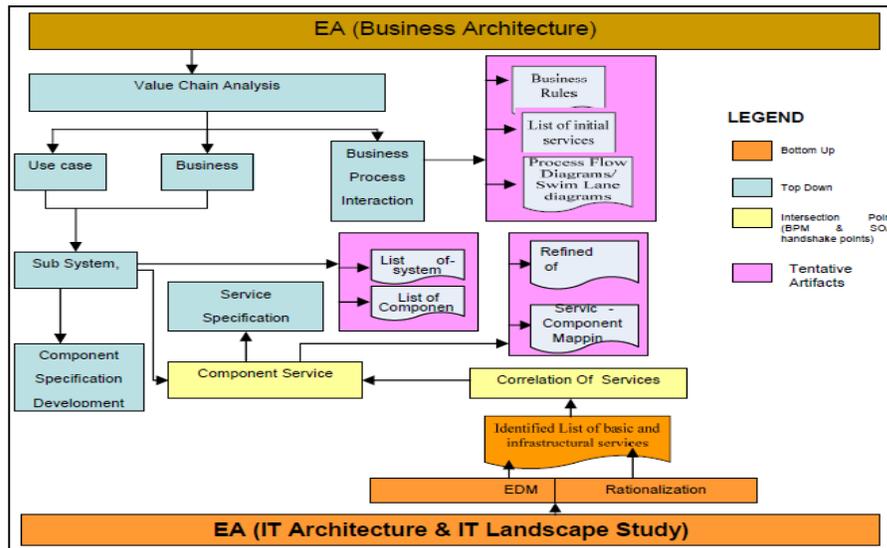


Figure 2.2 Les différentes approches d'identification de services.
Tirée de Srikanth Inaganti (2007, p. 4)

Le tableau ci-dessous présente une étude comparative réalisée sur les différentes approches d'identification de services rencontrées dans la littérature.

Tableau 2.1 Étude comparative des approches d'identification de services

Approches	Inputs d'analyse	Services identifiés	Avantages	Inconvénients
Descendante	- Processus d'affaires. - Domaines et objectifs métiers.	Métiers	-Alignement SI / Métier. -Éviter la redondance des services.	-Coûteuse -Risquée
Ascendante	Applications informatiques du SI actuel.	Applications	-Coût réduit. -Cartographie du SI.	-Alignement SI/Métier non garanti. -Investissement non justifié au départ.
Mixte	Couche applicative et métier.	-Métiers -Applications	Combinaison des avantages des approches descendantes et ascendantes	Combinaison des inconvénients des approches descendantes et ascendantes

2.4 Analyse critique des différentes approches d'identification de services

Après avoir fait le tour des différentes démarches d'identification de services trouvées dans la littérature, nous remarquons que les approches les plus répondues et les travaux les plus récents sont ceux basés sur l'analyse des processus d'affaires de l'entreprise. Cependant, les approches telles que l'approche descendante et l'approche mixte sont le plus souvent manuelles et exigent une intervention humaine dans l'analyse et la catégorisation des processus d'affaires. La non-automatisation de ces approches est due principalement à la nature des informations liées aux processus d'affaires, sujets d'analyses. En effet, dans une entreprise, on trouve généralement l'information concernant les processus métier, sous forme de graphiques ou de rapports écrits en langage naturel. Ces formes d'information ne facilitent pas l'interprétation automatique de ces processus d'affaires et rendent la participation humaine indispensable, surtout dans la phase d'analyse métier.

Pour remédier à cette situation, plusieurs initiatives ont été mises en branle. Une des solutions proposées consiste à faire appel au concept d'ontologie pour représenter, organiser et raisonner à travers un ensemble complexe de connaissances métiers (DongSu et al., 2008; Yousef et al., 2009). Une deuxième solution consiste à stocker les informations liées aux processus d'affaires sous un format exploitable et bien structuré, tel que XML, qui est facile à analyser par n'importe quel outil informatique. Une telle solution est adoptée par Dwivedi et Kulkarni (2008) qui utilisent, dans leur approche d'identification de service, des cartographies de processus d'affaires à base de UML, représentées en XMI (*XML Metadata Interchange*). XMI étant un format d'échanges d'informations de métadonnées UML basé sur XML. Ce format de persistance (XML) des informations métiers a permis aux auteurs de cette approche d'exploiter automatiquement les processus d'affaires et collecter les statistiques qui leurs sont liées.

En plus de la non-automatisation, la majorité des approches d'identification de services sont philosophiques et non systématiques, ce qui rend difficile leur application dans le monde réel. Arsanjani et al. (2008); Boerner et Goeken (2009); Erradi, et al. (2006); Huysmans et al.

(2007); Klose et al. (2007); Papazoglou et van den Heuvel, 2006; Srikanth et al (2007); Van Nuffel (2007) présentent un ensemble de lignes directrices et de bonnes pratiques à suivre, cependant ils ne décrivent pas réellement comment l'identification des services doit être menée. À l'encontre de Azevedo et al. (2009), qui proposent une approche systématique basée sur un ensemble d'heuristiques bien détaillées à suivre dans le processus d'identification des services, et ce, en se basant sur une analyse :

- sémantique des éléments de processus (règles métiers, besoins métiers);
- syntaxique des modèles de processus (*structural pattern*).

Afin de pouvoir exploiter le plein potentiel de SOA, certaines démarches d'identification de services adoptent une gouvernance SOA (Boerner et Goeken, 2009; Dan et al., 2008; Strnagl, 2007), cette gouvernance a pour objectif de traiter les problèmes liés à la mise en œuvre d'une SOA, et ce, en assurant une bonne intégration des ressources humaines, des processus et des données. En effet, une gouvernance SOA permet de :

1. éviter la multiplication incontrôlée des services;
2. gérer le cycle de vie de ces services;
3. assurer leur alignement avec la stratégie de l'entreprise.

Il existe dans le marché une panoplie d'outils, dits de gouvernance SOA, telle que *IBM Rational Method Composer*, *Oracle SOA Governance 11g*, etc. Ces outils permettent de faciliter une telle gouvernance et appliquer ses règles.

2.5 Conclusion

Le choix d'une approche d'identification de services (descendante, ascendante ou mixte) doit être fait en prenant en considération l'état de l'existant de l'entreprise : niveau d'expertise des analystes métier et TI, présence ou non d'une cartographie de processus d'affaires, complexité du SI actuelle, etc.

L'essentiel dans une telle approche est de bien respecter les principes de base de SOA, et bien répondre aux besoins métiers de l'entreprise.

CHAPITRE 3

PROPOSITION D'UNE APPROCHE D'IDENTIFICATION DE SERVICES

3.1 Introduction

La réussite d'un projet SOA passe par une identification précise et justifiée des services les plus appropriés à l'entreprise. La principale question qui se pose autour de cette problématique d'identification de services est : selon quels critères les services d'une architecture SOA seront identifiés?

Plusieurs travaux de recherche se sont intéressés au problème d'identification de services (Arsanjani et al., 2008; Boerner et Goeken, 2009; Huysmans, Verelst et Mannaert, 2007; Klose, Knackstedt et Beverungen, 2007; Srikanth et al., 2007; Van Nuffel, 2007). La plupart de ces travaux ont abouti à une démarche pragmatique dont l'automatisation est difficile, voire impossible. Cependant, l'utilisation de plus en plus étendue de l'architecture SOA, fait qu'il n'est plus envisageable aujourd'hui de continuer à appliquer de telles démarches.

Dans le présent chapitre nous allons d'abord proposer une approche formelle qui consiste en un ensemble de lignes directrices assurant une identification justifiée des services les plus appropriés au métier de l'entreprise, puis nous essayerons de l'automatiser le maximum des étapes de cette démarche.

Pour décrire notre démarche, nous procéderons de la manière suivante : nous commencerons par une présentation générale de l'approche proposée qui sera, ensuite, le sujet d'une description plus détaillée, et ce, en présentant avec minutie les différentes étapes amenant à une identification des services les plus commodes à une entreprise.

3.2 Principes de l'approche proposée

L'approche que nous proposons dans ce chapitre pour l'identification des services rentre dans le cadre des approches descendantes dont les principales entrées d'analyse sont : les processus d'affaires de l'entreprise. Les services identifiés par cette approche auront une granularité allant de la granularité la plus fine (services d'application) à la granularité la plus élevée (services d'orchestration) (Figure 3.1). Ceci permet d'avoir une architecture orientée services multicouche. Les services ayant la granularité la plus fine sont les services les plus réutilisables. Ainsi dans une architecture SOA multicouche, ayant la configuration illustrée par la Figure 3.1, les services d'application et les services entités doivent assurer un haut degré de réutilisabilité, contrairement aux services d'orchestration et aux services centrés-tâche, pour lesquels la réutilisabilité n'est pas un facteur important. Ces deux derniers types de services s'intéressent plutôt à la logique d'affaires encapsulée par les processus métier de l'entreprise.

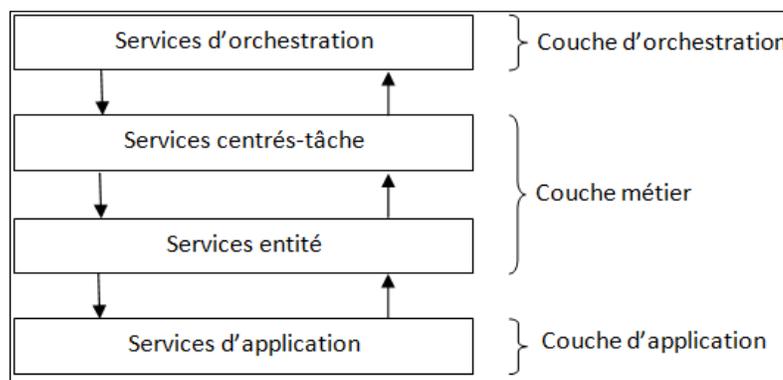


Figure 3.1 Types de services identifiés.

La présente approche permet d'identifier quatre types de services (Figure 3.1), à savoir :

- **Les services d'application** : Ces services représentent un groupe de services qui expriment des fonctionnalités réutilisables liées aux technologies. Ils décrivent l'environnement applicatif de l'entreprise et fournissent des fonctionnalités réutilisables liées à des transformations et des traitements d'ordre technique. Les services de type

Utility (Utilité) ou *Wrapper* (Enveloppe) présentent deux exemples typiques pour de tels services.

- **Les services entités** : Ces services ont pour rôle d'encapsuler et gérer les entités métiers de l'entreprise (client, compte bancaire, commande, etc.). Ce type de services offre un haut degré de réutilisabilité et peut être composé facilement par d'autres services de plus haut niveau (les services d'orchestration ou les services centrés-tâche).
- **Les services centrés-tâche (*Task-centric services*)** : Ces services encapsulent une logique métier spécifique à une tâche métier à gros-grains. Ce type de services est utilisé souvent quand les analystes jugent qu'une tâche métier, faisant appel à plusieurs services, est assez autonome qu'il est plus commode de l'encapsuler dans un autre service, plutôt que de la diviser en plusieurs tâches plus fines et les intégrer dans la couche d'orchestration.
- **Les services d'orchestration** : Ces services ont pour objectif d'assurer que l'architecture orientée services, mise en place, est en parfait alignement avec les processus d'affaires de l'entreprise. Les services d'orchestration encapsulent tout un processus d'affaires qui peut contenir des services centrés-tâche, des services entité, ou des services d'application.

En tant qu'approche descendante, la démarche que nous proposons se base sur l'analyse des processus d'affaires pour pouvoir identifier les services les plus appropriés à une entreprise. La modélisation graphique des processus d'affaires s'avère une solution idéale pour la représentation des besoins métiers d'un projet SOA (Van Nuffel, 2007). Cependant, Il existe plusieurs langages de modélisation BPM (*Business Process Modeling*) qui permettent cette représentation (BPMN, EPC, etc.). Toutefois, le choix du langage BPM le plus approprié, qui va servir comme un point de départ à une approche d'identification de services, est une tâche très critique. Van Nuffel (2007) soulève deux questions importantes sur le choix de la notation BPM la plus appropriée à une démarche d'identification de service :

- i. Quel langage BPM (*Business Process Modeling*) est le mieux placé pour agir comme une base aux lignes directrices d'une démarche d'identification de services?

- ii. Est-ce que le langage de modélisation choisi contient toutes les informations nécessaires à ces lignes directrices? Ou est-il nécessaire d'intégrer d'autres techniques et/ou artefacts?

La présente approche utilise comme intrants d'analyse, des processus d'affaires modélisés en UMN. Comme nous l'avons présenté dans le chapitre « Introduction aux architectures orientées services », et afin de répondre aux questions mentionnées ci-dessus et soulevées par Van Nuffel (2007) : UMN permet de présenter, dans une seule vue, tous les aspects pertinents d'un processus d'affaires (Ramudhin et al., 2007). En plus, UMN peut être intégré dans une structure logicielle qui persiste les données liées aux processus d'affaires dans une base de données centralisée, ce qui facilite l'exploitation et l'analyse statistique des informations et des activités liées à ces processus métiers. Ceci est très utile pour l'automatisation de certaines étapes de l'approche proposée. En effet, dans cette dernière, l'identification des services est réalisée, comme dans Dwivedi et Kulkarni (2008), à base d'analyses statistiques faites sur les processus d'affaires. Ces analyses, s'intéressent principalement aux activités métier, aux informations qu'elles utilisent et à la façon dont ces informations sont manipulées (mode d'utilisation, fréquence de présence dans une activité, etc.). Le terme « information » désigne ici tous les supports d'information que ce soit papier, électronique ou autre (des parties de documents ou des documents entiers, des rapports, des fenêtres d'écrans, etc.) qui sont nécessaires à l'exécution d'une activité.

La présente approche ressemble, de point de vue catégorisation du métier de l'entreprise en entités, à celle proposée par Bonnet (2005), sauf que ce dernier ne respecte pas, dans sa catégorisation, le principe phare de SOA, à savoir la réutilisabilité. En plus, malgré que Bonnet (2005) décrit en détail les mécanismes les plus importants à suivre dans l'implémentation technique d'une SOA, il ne montre pas comment identifier les services d'application (qui décrivent l'environnement applicatif de l'entreprise) et se limite aux services métier et d'orchestration.

L'approche que nous proposons rentre dans le cadre des initiatives d'automatisation du processus d'identification de services. Le degré d'automatisation de l'approche proposée est lié principalement au degré de flexibilité offert par le langage BPM avec lequel sont modélisés les processus d'affaires. La présente approche peut être appliquée en utilisant n'importe quel langage BPM, elle n'est pas limitée à UMN. Toutefois, il y a des critères auxquels le langage BPM choisi doit répondre. Ces critères vont être fixés tout au long de ce chapitre, durant la présentation des lignes directrices de l'approche proposée.

Cette approche formelle est similaire, de point de vue analyse structurelle des processus d'affaires (*structural patterns'analysis*), à l'approche d'Azevedo et al. (2009), à la différence que ce type d'analyse nous sert dans l'identification d'un certain type de services et non de tous. En effet, l'analyse structurelle nous est utile plutôt dans la phase d'élaboration de la couche d'orchestration, plus précisément dans la transformation des modèles métiers en code exécutable (BPEL) (Chun et al., 2008). Ceci va être abordé avec plus de détails au cours de la prochaine section « 3.3 Description détaillée de la démarche proposée ».

À ce stade d'analyse un peu prématuré, nous ne pouvons pas nous assurer que tous les principes de base de SOA seront respectés (découverte, réutilisabilité, composabilité, etc.). En effet, il existe des principes liés à des phases très avancées de mise en place d'une architecture orientée services (définition du contrat de service, développement du service, déploiement du service, etc.). Dans la présente approche, nous allons nous assurer juste de l'application des trois principes phares de SOA à savoir : la réutilisabilité, l'autonomie et la découvrabilité. L'application du principe de découvrabilité est assurée en décrivant avec minutie les services identifiés et les fonctionnalités qu'ils offrent. Le choix de noms significatifs pour ces services rentre aussi dans l'application du principe de découvrabilité. Les principes de réutilisabilité et d'autonomie, quant à eux, découlent directement de la logique encapsulée par le service, et de la définition de ses frontières. Pour cette raison, au cours de cette approche d'identification nous allons nous concentrer plus sur les principes de réutilisabilité et autonomie que sur le principe de découvrabilité.

Les principes d'autonomie et de découvrabilité doivent être respectés par tous les services, indépendamment de la couche à laquelle ils appartiennent, tandis que la réutilisabilité est un critère indispensable pour les services d'application et entité, étant donné que les services centrés-tâches et les services d'orchestration sont plutôt orientés logique d'affaires.

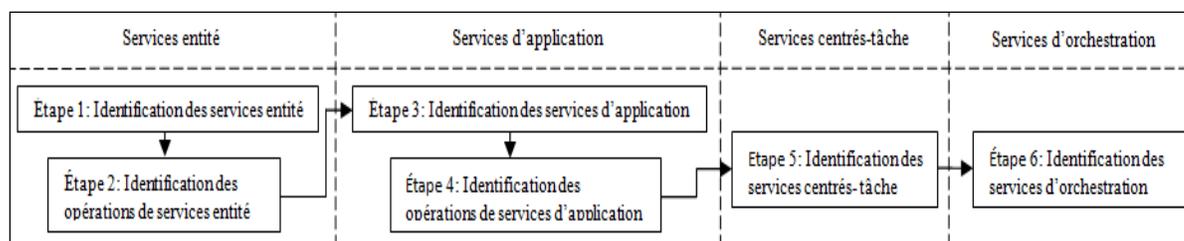


Figure 3.2 Les six étapes d'identification des services les plus appropriés.

L'approche d'identification de services que nous proposons est constituée de six étapes (Figure 3.2). Nous commençons d'abord par l'identification des services entité ainsi que leurs opérations, ensuite nous procédons à l'identification des services d'application et leurs opérations, pour enfin recenser les services centrés-tâche et les services d'orchestration. Ces derniers (les services d'orchestration) vont constituer la dernière couche de l'architecture orientée services, dont le rôle est de faire le lien entre le métier de l'entreprise et son système TI. Ces étapes seront expliquées avec minutie dans la prochaine section.

3.3 Description détaillée de la démarche proposée

Dans cette partie du mémoire, nous allons présenter en détail, les étapes qui vont nous conduire à une identification justifiée des services les plus commodes pour une entreprise. Tout au long de cette section, nous allons recenser les critères que doit vérifier un langage BPM pour être utilisé par cette approche.

Nous rappelons que les six étapes sur lesquelles se base la démarche proposée pour l'identification des services sont l'identification des :

- services entité,
- opérations des services entités,
- services d'application,
- opérations des services d'application,
- services centrés-tâche
- et services d'orchestration.

3.3.1 Identification des services entités

La présente approche commence par l'identification des services entités. Cette identification est faite à partir des informations, dites « clés » dans les processus d'affaires de l'entreprise. Une information clé est une information à grande valeur ajoutée pour un processus d'affaires. Elle est généralement très sollicitée au sein de l'entreprise. Son utilisation n'est pas restreinte à un seul processus d'affaires, elle peut être utilisée par plusieurs processus et dans plusieurs domaines métiers de l'entreprise. Étant donné l'exemple d'un processus d'affaires d'achat, parmi les informations clés qu'on peut trouver, nous pouvons citer : « bon de commande », « confirmation de commande », « description du produit », etc.

Les informations clés peuvent être regroupées sémantiquement afin de recenser les différentes entités métiers, sur lesquelles se base les processus d'affaires de l'entreprise. Nous attribuons, pour chaque entité identifiée, un candidat de service qui va la gérer. Reprenons le même exemple du processus d'achat, les informations clés « bon de commande » et « ordre de commande » peuvent être regroupées pour former l'entité métier « commande », à partir de laquelle nous pouvons identifier un service entité ayant le même nom « commande ».

Dans le cadre d'identification de services entité, deux principes doivent être assurés : la réutilisabilité et l'autonomie. Ces deux principes sont indispensables pour un tel type de service. En effet, les services entités, par définition, font partie des services les plus réutilisés

dans une architecture SOA. Pour être réutilisables, ces derniers doivent être autonomes et indépendants des autres services. À ce niveau, nous ne pouvons garantir que l'autonomie fonctionnelle (voir « 1.2.5 L'autonomie »), les autres types d'autonomie ne peuvent être assurés qu'au cours des phases de développement et déploiement du cycle de vie de SOA.

Étant donné que l'identification des services entité est faite à partir des informations clés, ces dernières doivent assurer les deux principes discutés ci-dessus (la réutilisabilité et l'autonomie). Le principe de réutilisabilité intervient principalement dans la phase de sélection des informations clés. L'autonomie quant à elle, intervient dans la phase de regroupement de ces informations, afin de sceller les différentes entités métier, qui doivent être indépendantes l'une de l'autre. La question qui se pose à ce niveau est : quelles sont les informations qui peuvent être considérées comme « clés »?

Partant du principe de réutilisabilité que doit respecter les candidats de services, à identifier à ce niveau, une information clé doit respecter au moins l'un des critères suivants :

- elle est très sollicitée dans le métier de l'entreprise,
- elle est fortement corrélée à une activité métier très sollicitée.

Le degré de sollicitation ainsi que le degré de corrélation entre une activité et une information sont déterminés à partir d'une analyse statistique des processus d'affaires. Durant cette analyse, des seuils seront fixés, à partir desquels nous pouvons déterminer, si une information sera considérée comme fortement corrélée à une activité et si une activité ou une information sera considérée comme très sollicitée. L'étape d'identification des services entités est subdivisée en trois étapes :

- Sélectionner les activités sujettes d'études.
- Recenser à partir de ces activités les informations les plus récurrentes, ainsi que les activités les plus sollicitées pour pouvoir recenser toutes les informations clés.
- Regrouper les informations clés afin d'identifier les services entités.

3.3.1.1 Sélection des activités sujettes d'étude

Dans les processus d'affaires, on peut trouver trois types d'activités : manuelles, automatiques (sans aucune intervention humaine) ou semi-automatiques. Dans le cadre de la présente approche, nous n'allons nous intéresser qu'aux activités automatiques ou semi-automatiques, qui sont supportées partiellement ou complètement par des systèmes informatiques. En effet, une activité manuelle telle que la signature d'un formulaire ou l'explication des termes d'un contrat pour un client ne peut pas être considérée comme candidat de service ni comme une opération de service. Bien évidemment, les activités automatiques ou semi-automatiques doivent impérativement utiliser des ressources informatiques, à partir desquelles nous pouvons recenser des services (exemple : les logiciels, les sites Web, etc.). De toute évidence, ces ressources doivent faire partie des propriétés de la société, ayant ainsi le droit de les modifier et recenser la logique qui sera encapsulée par les services à mettre en place.

Le nombre d'activités qu'on peut trouver dans les processus d'affaires d'une entreprise peut atteindre, généralement, l'ordre de plusieurs centaines d'activités. Ainsi, la tâche de sélection des activités sujettes de l'étude (automatiques et semi-automatiques) s'avère très fastidieuse, surtout si elle se fait manuellement, d'où la nécessité d'automatiser cette tâche. Cette automatisation dépend principalement des facilités offertes par le langage BPM avec lequel sont modélisés les processus d'affaires. Un tel langage doit être intégré à une structure logicielle qui permet de persister les données liées au processus d'affaires sous un format facilement exploitable (XML, base de données, fichier bien structuré, etc.). En plus, ce langage doit permettre de faire le lien entre les activités métiers et les ressources-y impliquées, ayant ainsi un moyen pour déterminer le degré d'automatisation de ces activités et cerner les ressources à partir desquelles le service va être recensé.

3.3.1.2 Identification des informations clés

Une fois que les activités sujettes d'études ont été recensées, nous allons maintenant commencer l'identification des informations clés dans les processus métier de l'entreprise. Ces informations vont par la suite être regroupées afin de pouvoir identifier les candidats de services entités. Comme nous l'avons mentionné précédemment (3.3.1 Identification des services entités), les informations clés peuvent être identifiées à partir des informations et des activités les plus récurrentes dans les processus d'affaires de l'entreprise. Dans les deux sections qui vont suivre, nous allons montrer, selon quels critères, se fait la sélection des informations et des activités les plus sollicitées.

3.3.1.3 Sélection des informations les plus récurrentes

Au cours de cette sous-étape, nous allons parcourir les informations utilisées par les activités métiers sélectionnées dans la section « 3.3.1.1 Sélection des activités sujettes d'étude » et recenser celles qui sont les plus récurrentes. Ces informations constituent le premier lot d'informations clés à cerner à partir des processus d'affaires de l'entreprise. À ce niveau, deux questions peuvent se poser : Pourquoi l'étude porte, juste, sur les informations utilisées par les activités automatiques et semi-automatiques et ne pas prendre en considération les informations utilisées par les activités manuelles? Selon quels critères pouvons-nous dire qu'une information est très récurrente dans les processus d'affaires de l'entreprise?

Pour répondre à la première question, les informations clés que nous allons recenser vont servir par la suite pour l'identification des services entités. Ses services vont être manipulés par des systèmes informatiques (logiciel, site Web, etc.) et n'ont pas manuellement par des êtres humains. Pour cela, au cours de l'identification des services entité, il faut cerner l'analyse, juste, sur les informations manipulées par des activités automatiques ou semi-automatiques qui utilisent des systèmes ou des outils informatiques.

Concernant la question liée au critère de sélection des informations les plus récurrentes, une des solutions consiste à fixer un seuil à partir duquel nous pouvons juger qu'une information est récurrente ou non. L'équation ci-dessous décrit le seuil que nous proposons pour cette fin :

$$\text{Seuil} = [\text{moyenne} + (\text{écart type}/Y)]. \quad (3.1)$$

Où $[X]$ est l'opérateur qui arrondi X à la valeur entière la plus proche et Y un entier qui détermine le degré de réutilisation des informations recensées. Une valeur égale à 1 de Y permet de recenser les informations qui sont très récurrentes dans les processus métier de l'entreprise. Cependant, en s'appuyant sur les testes que nous avons fait, choisir une telle valeur de Y permet de déceler un nombre très restreint d'informations récurrentes, à partir duquel nous ne pouvons pas identifier les différentes entités métiers d'une entreprise. Le nombre d'informations récurrentes recensées augmente en augmentant la valeur de Y . Suite à plusieurs expérimentations, nous recommandons l'utilisation d'une valeur de Y égale à 2 ou 3 pour avoir un nombre acceptable d'informations récurrentes. Toutefois, les analystes peuvent choisir librement la valeur qui correspond le plus à leur besoin de réutilisation. En effet, l'identification des services les plus appropriés à une entreprise ne peut pas être faite aveuglement, en suivant des règles mathématiques rigides. Le seuil proposé par la présente démarche est mentionné à titre de recommandation, selon lequel les analystes peuvent faire leur jugement sur le degré de sollicitation d'une information.

Cette étape aura comme artefacts, les informations les plus récurrentes dans les processus d'affaires de l'entreprise, c'est-à-dire celles dont le nombre d'occurrences est supérieur ou égal au seuil fixé par les analystes. Ces artefacts représentent le premier groupe d'informations clés identifiées. Le reste de ces informations va être recensé, au cours de la prochaine sous-étape, à partir des activités les plus sollicitées.

3.3.1.4 Sélection des activités les plus sollicitées et des informations qui leur sont corrélées

La sous-étape de sélection des activités les plus sollicitées dans les processus d'affaires de l'entreprise est similaire à l'étape précédente. Elle consiste à recenser les activités dont le nombre d'occurrences est supérieur à un seuil donné. Pour cette fin, nous utilisons la même équation (3.1) et la même recommandation de la valeur de Y ($Y=2$ ou $Y=3$) pour la détermination des activités les plus récurrentes. Une fois que ces dernières sont identifiées, nous allons les parcourir une par une, et voir quelles sont les informations qui lui sont corrélées. Une information est dite corrélée à une activité, si elle est présente dans la majorité de ses occurrences.

Prenant l'exemple suivant : Étant donnée une activité métier « créer compte client » et considérons que cette activité est sollicitée quatre fois dans les processus d'affaires de l'entreprise. Supposons que, dans trois de ces quatre occurrences, cette activité fait appel à l'information « Informations sur le client ». Nous pouvons dire dans ce cas, que l'information « Informations sur le client » est fortement corrélée à l'activité « Créer compte client », puisqu'elle est présente dans 75 % ($3/4$) des occurrences de cette activité. À ce niveau, nous supposons qu'une information est corrélée à une activité, si elle est présente dans au moins 50 % de ses occurrences. Ce seuil peut être ajusté selon les besoins des analystes.

Les informations corrélées, identifiées à ce niveau, vont former le deuxième groupe constituant l'ensemble des informations clés. Ces informations seront analysées durant l'étape suivante afin de pouvoir identifier les candidats de services, qui vont jouer le rôle de services entités.

3.3.1.5 Identification des services entités

Au cours de cette étape, nous allons analyser les informations clés que nous venons d'identifier et recenser les liens sémantiques entre elles. Ceci, afin de pouvoir les regrouper et former les entités métier manipulées par les processus d'affaires. Afin d'assurer le principe d'autonomie, les entités métier, identifiées à ce stade, doivent être indépendantes l'une de l'autre. Cette étape est totalement manuelle, elle ne peut pas être automatisée vu qu'elle nécessite une analyse sémantique des liens qui peuvent exister entre les informations recensées.

Étant donné l'ensemble suivant d'informations clés : « identifiant client », « rapport client » et « demande client ». En analysant ces informations, il nous est évident qu'il existe un lien sémantique entre ces informations. Ces dernières peuvent être alors regroupées et former ainsi l'entité métier « Client ». À cette entité nous pouvons associer le service entité « Client » qui pourrait la gérer.

En arrivant à ce stade, nous aurons donc identifié tous les services entités qui correspondent le plus aux besoins de l'entreprise, tout en respectant les principes de base de SOA que nous pouvons assurer à ce stade d'analyse, à savoir : la réutilisabilité et l'autonomie. Ces principes seront assurés aussi durant la prochaine étape, au niveau duquel il y aura l'identification des opérations des services entités.

3.3.2 Identification des opérations des services entités

Les opérations d'un service entité peuvent être de deux types : soit des opérations basiques ou des opérations métiers (Erl, 2005). Les opérations basiques sont des opérations standards, liées à la manipulation de l'entité métier gérée par le service, telles que la création, la modification et la suppression de l'entité. Tandis que les opérations métiers sont des opérations qui encapsulent une logique d'affaires, liée à une fonctionnalité que le service doit

assurer. Comme opération métier, nous pouvons citer, les opérations de recherche (exemple : rechercher par date une commande), les opérations de calcul (exemple : calculer le nombre de commandes réalisées durant le mois courant), les opérations de validation d'une information (exemple : valider une commande), etc. Toutefois, indépendamment du type d'opérations à identifier, ces dernières doivent être autonomes et offrir des fonctionnalités réutilisables.

Le processus d'identification des opérations, pour les services entité, dépend principalement du type de l'opération à identifier. Pour les opérations basiques ou standards, l'identification se fait à partir des informations clés, à partir desquelles sont identifiés les services entités. Le processus d'identification consiste à parcourir ces informations clés, et voir les différents types d'usages réalisés sur ces dernières (lecture, écriture, modification, etc.). Pour chaque type d'utilisation, nous associons l'opération basique qui lui correspond. Ainsi, le langage de modélisation, choisi comme support à cette approche, doit permettre la spécification du type d'usage des informations au niveau de chaque activité métier.

Les opérations basiques, identifiées à partir des informations clés, doivent offrir des fonctionnalités réutilisables. Ce principe de réutilisabilité est respecté, étant donné que ces opérations sont identifiées à partir des informations les plus sollicitées dans le métier de l'entreprise. Afin d'assurer le deuxième principe lié à l'autonomie, les opérations basiques ou standards doivent être indépendantes des autres services, évitant ainsi tout chevauchement fonctionnel, sinon le service en question ne sera plus considéré comme autonome.

Prenant l'exemple d'un service entité intitulé « Commande », identifié à partir de plusieurs informations clés, parmi lesquelles il y a l'information nommée « détail commande ». Supposons que cette information, tout au long des processus d'affaires, est soit lue, soit modifiée, ou créée par une activité métier. Dans ce cas, il est nécessaire pour le service « Commande » d'avoir des opérations qui permettent ces types de manipulations. Ces opérations pourraient avoir les noms, « obtenirDétailCommande », « modifierCommande »

et « créerCommande » pour permettre respectivement la consultation, la modification et la création de l'entité métier « Commande ».

Dans le cas d'opérations métiers, l'identification d'opérations se fait à partir des activités les plus sollicitées, identifiées dans la section « 3.3.1.4 Sélection des activités les plus sollicitées et des informations qui leur sont corrélées ». Ces activités présentent des candidats potentiels d'opérations de service. Les opérations identifiées à partir de ces activités assureront des fonctionnalités réutilisables, étant donné que ces dernières (les activités) sont très sollicitées dans les processus d'affaires. Une activité sollicitée ne peut être considérée comme opération métier, que si elle satisfait les trois critères ci-dessous :

- Elle manipule à chaque occurrence une et une seule entité métier (c'est-à-dire, des informations clés liées à une seule entité), et cette entité elle est toujours la même pour chaque occurrence. En effet, une activité, pour qu'elle soit considérée comme une opération de service elle doit être exclusive à ce dernier et indépendante des autres services (elle n'a pas besoin d'autres informations clés liées à d'autres services pour pouvoir être menée), ceci afin d'assurer le principe d'autonomie que doit respecter un service ainsi que ses opérations.
- Elle ne requiert pas une intervention humaine, étant donné qu'une opération de service est complètement automatique et ne requiert pas l'intervention humaine pour pouvoir accomplir la tâche métier qui lui est affectée.
- La fonctionnalité offerte par l'activité métier n'est pas encapsulée par une opération déjà identifiée.

Arrivant à ce niveau, nous aurons identifié tous les services entités qui permettent de gérer les différentes entités métiers de l'entreprise, ainsi que leurs opérations. L'identification de ce type de services se base principalement sur l'analyse statistique des processus d'affaires de l'entreprise. Un critère de base que nous pouvons relever à ce stade concernant le langage de modélisation qui convient le plus à notre démarche est que ce dernier doit permettre l'exploitation automatique des constituants des processus d'affaires (les activités métiers, les

pièces d'information manipulées ainsi que les ressources informatiques utilisées par chaque activité). Ceci, afin de pouvoir automatiser la collecte des statistiques liées aux processus d'affaires. En plus, le langage de modélisation choisi doit permettre la spécification du type d'usage de l'information au niveau de l'activité. Ceci nous est utile dans l'identification des opérations de type, basiques, des services entités.

3.3.3 Identification des services d'application

Les services d'application assurent une abstraction orientée services de l'environnement technique du système d'information de l'entreprise. Pour cette raison, il est difficile d'identifier, à partir des processus d'affaires, tous les services d'application qu'une entreprise doit avoir pour bien mener ses fonctions. Cependant, nous pouvons toujours recenser quelques services d'application candidats, surtout dans le cas où le langage BPM (choisi pour modéliser les processus d'affaires) permet de fournir assez de détails sur l'environnement TI de l'entreprise. Les services d'applications constituent le niveau d'abstraction le plus bas d'une architecture orientée services. Ces services sont destinés à offrir un haut degré de réutilisabilité et d'autonomie, ceci afin d'être facilement composables par des services appartenant à des couches plus élevées.

Dans la présente approche, l'identification des services d'application est réalisée à partir des informations clés recensées au niveau de la phase d'identification des services entité. En effet, au cours de cette phase, nous pouvons avoir des informations clés qui ne peuvent pas être regroupées, ou affectées à une entité métier. Toutefois, elles peuvent nécessiter des traitements d'ordre techniques, tout au long des processus d'affaires, dans lesquelles elles sont impliquées (exemple : génération d'un document, conversion d'un format à un autre, etc.). Ces informations clés doivent être groupées en unités de traitement, selon le type de traitement qu'ils requièrent. Ces unités seront affectées à un service d'application qui sera responsable de ces traitements. Lors du regroupement des informations clés, les analystes métier doivent s'assurer que les unités de traitement formées sont complètement

indépendantes les unes des autres, de façon à ne pas avoir un chevauchement fonctionnel lié aux traitements techniques à réaliser. Ceci nous permet d'assurer le deuxième principe de SOA auquel nous nous intéressons dans cette approche, à savoir l'autonomie des services. Le premier principe lié à la réutilisation est assuré, quant à lui, par le biais des informations clés (source d'identification des services d'application), qui sont les plus utilisées est les plus significatives dans les processus d'affaires de l'entreprise.

Étant donné le caractère technique des services d'application, l'étape d'identification de ce type de services doit être réalisée conjointement avec les analystes TI de l'entreprise, qui maîtrisent le mieux le système TI et ses fonctionnalités.

3.3.4 Identification des opérations des services d'application

Le processus d'identification des opérations des services d'application est similaire à celui d'identification des opérations des services entités, à l'exception que nous n'identifions pas, pour les services d'application, des opérations de type métier. En effet, un service d'application ne doit contenir, par définition, que des opérations purement techniques. Ces opérations correspondent aux types de manipulations réalisées sur les informations qui nous ont servi à l'identification des services d'application. Ces opérations doivent tenir compte des contraintes techniques imposées par l'environnement TI de l'entreprise. En plus, elles doivent être indépendantes des autres services afin de respecter le principe d'autonomie.

Le processus d'identification des opérations liées aux services d'application est le suivant : pour chaque service d'application, nous parcourons les informations clés, qui nous ont servi à son identification, et nous recensons les différents types d'usage de ces informations (consultation, modification, création, etc.), tout au long des processus d'affaires. Selon ces types de manipulation, nous recensons les différents traitements requis que doit assurer chaque service (exemple : conversion, génération de documents, etc.). Pour chaque traitement identifié, nous associons une opération de service.

Par exemple, dans un processus d'affaires, on peut avoir le besoin de générer des documents « Word » ou « PDF » à partir d'informations représentées ou échangées sous un format XML. Pour cela, nous aurons besoin d'un service d'application qui aura comme tâche de convertir un document XML vers les formats voulus. On aura ainsi des opérations de service ayant par exemple comme nom « convertToWord » ou « convertToPDF ».

3.3.5 Identification des services centrés-tâche

L'identification des services centrés-tâche ou « *Task-centric services* » vient suite à l'identification des services d'application et des services entités (ainsi que leurs opérations). Les services centrés-tâche constituent le deuxième niveau de la couche métier d'une architecture orientée services (le premier niveau est constitué par les services entités) (Figure 3.1). Ce type de services présente des tâches (ou activités) métiers à gros-grain.

À ce stade, nous ne donnons pas une grande importance au principe de réutilisabilité. En effet, les services centrés-tâche sont destinés à encapsuler une logique métier pas nécessairement réutilisable. Cependant, il est toujours préférable d'avoir des services métiers réutilisables.

La procédure d'identification des services centrés-tâches consiste principalement à identifier d'abord, les activités à gros-grain candidates à être des services centrés-tâche et définir ensuite, la logique qui va être encapsulée par chacun de ces services.

3.3.5.1 Identifier les activités candidates à être des services centrés-tâche

Les services centrés-tâches, comme leur nom l'indique, sont des services responsables du bon déroulement de certaines tâches, pas nécessairement fréquentes dans le métier de l'entreprise. La réutilisabilité n'est pas donc un critère de base pour ce genre de service. Pour cette raison, toutes les activités métiers (sélectionnées au niveau de la section « 3.3.1.1

Sélection des activités sujettes d'étude ») feront l'objet d'une étude, indépendamment de leurs degrés de sollicitation. Contrairement aux services d'application et des services entités, pour lesquels l'identification est réalisée à partir des activités les plus sollicitées.

Les services centrés-tâche permettent d'encapsuler des tâches métiers à gros-grain. L'identification de ce type de tâches est réalisée à partir des activités qui impliquent, à chaque occurrence, plusieurs services et à chaque fois elles utilisent les mêmes, indépendamment de leurs nombres d'occurrences. Ces activités sont considérées comme des activités à gros-grain, au niveau desquelles les concepteurs de processus d'affaires ont choisi d'encapsuler une logique métier, qui correspond à la composition de plusieurs services. Afin de respecter ce choix d'encapsulation et afin de rester conforme au processus d'affaires de l'entreprise, cette logique fera partie d'un service centré-tâche, au lieu de la diviser en plusieurs activités et l'intégrer dans la couche d'orchestration. Toutefois, l'activité sélectionnée doit être complètement automatique, étant donné qu'un service ne requiert pas l'intervention humaine pour pouvoir accomplir la tâche métier qui lui est affectée.

L'identification des services centrés-tâche se termine par la réalisation de graphes de dépendances qui décrivent les liens entre les services centrés-tâche et les services qui les composent. À partir de ce type de graphes, les analystes peuvent décider soit de créer un nouveau service centré-tâche qui va composer d'autres services, ou combiner les services composants en un seul service centré-tâche. La figure 3.3 présente un exemple de graphe de dépendance à partir duquel, les analystes peuvent décider soit de modéliser un nouveau service S1 qui va contrôler S2, S3 et S4, soit combiner ces trois services en un seul.

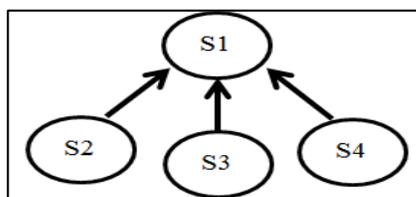


Figure 3.3 Diagramme de dépendance entre services.

La tâche d'identification de services centrés-tâches peut être automatisée selon les facilités offertes par le langage BPM choisi pour modéliser les processus d'affaires. Si ce langage est incarné dans un outil qui offre un moyen automatique d'exploiter les informations liées à chaque activité (informations, ressources utilisées, etc.), nous pourrions ainsi extraire automatiquement la liste d'activités candidates pouvant être un service centré-tâche. Toutefois, c'est toujours aux analystes de décider si une activité sera retenue comme un candidat de service ou non. Ils peuvent, évidemment ajouter d'autres candidats de service, en fonction des besoins métiers de l'entreprise.

3.3.5.2 Définir la logique des candidats de service centrés-tâche

Les services centrés-tâche permettent d'encapsuler une logique métier, qui va coordonner des services des couches inférieures. Au cours de cette étape, nous allons étudier avec minutie cette logique métier et recenser les différentes interactions qui peuvent exister entre les services composants. À cette fin, nous allons utiliser le diagramme de séquence d'UML. Ce type de diagramme assure une représentation intuitive des interactions entre plusieurs entités. Il vient pour fournir plus de détails sur les échanges réalisés entre les services, surtout que ces échanges sont encapsulés dans des activités et non modélisés par les processus d'affaires. Les diagrammes de séquence permettent d'avoir une vue commune de la logique métier encapsulée par le service centré-tâche, qui va être traduite en code, plus tard, dans la phase de développement de services.

3.3.6 Définir la couche d'orchestration

La couche d'orchestration constitue la couche la plus haute d'une architecture orientée services. Son rôle est de s'assurer que les services, appartenant aux couches inférieures, correspondent bien aux processus d'affaires de l'entreprise. Cette couche est le résultat de la transformation des processus d'affaires (ou partie d'un processus) en orchestrations exécutables. Ces orchestrations peuvent être par la suite converties en services, dits

d'orchestration. Cette approche utilise BPEL comme langage de développement des orchestrations à mettre en place. Ce choix vient du fait que BPEL représente le standard le plus reconnu et le plus utilisé pour l'exécution des processus d'affaires.

Comme nous l'avons précisé dans « 1.6.2 Langage d'exécution des processus d'affaires BPEL », BPEL est un langage structuré, basé sur XML. Un tel langage rend la tâche de définition de services orchestration, très technique et pas évidente pour les analystes métier. Toutefois, il existe plusieurs outils graphiques qui permettent la création de scripts BPEL, sans pour autant, avoir des connaissances poussées sur ce langage. Cependant, il y a toujours le risque de ne pas reproduire, dans la couche d'orchestration, la même logique représentée par les processus d'affaires. Afin d'éviter ce risque, nous proposons une procédure complètement automatisée qui permet de générer des services d'orchestration à partir des processus d'affaires, assurant ainsi une reproduction fidèle de la logique métier encapsulée par les processus d'affaires. L'identification des processus à orchestrer et la traduction de ces derniers en orchestrations BPEL seront le sujet des deux prochaines sections.

3.3.6.1 Identification des processus à orchestrer

Une entreprise peut, théoriquement, orchestrer tous ses processus d'affaires, à condition que ces derniers soient automatisés ou semi automatisés. En effet, l'identification des services d'orchestration est, souvent, très aléatoire et ne suit pas des règles bien précises. Elle dépend principalement des choix des analystes et des besoins de l'entreprise. Cependant, l'entreprise pourrait ainsi engendrer des coûts inacceptables, sans pour autant avoir une valeur ajoutée. Pour cela, il est nécessaire de suivre un ordre de priorité pour chaque processus à orchestrer. Cet ordre peut être spécifié en respectant certains critères tels que : l'effort estimé, le coût engendré et le ROI (*Return On Investment*) espéré par cette orchestration, les politiques de sécurité appliquées au processus à orchestrer, etc. Pour cette fin, les analystes métier peuvent avoir recours à des méthodes d'analyse multicritères tel que AHP (*Analytic Hierarchy*

Process), WSM (*Weight Sum Method*), WPM (*Weight Product Method*), etc. Une telle analyse vise à justifier les choix de priorités à affecter à chaque processus d'affaires.

L'artefact de cette étape est une liste complète des processus d'affaires à orchestrer, avec l'ordre de priorité de chacun.

3.3.6.2 Orchestration des processus d'affaires

L'orchestration d'un processus d'affaires passe par les deux étapes suivantes :

- La transformation du processus métier en un modèle intermédiaire (qui décrit le processus de point de vue TI) (Figure 3.4 (a)).
- La génération du code BPEL de l'orchestration à partir de ce modèle intermédiaire (MI) (Figure 3.4 (b)).

Ces étapes vont être étudiées avec minutie au cours des deux prochaines sections.

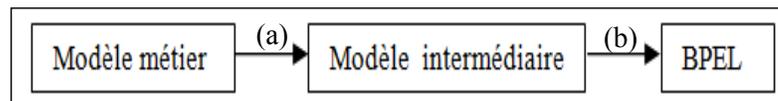


Figure 3.4 Processus d'orchestration des services.

3.3.6.3 Transformation d'un modèle métier en un modèle intermédiaire

Les processus d'affaires sélectionnés précédemment « 3.3.6.1 » ne peuvent pas être directement traduits en BPEL. En effet, ces processus peuvent inclure des tâches manuelles, ou autres utilisant des ressources non informatiques qui ne peuvent pas être orchestrées par des moteurs d'orchestration. Pour cela, il est indispensable d'épurer ces processus d'affaires et enlever toute activité ou information non tenue en compte par l'orchestration à mettre en place. On obtient ainsi un modèle intermédiaire (MI), qui présente une vue TI du modèle

métier de départ. Ce MI s'intéresse juste aux échanges entre systèmes informatiques (Logiciel, portail Web, service Web, etc.), réalisés tout au long d'un processus d'affaires.

Le MI modélisé à cette étape va être converti en un script BPEL, prêt à être déployé sur un moteur d'orchestration, à quelques modifications près. En effet, si le langage BPM, avec lequel sont modélisés les MI, fournit assez de détails concernant les systèmes informatiques utilisés, ainsi que les informations échangées, il n'y aura pas dans ce cas beaucoup de manipulation à faire au script BPEL pour qu'il soit prêt à être déployé dans un moteur d'orchestration.

Le passage d'un MI vers un script BPEL, peut introduire beaucoup de confusion et être source de plusieurs erreurs. Une solution à ce problème est de transformer automatiquement le MI en code BPEL, éliminant ainsi toute source d'erreur. Afin de permettre cette automatisation, le MI doit respecter certains critères, à savoir :

- Les informations représentées par un MI doivent être de type électronique (courriel, message XML, etc.). Les informations papier (ou de même type) ne sont pas permises dans un MI. Ceci est évident puisque le BPEL à générer va assurer l'orchestration de systèmes informatiques, qui échangent des informations purement électroniques.
- Le MI doit spécifier, pour chaque activité métier, le système informatique utilisé ainsi que les différents messages échangés avec ce système pour mener cette activité.
- Le MI doit spécifier clairement le point de départ et le point de fin du processus. Ces points de départ et de fin doivent être uniques, étant donné qu'ils sont uniques pour une orchestration BPEL.

Les MI qui respectent les critères ci-dessus peuvent être transformés automatiquement en scripts BPEL.

3.3.6.4 Génération automatique d'un BPEL à partir d'un modèle intermédiaire

La génération automatique d'un script BPEL à partir d'un MI permet de réduire les erreurs de reproduction de la logique métier encapsulée par un MI, et accélérer le processus de mise en place de la couche d'orchestration. Cependant, le code BPEL et le MI sont, de point de vue structure, complètement différents l'un de l'autre. Le MI présente un modèle élaboré avec un langage de modélisation graphique (dans notre cas UMN), dans lequel on trouve des activités et des nœuds de décision connectés l'un à l'autre. Le MI est destiné à des fins d'analyse et de représentation graphique des processus d'affaires. Le MI n'inclut pas des détails ou des informations techniques liées à la manière dont ce processus va être exécuté. Le BPEL, quant à lui, représente un langage structurel qui décrit techniquement la manière avec laquelle va être exécutée la logique métier décrite par une orchestration. BPEL est supporté par la plupart des plateformes d'exécution. Étant donnée cette grande différence entre un langage de modélisation graphique (tel que UMN) et un langage structurel d'orchestration de processus (tel que BPEL), la question suivante peut se poser : Comment un modèle d'affaires graphique peut être transformé automatiquement à un script BPEL, structurellement bien organisé?

La translation d'un MI en un code BPEL passe par l'analyse structurelle du premier et l'identification des différents patrons structurels « *Structural patterns* » (boucles répétitives, conditionnelles, branches parallèles, etc.) qui peuvent être traduits facilement en code BPEL. Le problème revient alors à identifier des patrons structurels et la génération du code BPEL correspondant à chaque patron (Figure 3.5).

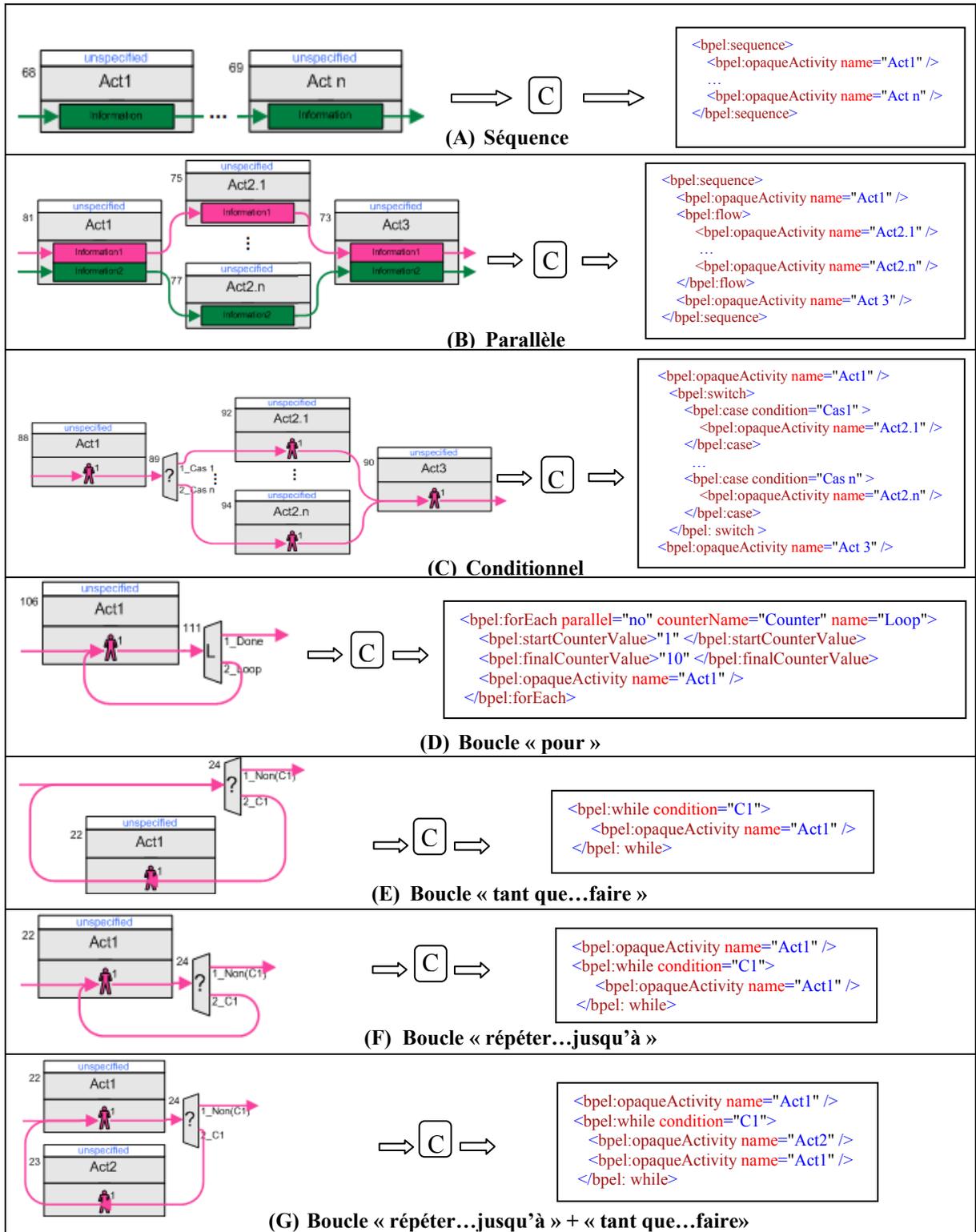


Figure 3.5 Translation d'un modèle MI en un code BPEL.

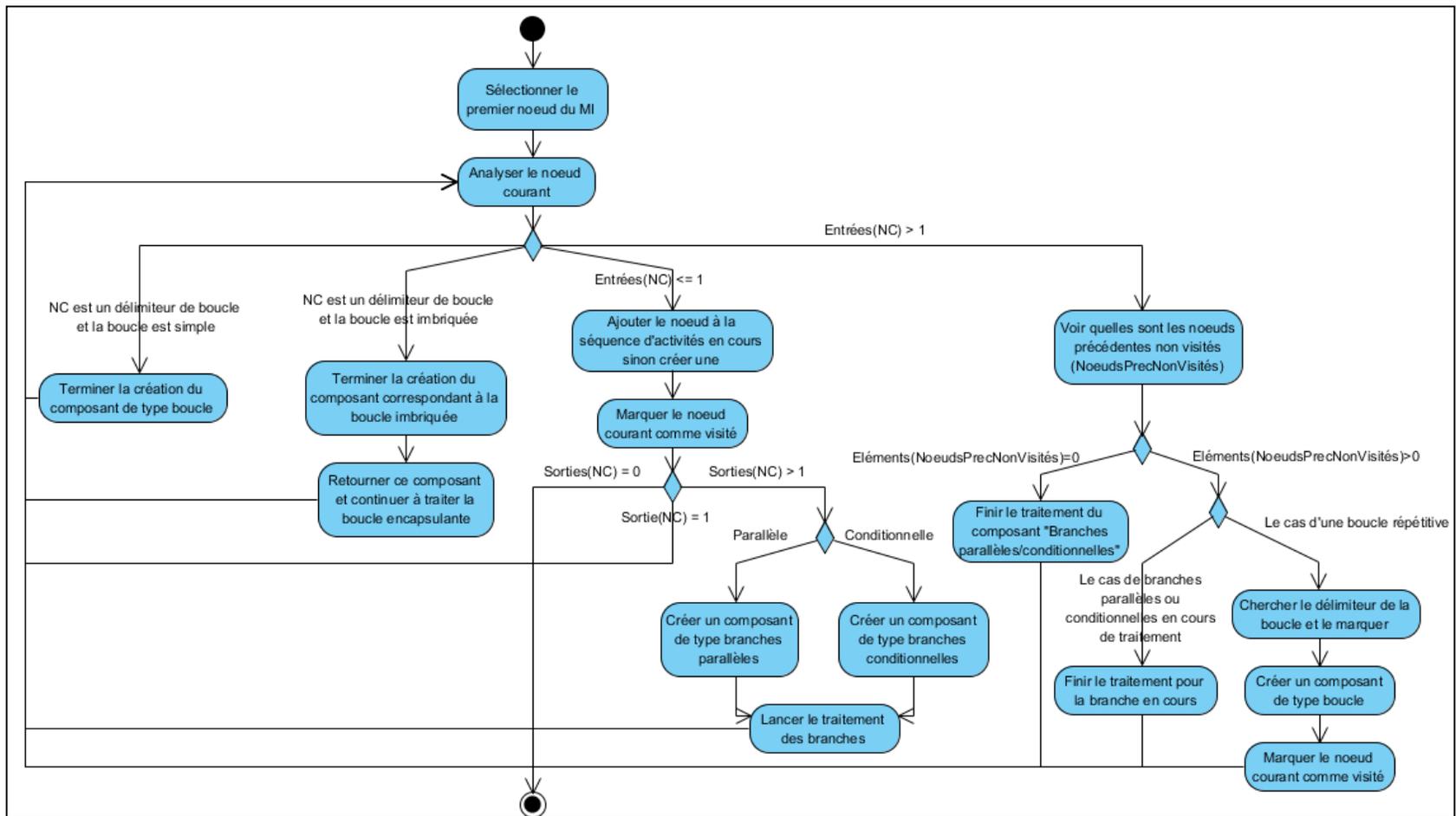


Figure 3.6 Algorithme de génération d'un BPEL à partir d'un MI.

La figure 3.6 présente l'algorithme que nous proposons pour l'identification des différents « *patterns* » qui peuvent exister dans un modèle d'affaires. Cet algorithme consiste à identifier, récursivement, les différents patrons illustrés par la figure 3.5. Ces patrons seront retournés sous forme de composants, qui vont être par la suite traduits en code BPEL. Le principe général de l'algorithme est le suivant : nous parcourons un par un les nœuds du MI et nous analysons, pour chacun, ses nœuds précédents et suivants.

Ci-dessous les différents cas qui peuvent figurer dans une transformation d'un MI en code BPEL (Figure 3.7). Dans ce qui suit « Entrées(NC) » et « Sorties(NC) » sont deux fonctions qui permettent de calculer, respectivement, le nombre de nœuds précédents et suivants du nœud courant (NC) :

- Le nœud courant a au plus une seule entrée (Entrées(NC) \leq 1) :
 - Dans le cas où NC possède une seule sortie (Sorties(NC)=1), NC sera alors considéré comme un nœud faisant partie d'une simple séquence d'activités.
 - En cas où NC possède plusieurs sorties ou successeurs (Sorties(NC) $>$ 1), on sera alors dans le cas de branches parallèles ou conditionnelles qui vont être traitées une par une.
 - Sinon dans le cas où NC ne possède pas de successeurs (Sorties(NC)=0), on atteint alors le dernier nœud du MI et on finit le processus.
- Le nœud courant a plusieurs entrées (Entrées(NC) $>$ 1) : dans ce cas, nous analysons le NC afin de recenser, parmi les nœuds précédents, ceux qui ne sont pas encore traités ou visités. Ces nœuds vont former l'ensemble « NoeudsPrecNonVisités ». Si tous les nœuds précédents sont déjà traités, alors il s'agit d'un cas de jointure de branches conditionnelles ou parallèles (Figure 3.7 (a)). Sinon s'il existe des nœuds précédents non visités, nous serons alors soit :
 - dans un cas de branches parallèles ou conditionnelles qui sont en train d'être traitées (Figure 3.7 (b)),
 - ou dans le cas d'une boucle répétitive (Figure 3.7 (c)).

- Dans le premier cas (branches parallèles ou conditionnelles), nous devons terminer le traitement de toutes les branches pour pouvoir achever le traitement du nœud courant. Dans le deuxième cas (boucle répétitive), il faut chercher le nœud qui va marquer la fin de cette boucle et le mettre comme délimiteur (Exemple : dans la figure 3.7 (c), le nœud 24 joue le rôle du délimiteur de boucle).
- Un cas de délimiteur de boucle : dans ce cas, nous pouvons avoir deux possibilités, soit la boucle en cours est simple, dans ce cas nous terminons la création du composant qui lui correspond, soit la boucle est imbriquée ou encapsulée dans une deuxième boucle, dans ce cas, nous aurons à créer le composant correspondant à la boucle imbriquée et l'affecter à la boucle encapsulante (Figure 3.7 (d)).

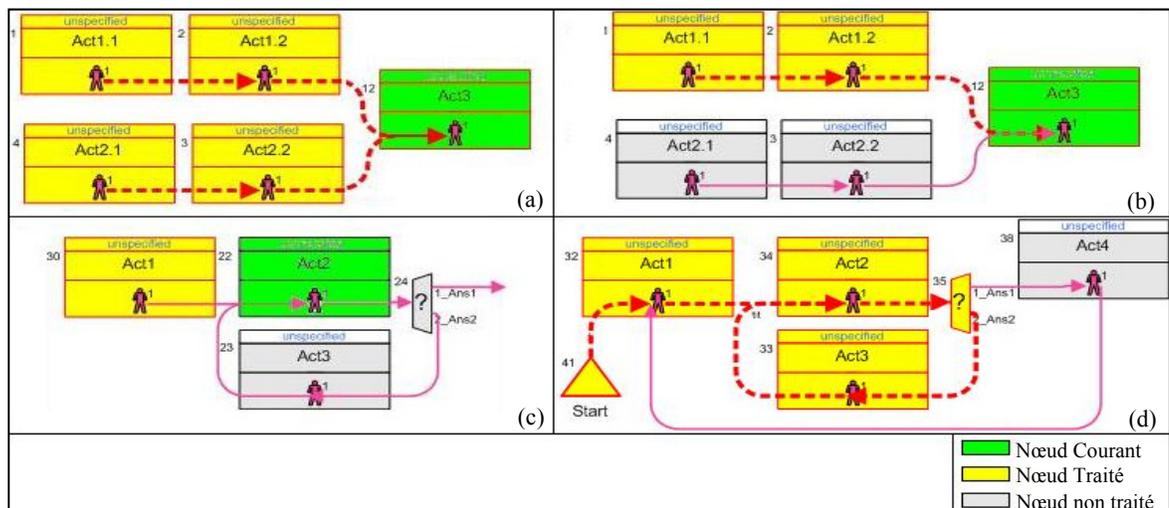


Figure 3.7 Exemples de parcours dans un MI.

Suite à l'exécution de cet algorithme, nous aurons identifié les différents patrons structurels qui forment un MI donné. Ces patrons seront traduits par la suite en code BPEL, qui va être à son tour déployé sur un moteur d'orchestration. Afin de réduire l'effort que doit faire un développeur avant de déployer le code BPEL généré, il est préférable de choisir un langage BPM qui offre assez de détails concernant les ressources informatiques utilisées, la structure des messages échangés, leurs types, etc.

Arrivant à ce stade, nous aurons défini les traits de base de l'architecture SOA à mettre en place, en définissant pour chaque couche orientée services les candidats de service qui vont la construire. Ci-dessous un algorithme qui résume l'approche que nous proposons pour l'identification des services.

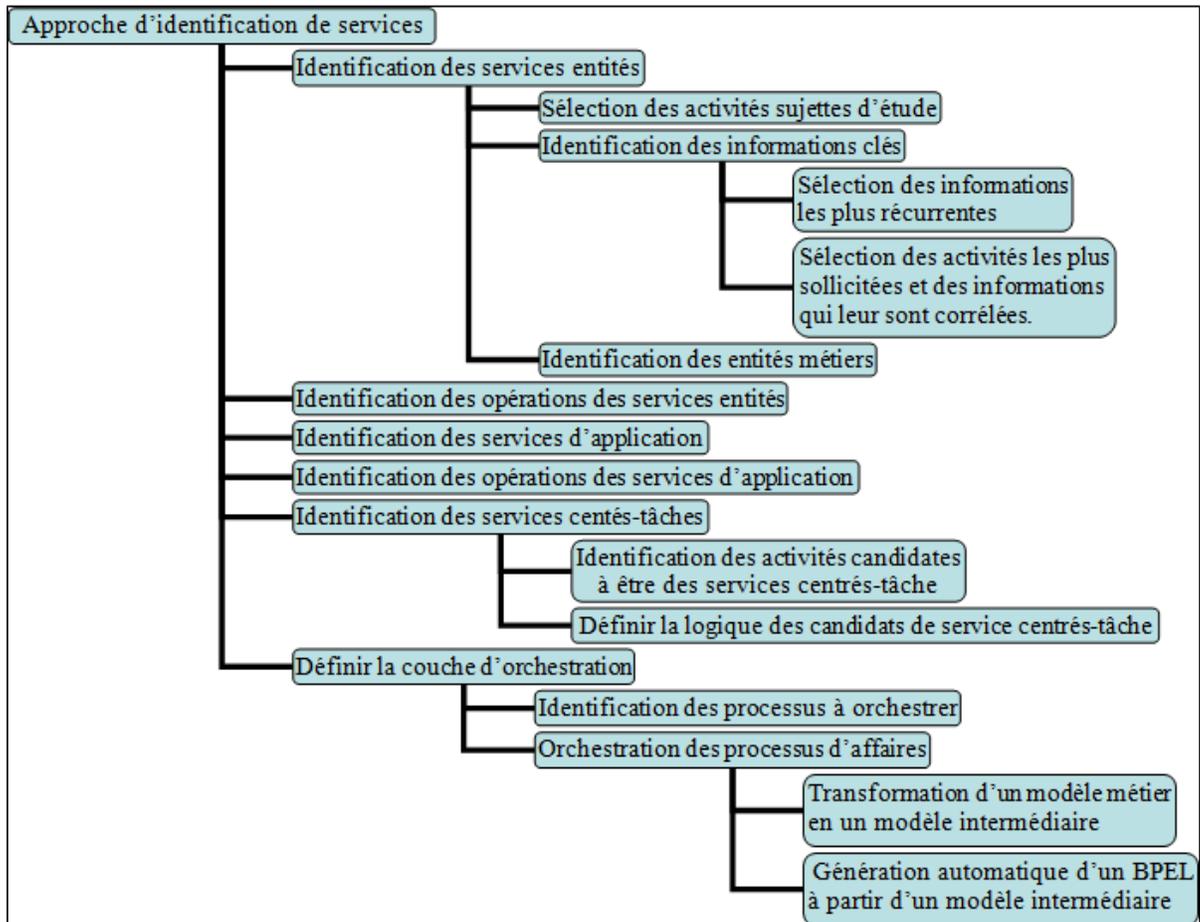


Figure 3.8 Algorithme résumant l'approche proposée.

3.4 Conclusion

Nous avons proposé dans ce chapitre une nouvelle approche permettant de faciliter le processus d'identification des services les plus appropriés à une entreprise. Cette approche, basée sur l'analyse statistique des processus d'affaires, essaye de scruter plusieurs démarches d'identification de service, en automatisant, le plus possible, les étapes amenant à

l'identification des services les plus appropriés. Cette automatisation est conditionnée par la capacité du langage BPM d'offrir une flexibilité suffisante concernant l'exploitation des données liées au processus d'affaires.

Au cours de ce chapitre, nous avons pu présenter avec détails l'approche proposée pour l'identification des services les plus appropriés à une entreprise, en exposant avec minutie les lignes directrices amenant à cette identification. Nous avons également pu justifier tous les critères de sélection sur lesquels nous nous sommes basés dans cette approche. Cette approche sera validée dans le chapitre suivant à travers son application à un cas réel.

CHAPITRE 4

VALIDATION DE L'APPROCHE PROPOSEE

4.1 Introduction

Ce chapitre a pour ambition de valider l'approche proposée et conçue au chapitre précédent. Nous allons appliquer la présente démarche sur un cas réel d'une banque d'affaires, composée de cinquante succursales qui offrent différents types de produits et services financiers : compte chèque, épargne, prêts pour entreprises, fonds de placement, etc. Le métier de la banque d'affaires, sujette d'étude, est représenté par onze processus d'affaires modélisés en notation UMN.

Dans ce chapitre nous commençons d'abord, par l'identification des services entités et les services d'application, ainsi que leurs opérations respectives. Ensuite, nous allons identifier les services centrés-tâche qui répondent le mieux aux besoins de la banque. Avant de conclure le chapitre, nous allons montrer comment identifier les services qui vont constituer la couche d'orchestration.

4.2 Identification des services entités

Au cours de cette étape, nous allons identifier les services qui permettent de gérer les entités les plus significatives dans le métier de l'entreprise.

4.2.1 Sélection des activités sujettes d'étude

Cette étape consiste à recenser les activités automatiques et semi-automatiques qui utilisent des ressources informatiques à partir desquelles, nous pouvons créer des services. UMN peut persister les données liées aux processus d'affaires dans une base de données relationnelle, ceci nous facilite beaucoup la tâche de filtrage des activités métiers qui sont, dans notre cas

au nombre de 452 activités. UMN nous permet aussi de desceller facilement le lien entre les activités et les ressources (matérielles, humaines, etc.) qu'elles utilisent, donnant ainsi une idée sur le degré d'automatisation de chaque activité.

Étant donné que, dans notre cas, les données métiers qui portent sur les processus d'affaires sont stockées dans une base de données, alors leur exploitation sera effectuée par le biais de requêtes SQL (*Structured Query Language*). Le filtrage des activités est réalisé par le biais de deux requêtes imbriquées. La première requête permet de recenser toutes les ressources de type informatiques susceptibles d'être modifiées, afin de pouvoir créer ou dialoguer avec des services. La deuxième requête filtre toutes les activités qui utilisent les ressources retournées par la première requête. Le résultat de ces requêtes nous donne un ensemble de 112 activités automatiques ou semi-automatiques (voir Annexe I).

4.2.2 Identification des informations clés

L'identification des informations clés est faite à partir des informations et des activités les plus sollicitées dans les processus d'affaires de l'entreprise. L'identification de ces dernières (les informations et les activités sollicitées) sera le sujet des deux prochaines sections.

4.2.2.1 Sélection des informations les plus récurrentes

Les informations les plus récurrentes constituent le premier lot d'informations clés à cerner à partir des processus d'affaires de l'entreprise. L'Annexe II présente une liste complète de toutes les informations métiers ainsi que leurs nombres d'occurrences. Cette liste d'informations est recensée à partir d'une requête SQL dont le résultat est exporté vers une feuille de calcul « Excel ».

La sélection des informations les plus récurrentes est réalisée suivant un seuil que nous fixons suivant l'équation (3.1). Le tableau ci-dessous présente les valeurs des différentes métriques servant au calcul de ce seuil, une valeur d'Y égale à 2 donne un seuil S égale à 14.

Tableau 4.1 Métriques de sélection des informations récurrentes

Moyenne	8,18
Écart type	12,15
Y	2
S = [Moyenne + Écart Type/Y]	14

Le seuil fixé peut être validé par le graphique ci-dessous, qui décrit la dispersion des nombres d'occurrences des informations dans les processus d'affaires. Ce graphique permet aux analystes d'avoir une idée sur la qualité du seuil choisi. En analysant (Figure 4.1), nous pouvons remarquer qu'il existe des informations ayant des occurrences dont la valeur est proche du seuil. Les analystes peuvent juger que ces informations sont assez significatives qu'elles peuvent être ajoutées à l'ensemble d'informations récurrentes.

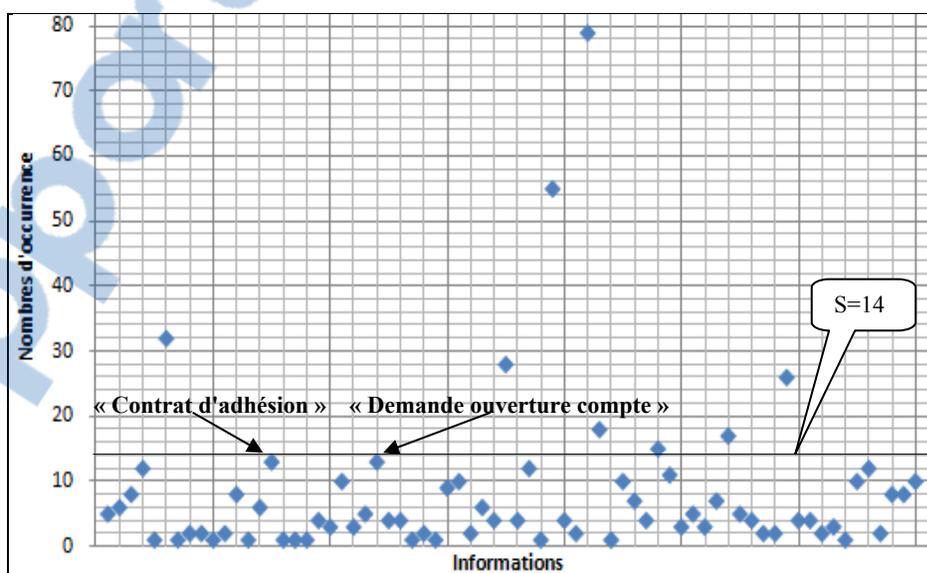


Figure 4.1 Occurrences des informations métiers.

Dans notre cas de figure, nous avons huit informations dont le nombre d'occurrences est supérieur au seuil. En analysant le graphique de la figure 4.1, nous constatons l'existence de deux informations dont le nombre d'occurrences est légèrement inférieur au seuil fixé, à savoir : « Demande ouverture compte » et « Contrat d'adhésion ». L'analyse des processus d'affaires et des informations récurrentes déjà recensées (dont le nombre d'occurrences est supérieur au seuil) nous amène à considérer les informations « Demande ouverture compte » et « Contrat d'adhésion » comme assez significatives dans le métier de l'entreprise pour qu'elles soient ajoutées à la liste des informations clés (Tableau 4.2).

Tableau 4.2 Premier lot d'informations clés

Informations	Occurrences
Avis d'ouv de cpte	32
Demande ouverture compte	13
Contrat d'adhésion	13
Form instr. Placement	28
Formulaire ouv. compte	55
Instructions de placement	79
Instructions telephonique	18
Notes au dossier client	15
Plan d'action	17
Rap. connaître son client	26

Ainsi, nous descellons le premier lot d'informations clés, identifiées à partir des informations les plus récurrentes dans les processus d'affaires. Le deuxième lot sera identifié au cours de la prochaine étape, au niveau duquel seront recensées les informations fortement corrélées aux activités métiers les plus sollicitées.

4.2.2.2 Sélection des activités les plus sollicitées et des informations qui leur sont corrélées

Le processus de sélection des activités les plus sollicitées est similaire à celui pour la sélection des informations les plus récurrentes. À ce stade, nous allons utiliser la même équation (3.1), pour déterminer le seuil à partir duquel une activité sera considérée comme

très sollicitée. Le tableau ci-dessous résume les différentes métriques utilisées pour la détermination du seuil à fixer.

Tableau 4.3 Métriques de sélection des activités sollicitées

Moyenne	2,81
Écart type	2,94
Y	2
S = [Moyenne + Écart Type/Y]	4

L'analyse de la figure 4.2 nous permet de valider le résultat trouvé par le tableau ci-dessus. Cette figure nous montre bien qu'un seuil S égal à 4 permet de sceller les activités les plus sollicitées dans les processus d'affaires de la banque.

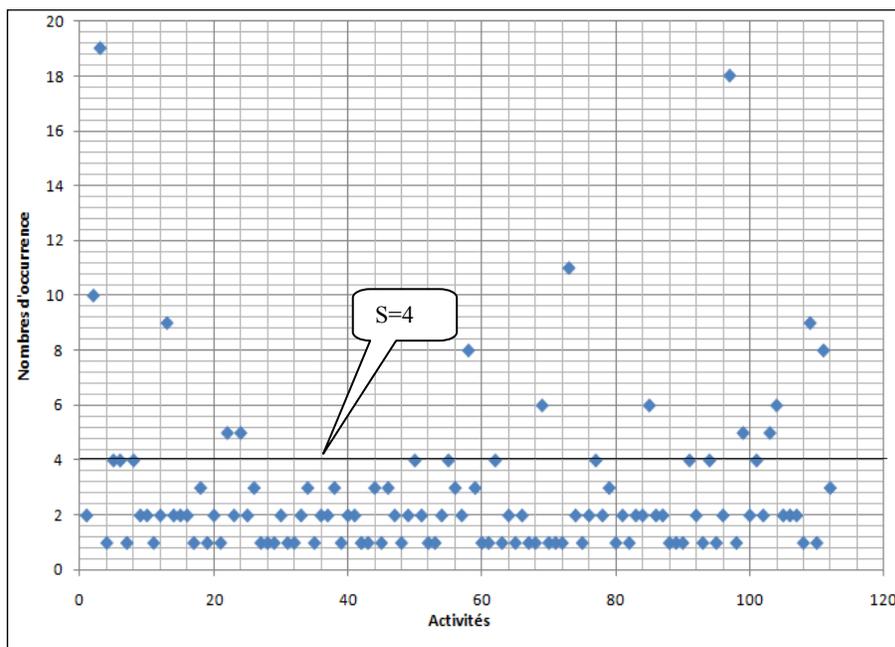


Figure 4.2 Occurrences des activités métiers.

À partir du seuil illustré par la figure ci-dessus (S=4), nous pouvons recenser les 25 activités les plus sollicitées dans les processus d'affaires de l'entreprise (Annexe III). Ces activités seront parcourues une par une afin de sceller les informations qui leur sont corrélées. Ces

informations vont constituer le deuxième lot d'informations clés. Une information est d'autant corrélée à une activité qu'elle est présente dans les occurrences de cette dernière.

Afin de faciliter la tâche d'identification des informations corrélées aux activités les plus sollicitées, nous allons créer automatiquement une matrice qui détermine pour chaque information son taux de présence dans toutes les activités métiers. Pour des raisons de simplicité, et afin de mieux comprendre le processus de sélection des informations corrélées, nous allons présenter un prototype de matrice utilisé réellement (Tableau 4.4).

Tableau 4.4 Prototype de matrice pour l'identification des informations corrélées

Information	Occurrences	Act1	Act2	Act3	Act4	Act5	Act6	Act7	Act8	Act9	Act10
		5	1	6	10	2	3	1	6	3	5
Inf1	2	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Inf2	5	0%	0%	0%	0%	0%	20%	0%	0%	0%	0%
Inf3	8	0%	0%	40%	0%	0%	0%	0%	0%	0%	0%
Inf4	12	100%	0%	0%	67%	0%	0%	0%	100%	0%	0%
Inf5	10	0%	0%	0%	0%	0%	0%	100%	0%	0%	100%
Inf6	34	0%	0%	0%	0%	0%	100%	0%	0%	0%	0%
Inf7	5	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Inf8	10	100%	0%	40%	0%	0%	0%	0%	0%	0%	30%
Inf9	10	0%	0%	0%	0%	0%	80%	0%	0%	0%	0%
Inf10	8	30%	0%	0%	80%	0%	0%	0%	0%	0%	0%

■ Activité ou information très sollicitée
■ Information fortement corrélée
■ Information clé

Prenant l'exemple de la matrice ci-dessus, les activités « Act1 », « Act3 », « Act4 », « Act8 » et « Act10 » sont les activités les plus sollicitées dans les processus d'affaires, étant donné que leurs occurrences sont supérieures ou égales au seuil fixé (nous reprenons ici les mêmes seuils trouvés dans les étapes précédentes, à savoir : 14 pour les informations et 4 pour les activités). De la même manière, « Inf6 » est considérée comme information très récurrente. Cette information fait déjà partie des informations clés. Prenant l'exemple de « Inf4 », cette information est présente dans la majorité des occurrences des activités très sollicitées « Act1 », « Act4 » et « Act8 », avec respectivement un taux de présence de 100 %, 67 % et

100 %. À ce stade, nous pouvons dire que cette information (« Inf4 ») est fortement corrélée aux activités sollicitées : « Act1 », « Act4 » et « Act8 ». Le même cas se présente pour les informations « Inf8 » et « Inf10 » qui sont fortement corrélées, respectivement à « Act1 » et « Act4 ». De ce fait, Inf4, Inf8 et Inf10 formeront le deuxième lot des informations clés.

Tableau 4.5 Informations corrélées aux activités métiers sollicitées

Informations corrélées	Occurrences
Avis de refus	12
Code de titre	8
Confirmation de transaction (nom et hre)	6
Dossier épargne collect.	10
Enregistrement vocal	6
inscription r-v	4
Liste des fonds adm.	10
Mise à jour FOC	7
Numéro de compte	11
Numéro d'investisseur	5
Rens. sur plaintes client	10
Risques liés à un emprunt	12
Suivi au plan d'action	8

Dans le cas de la banque, sujette d'étude, nous pouvons recenser 13 informations corrélées (Tableau 4.5). Ces informations seront ajoutées aux informations récurrentes identifiées précédemment (4.2.2.1 Sélection des informations les plus récurrentes), formant ainsi l'ensemble des informations clés à partir desquelles il y aura l'identification des différents services entité.

4.2.3 Identification des services entités

Au cours de cette partie, nous allons analyser les informations clés, descellées tout au long de la section précédente, et recenser les liens sémantiques qui existent entre elles. Ceci, nous permettra de dégager les différentes entités métiers manipulées dans les processus d'affaires de la banque, à partir desquelles il y aura l'identification des services entités.

Le tableau ci-dessous présente les différentes entités métiers identifiées, leurs nombres d'utilisations ainsi que les informations clés servant à leur identification. Le nombre de sollicitations d'une entité métier n'est autre que la somme des nombres d'occurrences de ses informations clés. Prenant l'exemple des informations : « Plan d'action » et « Suivi au plan d'action ». Il est clair que ces informations ont un lien sémantique entre elles, étant donné qu'elles interviennent toutes dans la gestion des plans d'action des clients. Ceci nous amène donc à dégager l'entité métier « Plan d'action », qui sera responsable sur les informations clés, à partir desquelles elle est identifiée. À partir de cette entité, nous pouvons identifier le service entité « PlanAction », qui assurera la gestion des plans d'action. De la même façon, nous pouvons dégager les entités « Compte », « Placement », « Client », « Rendez-vous » et « Instruction de placement », ainsi que leurs services entités respectifs.

Tableau 4.6 Entités métiers identifiées

Information clé	Entité métier	Service entité	Nombre d'utilisations
Contrat d'adhésion	Compte	Compte	99
Demande ouverture compte			
Mise à jour FOC			
Formulaire ouv. Compte			
Numéro de compte			
Code de titre de Placement	Placement	Placement	18
Liste des fonds adm.			
Numéro d'investisseur	Client	Client	56
Dossier épargne collect.			
Notes au dossier client			
Rap. connaître son client			
Plan d'action	Plan d'action	PlanAction	25
Suivi au plan d'action			
inscription r-v	Rendez-vous	RDV	4
Form instr. Placement	Instruction de placement	InstructionPlacement	107
Instructions de placement			

Le choix de regroupement des informations clés peut différer d'un analyste métier à un autre, et peut être le sujet de plusieurs discussions. Dans notre cas de figure, il est possible d'affecter, par exemple, les informations « Instructions de Placement » et « Form instr.

Placement » à l'entité « Placement » ayant ainsi une seule entité métier pour la gestion des placements ainsi que leurs instructions (Tableau 4.6). Dans ce cas, le degré de sollicitation de cette activité sera très élevé relativement aux autres entités (125 utilisations). Pour des raisons de performance et de qualité des services à identifier (temps de réponse, meilleure réutilisation, etc.), nous jugeons que c'est plus intéressant de créer un service à part, responsable sur la gestion des instructions de placement que de mettre le tout dans le même service « Placement ».

Indépendamment des choix de regroupement à prendre, il faut toujours s'assurer que les entités métiers identifiées sont indépendantes l'une de l'autre, ceci, afin d'assurer le principe d'autonomie que doit respecter un service entité. À partir du tableau 4.6, nous pouvons remarquer que chaque entité métier est identifiée à partir d'informations clés qui sont indépendantes de celles servant à l'identification des autres entités. De ce fait, nous pouvons dire que les entités identifiées à ce stade sont autonomes. Ainsi, les services entités seront à leurs tours autonomes.

4.3 Identification des opérations des services entités

Rappelons qu'un service entité peut avoir deux types d'opérations : basiques ou métiers. Le processus d'identification d'opérations pour un service entité dépend du type de l'opération à identifier. Les opérations de type basique sont des opérations liées aux types d'usage de l'entité métier, telles que : la création, la modification, la suppression, etc. Ces types d'usage sont déduits à partir des informations clés servant à l'identification de l'entité en question. Le Tableau 4.7 illustre pour chaque service entité, les informations clés à partir desquelles il est identifié, les types d'usage de ces informations, ainsi que les opérations identifiées.

Prenant l'exemple du service entité intitulé « InstructionPlacement », identifié à partir des informations : « Form instr. Placement » et « Instructions de placement » (Tableau 4.7). Étant donné les types d'usages reliés à ces deux informations, le service « InstructionPlacement » doit permettre la création, la modification et la consultation

d'instructions de placement. Ces fonctionnalités seront offertes respectivement par les opérations suivantes : « creerInstructionPlacement », « modifierInstructionPlacement », et « obtenirDetail InstructionPlacement ». L'Annexe IV présente une description détaillée des opérations basiques identifiées au cours de cette étape.

Tableau 4.7 Identification des opérations basiques pour les services entité

Informations clés	Types d'usage			Services entité	Opération
	Lecture	Création	Modification		
Contrat d'adhésion	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Compte	creerCompte / modifierCompte / obtenirDetailCompte
Demande ouverture compte	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
Mise à jour FOC	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
Formulaire ouv. compte	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
Numéro de compte	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
Code de titre de placement	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Placement	creerPlacement / modifierPlacement / obtenirDetailPlacement / obtenirListeFondsAdmissibles
Liste des fonds adm.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
Numéro d'investisseur	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Client	creerInformationsClient / modifierInformationsClient / obtenirInformationsClient / inscrireNotes / modifierNotes / obtenirNotes / creerDossierEpargneCollective / modifierDossierEpargneCollective
Dossier épargne collect.	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
Notes au dossier client	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
Rap. connaître son client	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
Plan d'action	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	PlanAction	créerPlanAction / modifierPlanAction / obtenirPlanAction / initierSuivi / modifierSuivi / obtenirDetailSuivi
Suivi au plan d'action	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
Inscription r-v	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	RDV	fixerRDV / modifierRDV / obtenirDetailRDV
Form instr. Placement	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	InstructionPlacement	creerInstructionPlacement / modifierInstructionPlacement / obtenirInstructionPlacement
Instructions de placement	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		

Une fois que les opérations basiques sont identifiées, nous allons maintenant procéder à l'identification des opérations de type métiers, des services entité. Ces opérations encapsulent une logique d'affaires liée à une fonctionnalité qu'un service entité doit assurer. L'identification de ces opérations métiers est réalisée à partir des activités les plus sollicitées, identifiées dans la section « 4.2.2.2 Sélection des activités les plus sollicitées et des informations qui leur sont corrélées ». Rappelons qu'une activité sollicitée ne peut être considérée comme opération métier que si elle respecte les trois critères ci-dessous :

- C1 : Elle manipule à chaque occurrence une et une seule entité métier (ou service entité), et cette entité est toujours la même pour chaque occurrence.
- C2 : Elle est complètement automatique et ne requiert pas une intervention humaine.
- C3 : La fonctionnalité offerte par l'activité métier n'est pas encapsulée par une opération déjà identifiée.

Le filtrage des activités, suivant le critère (C1), sera assuré par la matrice du tableau 4.8. Afin d'illustrer la procédure de sélection des activités qui satisfont le critère C1, nous allons reprendre, ci-dessous, le même exemple de prototype de matrice présenté dans « 4.2.2.2 Selection des activités les plus sollicitées et des informations qui leur sont corrélées ».

Tableau 4.8 Prototype de matrice pour l'identification des informations corrélées

Information	Occurrences	Act1	Act2	Act3	Act4	Act5	Act6	Act7	Act8	Act9	Act10
		5	1	6	10	2	3	1	6	3	5
Inf1	2	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Inf2	5	0%	0%	0%	0%	0%	20%	0%	0%	0%	0%
Inf3	8	0%	0%	40%	0%	0%	0%	0%	0%	0%	0%
Inf4	12	100%	0%	0%	67%	0%	0%	0%	100%	0%	0%
Inf5	10	0%	0%	0%	0%	0%	0%	100%	0%	0%	100%
Inf6	34	0%	0%	0%	0%	0%	100%	0%	0%	0%	0%
Inf7	5	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Inf8	10	100%	0%	40%	0%	0%	0%	0%	0%	0%	30%
Inf9	10	0%	0%	0%	0%	0%	80%	0%	0%	0%	0%
Inf10	8	30%	0%	0%	80%	0%	0%	0%	0%	0%	0%

■ Activité ou information très sollicitée
 ■ Information fortement corrélée
 ■ Information clé

Admettant que « Inf4 » et « Inf8 » sont deux informations clés servant à l'identification du service entité ayant le nom « ServiceEntité1 ». En consultant la matrice ci-dessus, nous pouvons remarquer que les informations « Inf4 » et « Inf8 » sont présentes dans toutes les occurrences de « Act1 ». Ceci nous amène à dire que l'activité « Act1 » utilise exclusivement le service « ServiceEntité1 ». « Act1 » satisfait ainsi le premier critère C1 de sélection des opérations métier.

Les activités métiers les plus sollicitées qui satisfont le critère C1 sont illustrées par le Tableau 4.9. Ces activités sont analysées afin de voir leur conformité avec les critères C2 et C3. Le Tableau 4.9 décrit pour chaque activité retenue, le service auquel elle est exclusive, l'opération métier identifiée à partir de cette activité, ainsi qu'une description des fonctionnalités qui vont être assurées par cette opération.

Tableau 4.9 Opérations métiers identifiées pour les services entité

Activités métiers	C2	C3	Services utilisés en exclusivité	Opérations identifiées	Description
Approuver instruction placement	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Instruction de placement	approuverInstruction	Approuver une instruction de placement en vue de la traiter par après
Apporter les corrections	<input type="checkbox"/>	<input type="checkbox"/>	-	-	-
Déposer le montant	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Compte	depotMontant	Mettre à jour le compte avec le montant déposé
Rechercher le numéro de code du titre	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Placement	rechercheCodeTitrePlacement	Rechercher le code du titre de placement
Retrait somme	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Compte	retraitMontant	Mettre à jour le compte avec le montant tiré
Traitement de l'instruction de placement	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Instruction de placement	traiterInstruction	Traduire une instruction de placement en une transaction à traiter
Vérifier et analyser les informations du FOC	<input type="checkbox"/>	<input type="checkbox"/>	-	-	-
Vérifier la disponibilité de l'agent	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	RDV	diponibilitéAgent	Voir si un agent est disponible pour un rendez-vous
Vérifier l'admissibilité du titre	<input type="checkbox"/>	<input type="checkbox"/>	-	-	-
Vérifier la convenance de la transaction	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Instruction de placement	verifierConvenanceTransaction	Vérifier si une transaction est conforme ou non à son instruction de placement
Vérifier les éléments de l'offre au	<input type="checkbox"/>	<input type="checkbox"/>	-	-	-
Vérifier l'offre Expert	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Instruction de placement	verifierOffre	Faire une liste de vérifications automatiques sur les instructions d'une offre de placement

4.4 Identification des services d'application

Les services d'applications sont identifiés à partir des informations clés qui ne nous ont pas servis dans l'identification des services entité. La liste de ces informations est illustrée par le Tableau 4.10 au niveau duquel nous essayons de regrouper ces dernières selon les traitements techniques qu'elles requièrent. Étant donné le caractère technique des services d'application, cette étape de regroupement d'information doit être réalisée conjointement avec les analystes TI, qui maîtrisent le mieux le SI de l'entreprise et ses fonctionnalités.

Les services illustrés par le tableau ci-dessous offrent un haut degré de réutilisation, étant donné que ces derniers sont identifiés à partir des informations clés. Chacun de ces services est autonome et offre des traitements techniques indépendants des traitements réalisés dans les autres services. Malgré que les descriptions des services « GenererAvis » et « GenererDocuments » se ressemblent, ces deux services offrent des fonctionnalités totalement différentes. En effet, le service « GenererAvis » assure la génération d'avis échangés, qui peuvent être le sujet de plusieurs modifications tout au long des processus d'affaires et qui peuvent être sous plusieurs formats (email, document ou message

électronique, XML, etc.). Tandis que le service « GenererDocuments » assure la génération de documents non modifiables (PDF par exemple), destinés généralement à être imprimés.

Tableau 4.10 Identification des services d'application

Information clé	Service d'application	Description
Avis de refus	GenererAvis	Génération d'avis (courriel, document électronique, etc.), qui seront échangés entre les différents acteurs et systèmes informatiques.
Avis d'ouv. de cpte		
Avis de Mise à jour de cpte		
Confirmation de transaction (nom et hre)	Gestion SystemeTelephonique	Gestion et traitement des transactions téléphoniques faites par le client.
Enregistrement vocal		
Instructions téléphoniques		
Rens. sur plaintes client	GenererDocuments	Génération de documents dont le contenu sera généré automatiquement à partir de données stockées sous un format natif
Risques liés à un emprunt		

4.5 Identification des opérations des services d'application

L'identification des opérations des services d'application est réalisée à partir des types de manipulations réalisées sur les informations clés. Le Tableau 4.11 présente, pour chaque service d'application, les différentes opérations identifiées à partir des types d'usages des informations clés.

Tableau 4.11 Identification des opérations des services d'application

Informations clés	Type d'information	Types d'usage			Services d'application	Opérations
		Lecture	Création	Modification		
Avis de refus	email	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	EnvoiAvis	envoiAvisRefus / envoiAvisOuvCompte / envoiAvisMajCompte
Avis d'ouv. de cpte	message électronique (XML)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
Avis de Mise à jour de cpte	message électronique (XML)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
Confirmation de transaction (nom et hre)	Enregistrement vocal	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	GestionSysteme Telephonique	creerConfirmationTransaction / lireConfirmationTransaction / enregistreCommunicationTel / lireCommunicationTel / enregistrerInstructionTel / lireInstructionTel
Enregistrement vocal	Enregistrement vocal	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
Instructions téléphoniques	Enregistrement vocal	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
Rens. sur plaintes client	Document PDF	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	GenererDocum ents	genereDocumentPlaintesClient / genererRisquesEmprunt
Risques liés à un emprunt	Document PDF	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		

Afin d'illustrer le processus d'identification d'opérations des services d'application, nous allons prendre comme exemple les opérations identifiées pour le service « GenererAvis ». Ce service doit gérer trois types d'avis : « avis de refus », « avis d'ouverture de compte » et « avis de mise à jour de compte ». L'avis de refus est un avis envoyé par courriel, tandis que les autres avis sont des messages électroniques de notification, échangés entre systèmes informatiques. Chaque avis est généré par une opération de service. Une fois généré, l'avis sera envoyé en vue de le consulter ou le modifier par ses receveurs, en dehors des frontières des services d'application. Pour cette raison, nous n'aurons pas d'opérations pour la modification et la lecture des avis. Une description détaillée des opérations identifiées est le sujet de l'Annexe V.

4.6 Identification des services centrés-tâche

La procédure d'identification des « *Task-centric services* » consiste principalement à identifier d'abord, les activités à gros-grain candidates à être des services centrés-tâche et à définir ensuite, la logique qui va être encapsulée par ces services.

4.6.1 Identifier les activités candidates à être des services centrés-tâche

Les services centrés-tâches sont des services qui veillent sur le bon déroulement de certaines tâches métiers, non nécessairement réutilisables. Pour cette raison, l'identification de ce type de services est réalisée à partir des activités métiers recensées dans « 4.2.1 Sélection des activités sujettes d'étude », indépendamment de leurs degrés de sollicitation. Seules les activités à gros-grain vont être retenues. Rappelons qu'une activité à gros-grain est une tâche métier atomique qui implique à chaque occurrence plusieurs services et à chaque fois elle utilise les mêmes. Ces activités correspondent à une logique métier atomique qui correspond à la composition de plusieurs services. Le lien entre les activités et les services peut être déterminé automatiquement à partir de la matrice présentée au Tableau 4.8, qui décrit pour chaque activité les informations utilisées et leurs taux de présence. Le Tableau 4.12, présente

les différentes activités candidates à être des services centrés-tâche, et pour chacune d'elles, les services qu'elle utilise.

Tableau 4.12 Candidats de services centrés-tâche

Tâche métier	Service centré-tâche	Service impliqué	Description
Traiter transaction	TraitementTransaction	Compte/GenererAvis	Ce service permet de traiter une transaction bancaire suite à une instruction du client ou une instruction de placement. Une transaction consiste à faire un retrait ou un dépôt.
Préparer un plan d'action	PreparationPlanAction	Client/PlanAction	Ce service assure la préparation d'un plan d'action pour un client en considérant ses informations.
Vérifier la situation du compte	VerificationCompte	Compte/Client/PlanAction	Vérification complète de la situation du compte.
Transmettre un avis sur les instructions de placements refusés	TransAvisRefus	InstructionPlacement/GenererAvis	Transmettre un avis de refus sur les instructions de placement non admises.
Prendre RDV	PriseRDV	PlanAction/RDV	Lors d'une prise de rendez-vous, l'agent fait, avec le client, les changements nécessaires sur le plan d'action et précise les informations sur le rendez-vous (date, heure, agent, etc.).
Proposer produits	PropositionProduits	Client/PlanAction	Ce service permet de donner une liste de produits à proposer au client en tenant en compte le plan d'action réalisé avec ce dernier.

Cette phase d'identification des services centrés-tâche est complétée par la réalisation d'un graphe de dépendance qui décrit les liens entre les services centrés-tâche et les services auxquels ils font appel (Figure 4.3). Une liste complète des graphes de dépendances peut être trouvée dans l'Annexe VI.

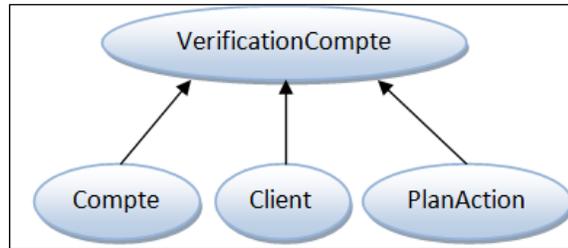


Figure 4.3 Diagramme de dépendance pour le service « VerificationCompte ».

Une fois que l'identification des services centrés-tâche et des services qui les composent est terminée, il est nécessaire de définir la logique métier qui va être encapsulée par chaque service centré-tâche.

4.6.2 Définir la logique des candidats de service centrés-tâche

Au cours de cette étape, nous allons décrire à l'aide de diagrammes de séquence, la logique métier que doit encapsuler un service centré-tâche. Ce diagramme permettra de représenter les différentes interactions réalisées au sein du service centré-tâche. Afin d'illustrer cette étape, nous allons prendre comme exemple le service « PriseRDV » qui assurera la prise d'un rendez-vous avec un client de la banque.

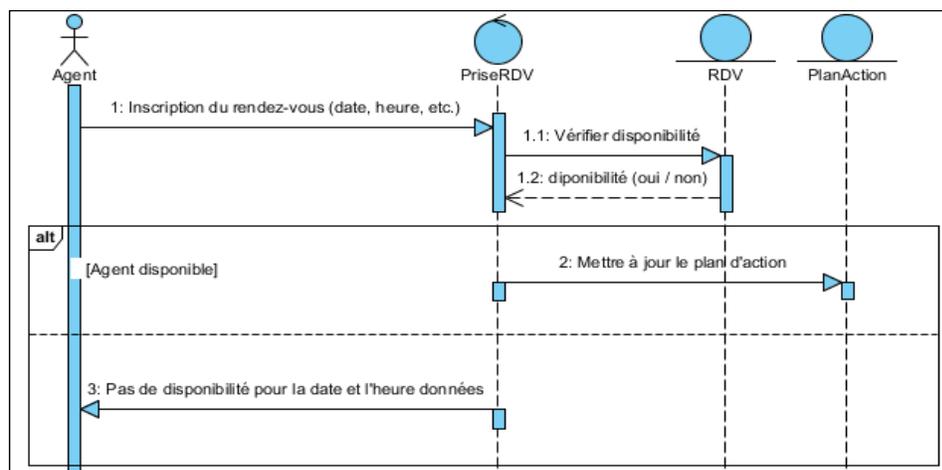


Figure 4.4 Diagramme de séquence pour le service « PriseRDV ».

Le diagramme de séquence de la Figure 4.4 illustre les échanges réalisés au sein du service « PriseRDV ». Un agent bancaire spécifie au service « PriseRDV » toutes les informations reliées au rendez-vous (date, heure, client, etc.). Ce dernier (le service) commence par vérifier la disponibilité de l'agent. Si l'agent est disponible pour cette date, alors il y aura mise à jour du plan d'action avec le client. Sinon l'agent sera avisé de la non-disponibilité pour la date et l'heure données. Annexe VII illustre les diagrammes de séquence du reste des services centrés-tâche identifiés. La logique décrite par ces diagrammes de séquence va guider les développeurs, par la suite, dans la phase de développement des services centrés-tâche.

4.7 Définir la couche d'orchestration

Au cours de cette étape, nous allons construire la couche la plus élevée de l'architecture orientée services à mettre en place, à savoir la couche d'orchestration de services. Cette couche est le résultat de la transformation des processus d'affaires en orchestrations BPEL. Afin d'assurer une reproduction fidèle de la logique métier encapsulée par les processus d'affaires, le passage d'un modèle d'affaires à une orchestration BPEL sera complètement automatisé. Cependant, la sélection des processus d'affaires à orchestrer reste une tâche complètement manuelle qui demande une analyse approfondie de ces processus. L'identification des processus à orchestrer, pour la banque sujette d'étude, et la traduction de ces processus en orchestration BPEL, seront le sujet des deux prochaines sections.

4.7.1 Identification des processus à orchestrer

L'orchestration des processus d'affaires doit être réalisée processus par processus, en respectant un ordre de priorité bien déterminé. Cet ordre doit être spécifié en respectant certains critères qui peuvent être qualitatifs ou quantitatifs. Ces critères doivent être triés selon leur importance et en respectant les objectifs de l'entreprise. Dans notre cas de figure, nous avons comme critères :

- **La sécurité** : c'est un critère qualitatif qui décrit le niveau de sécurité appliquée à un processus donné.
- **Le coût** : c'est un critère quantitatif qui présente le coût actuel engendré par un processus d'affaires.
- **Le gain espéré** : c'est un critère quantitatif qui permet de déterminer le gain espéré suite à l'orchestration d'un processus d'affaires.
- **Le temps** : c'est un critère quantitatif lié à la durée d'un processus d'affaires.
- **Le nombre d'activités** : c'est un critère quantitatif qui présente le nombre d'activités d'un processus d'affaires.

Étant donné les critères ci-dessus, le processus qui requiert le moins de politiques de sécurités, qui est actuellement le plus couteux, le plus long, dont le gain espéré est le plus important et qui a le moins d'activités sera le plus priorisé. Afin de pouvoir choisir le processus qui répond le plus à ces critères, nous avons besoin d'utiliser une méthode d'analyse multicritères. Parmi les méthodes qui offrent ce type d'analyse, nous choisissons la méthode AHP (*Analytical Hierachical Process*) (Annexe VIII), étant donné qu'elle :

- permet de décomposer un problème complexe en modules plus simples,
- permet de définir un ordre de priorité pour chaque critère,
- fourni un moyen pour contrôler la consistance des jugements réalisés sur les critères.

Étant donné que les critères sont fixés, nous allons maintenant présenter les différentes alternatives sur lesquelles nous allons appliquer la méthode AHP. Ces alternatives ne sont autres que les processus de la banque qui sont le sujet d'une orchestration (Tableau 4.13).

Tableau 4.13 Processus d'affaire à orchestrer

Processus d'affaires	Description
Ouverture compte	Description des règles métiers et des étapes à suivre lors de l'ouverture d'un compte client.
Fermeture de compte	Fermeture d'un compte client suite à une demande de ce dernier.
Ordre d'achat de titre	Traitement d'un ordre d'achat d'un titre de placement lancé par le client.
Ordre de vente de titre	Traitement d'un ordre de vente d'un titre de placement lancé par le client.
Transaction d'achat d'un titre	Une fois validé, l'ordre d'achat sera traduit en une transaction d'achat qui doit être traitée par le système bancaire.
Transaction de vente d'un titre	Traitement d'une transaction de vente d'un titre suite à un ordre du client.

Les processus illustrés par le tableau ci-dessus vont être priorisés, via la méthode AHP, en respectant les critères fixés précédemment. Ces critères doivent être, à leur tour, pondérés (selon leur importance) par les analystes métier. La Figure 4.5 montre la matrice de pondération des différents critères de sélection. Nous utilisons comme outil d'analyse AHP, le logiciel *Expert Choice 2000*.

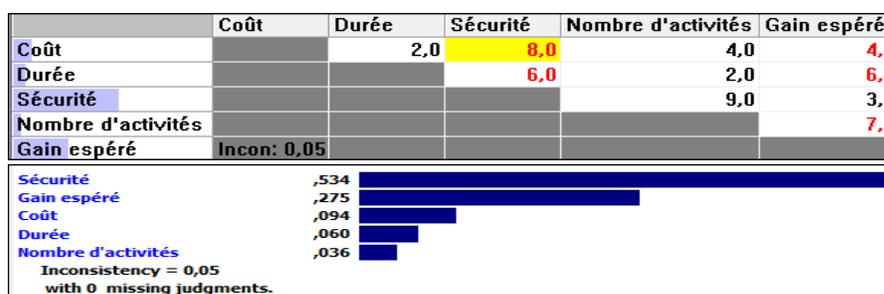


Figure 4.5 Pondération des critères de sélection.

Comme le montre la Figure 4.5, nous trouvons en premier lieu le critère de sélection « Sécurité » qui est jugé comme étant le critère le plus important. En effet, vu le domaine d'activité (financier) de l'entreprise sujette d'étude, cette dernière donne beaucoup plus d'importance à la sécurité qu'aux autres critères. Le processus qui requiert le moins de politiques de sécurité et le moins critique sera le plus priorisé. En deuxième lieu, nous

trouvons le critère de sélection « Gain espéré ». Ceci s'explique par le fait que les décideurs préfèrent commencer par l'orchestration des processus d'affaires les plus prometteurs de point de vue ROI, qui ne requièrent ni un grand effort, ni beaucoup de dépenses et de temps pour les orchestrer. En troisième et quatrième lieu, nous trouvons les critères « Coût » et « Durée », qui présentent respectivement le coût et la durée actuelle de la réalisation des processus d'affaires. Les décideurs préfèrent de commencer par orchestrer les processus dont les coûts et les durées actuels sont les plus importants, ceci afin d'assurer un meilleur ROI et réduire les délais d'achèvement des processus. En dernier lieu, nous trouvons le critère « nombre d'activités » qui est jugé le moins important vis-à-vis des autres critères. Les processus d'affaires qui présentent le moins d'activités seront les plus priorisés, étant donné qu'ils encapsulent le moins de règles d'affaires et requièrent moins d'effort pour leur orchestration. En analysant la matrice (Figure 4.5), nous remarquons un léger ratio de cohérence de 0.05 dans la pondération des décideurs, ce ratio demeure négligeable étant donné qu'il est inférieur à 0.1. Ceci nous rassure sur la consistance des jugements réalisés par les décideurs.

Une fois les critères de sélection pondérés, nous allons maintenant pondérer les différentes alternatives par rapport aux critères choisis. La Figure 4.6 présente un exemple de matrice de pondération des alternatives par rapport au critère « Sécurité ». Les matrices de pondération pour le reste des alternatives seront le sujet de l'Annexe IX.

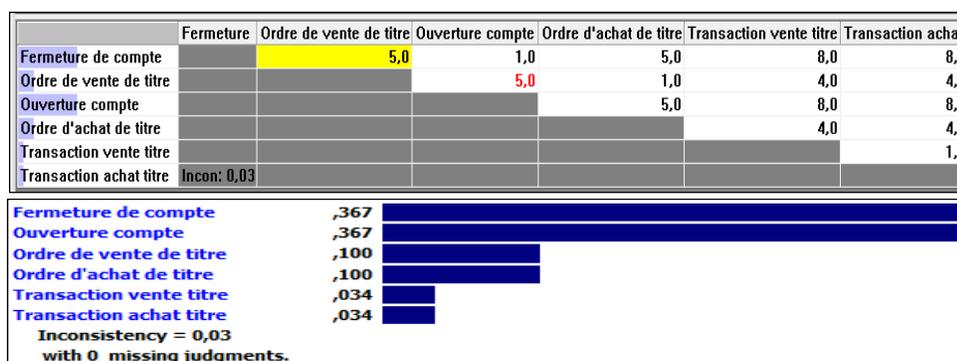


Figure 4.6 Matrice de pondération des alternatives par rapport au critère « Sécurité ».

La Figure 4.6 montre que les processus « Fermeture de compte » et « Ouverture de compte » sont les processus qui requièrent le moins de politiques de sécurité, étant donné qu'ils sont des processus réalisés en interne (au sein de la banque) et qui sont protégés par les politiques instaurés au sein de la banque. Les autres processus liés aux ventes et achats de titres sont des processus exposés à l'extérieur (ce sont des services bancaires exposés via le portail de la banque ou par téléphone), et qui requièrent l'implémentation de politiques de sécurité relativement poussées, surtout dans le cas des processus de transaction de vente et d'achat de titre. Le léger ratio de cohérence (0,03) indique que les jugements réalisés sont assez consistants.

Une fois que toutes les matrices de pondérations sont complétées, nous pouvons obtenir le vecteur de priorité globale (Figure 4.7), qui donne la priorité affectée à chaque processus. Ce vecteur est obtenu avec un ratio d'incohérence faible (de 0,04), ce qui est considéré comme acceptable (<0.10).

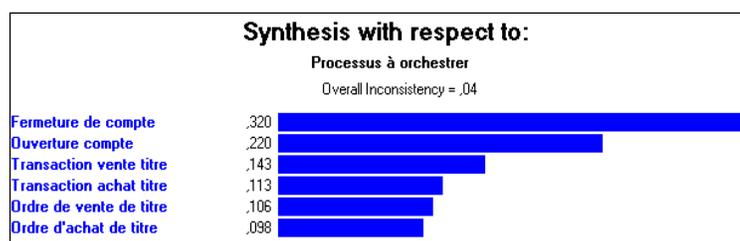


Figure 4.7 Vecteur de priorité global des alternatives de processus.

Le résultat de l'analyse multicritères, présenté par la Figure 4.7, montre que le processus de fermeture de compte est le processus le plus priorisé pour être orchestré, ceci avec un poids de 32 %. Nous trouvons ensuite le processus d'ouverture de compte, puis les processus liés aux transactions de vente et achat de titre. Enfin, nous trouverons les processus d'ordre de vente et achat de titre. Ainsi, nous avons une liste de processus à orchestrer qui suit un ordre de priorité justifié, fixé selon les objectifs de la banque sujette de l'étude.

Afin de nous assurer que l'ordre retenu, suite à l'analyse AHP, est assez robuste et fiable, nous pouvons compléter notre analyse par une analyse de sensibilité qui s'effectue en faisant varier les poids des différents critères de sélection et voir l'incidence de ces variations sur l'ordre de priorité retenue.

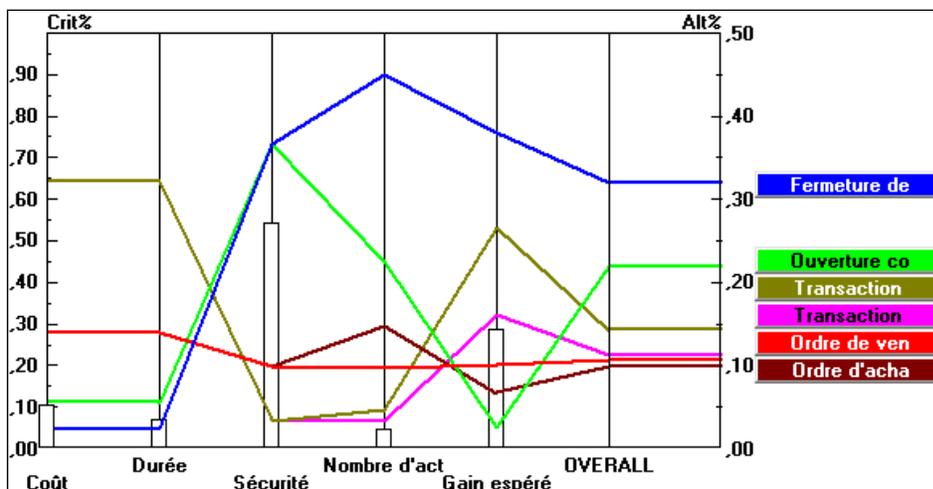


Figure 4.8 Graphe de performance pour l'analyse de sensibilité.

Étant donnée la Figure 4.8 du graphe de performance de la synthèse AHP, nous pouvons remarquer que l'ordre de priorité retenu pour les processus peut changer suite à une variation de n'importe quel critère de sélection. Cependant, cette variation varie d'un critère à un autre :

- **Coût** : une variation de 21 % (9,4 % à 30,4 %) peut changer l'ordre de priorité des processus pour favoriser le processus de « Transaction de vente de titre » sur le processus « Ouverture compte », cependant le choix de commencer par le processus de « Fermeture de compte » reste toujours valable jusqu'à une variation de 34 % (9,4 % à 43,4 %), ce qui est considéré comme variation énorme.
- **Durée** : pour ce critère, une variation de 22 % (6 % à 28 %) changera l'ordre de priorité retenue en favorisant le processus « Transaction de vente de titre » sur le processus « Ouverture compte ».

- **Sécurité** : une variation de 6 % (53,4 % à 59,4 %) pour ce critère rendra le processus « Ordre de vente » plus prioritaire que le processus « Transaction achat de titre ».
- **Nombre d'activités** : une variation au-delà de 10 % (3,6 % à 13,6 %) changera l'ordre de priorité des processus d'affaires à orchestrer en favorisant le processus « Ordre de vente de titre » sur le processus « Transaction d'achat de titre ».
- **Gain espéré** : ce critère peut changer l'ordre de priorité d'orchestration des processus à partir d'une variation de 18,6 % (27,5 % à 46,1 %).

En résumé, nous pouvons dire que l'ordre de priorité retenue suite à l'analyse multicritère est assez robuste et fiable, étant donné que cet ordre est peu sensible à la majorité des critères, sauf au critère de « Sécurité » auquel il est moyennement sensible. En effet, un changement d'avis des décideurs de 6 % pourrait changer l'ordre de priorité d'orchestration des services. Dans notre cas, les décideurs sont assez confiants de leur pondération initiale pour le critère de sécurité, étant donné que la sécurité est le facteur le plus important pour les banques. Ainsi, les processus d'affaires seront orchestrés selon l'ordre mentionné par le Tableau 4.14, qui spécifie pour chaque processus le service d'orchestration correspondant.

Tableau 4.14 Ordre d'orchestration des processus d'affaires et les services d'orchestration identifiés

Processus d'affaires	Service d'orchestration
Fermeture de compte	FermetureCompte
Ouverture compte	OuvertureCompte
Transaction de vente d'un titre	TransactionVenteTitre
Transaction d'achat d'un titre	TransactionAchatTitre
Ordre de vente de titre	OrdreVenteTitre
Ordre d'achat de titre	OrdreAchatTitre

4.7.2 Orchestration des processus d'affaires

Au cours de cette étape, nous allons orchestrer les processus recensés au niveau de l'étape précédente, tout en respectant l'ordre de priorité fixé suite à l'analyse multicritère. Une telle orchestration va se réaliser en deux étapes :

- La transformation du modèle métier en un MI.
- La génération du code BPEL de l'orchestration à partir du MI.

Ces deux étapes seront le sujet des deux prochaines sections.

4.7.2.1 Transformation d'un modèle métier en un modèle intermédiaire

Rappelons qu'un MI est un modèle qui représente un processus d'affaires de point de vue TI. La Figure 4.9 présente le MI du processus d'ouverture d'un compte client. Cette figure présente la séquence d'activités à suivre pour créer un compte bancaire et les échanges réalisés entre les systèmes informatiques pour mener cette création. Le processus de création d'un compte client consiste d'abord (Figure 4.10), à collecter et à vérifier les informations fournies par le client. Une fois que ses informations sont validées, un document « Plaintes client » sera imprimé et expliqué pour le client. Ce document présente la procédure que le client doit suivre en cas où il veut faire une plainte auprès de la banque. Ensuite, le compte sera créé et un avis d'ouverture de compte sera généré. Les informations liées à la demande d'ouverture de compte seront le sujet d'un contrôle de qualité de saisie, ceci pour des fins d'optimisation du processus en question. Une fois que le MI est conçu, il ne reste qu'à générer le code BPEL assurant son orchestration.

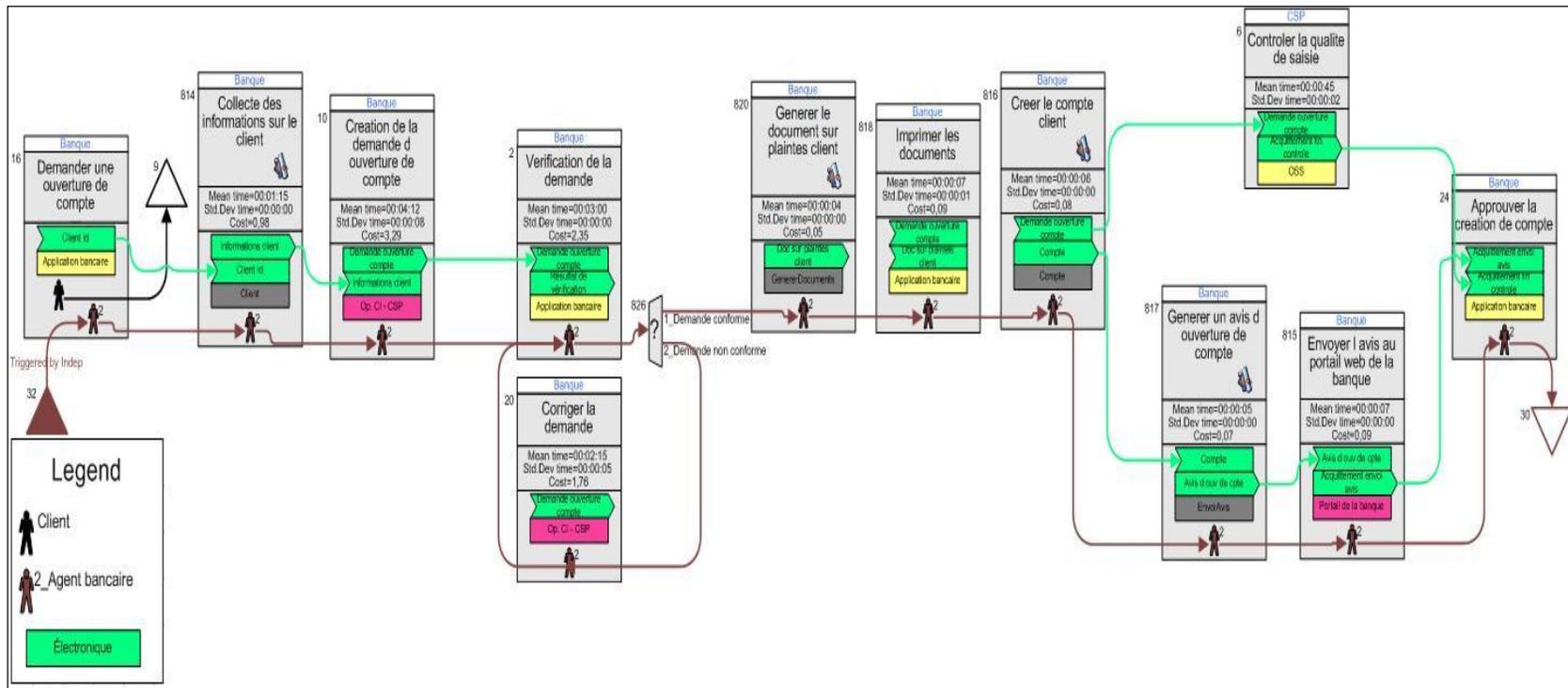


Figure 4.9 Modèle intermédiaire de processus de création d'un compte bancaire

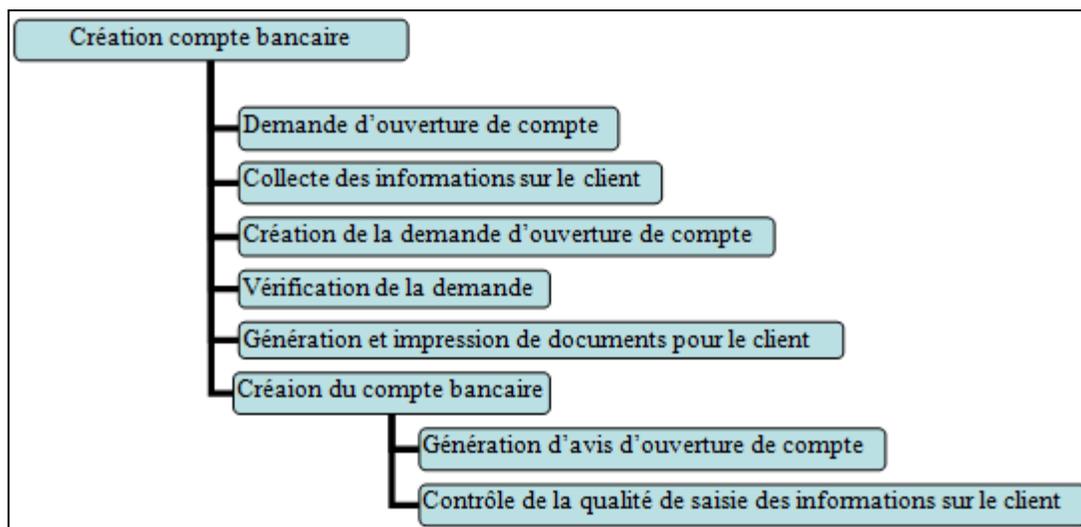


Figure 4.10 Organigramme du processus de création d'un compte bancaire.

4.7.2.2 Génération automatique d'un BPEL à partir d'un modèle intermédiaire

Arrivant à ce stade, nous avons un MI qui décrit un processus d'affaires de point de vue TI. À partir de ce modèle, nous allons générer automatiquement un code BPEL au niveau duquel nous allons reproduire toute la logique métier encapsulée. La génération automatique est réalisée en deux étapes. La première étape consiste à analyser structurellement le MI et générer une arborescence qui représente les différentes structures y présentes (boucles, branches décisionnelles, etc.), ayant ainsi une structure proche du code BPEL à générer. La Figure 4.11 décrit l'arborescence liée au processus de création d'un compte client, au niveau duquel nous trouvons les différentes étapes et règles métiers amenant à cette création.

Suite à la génération de l'arborescence, l'analyste doit d'abord, choisir le type du code BPEL à générer (BPEL abstrait ou BPEL exécutable), ensuite valider son choix et lancer le processus de génération du code BPEL. Ainsi, une orchestration va se générer en prenant en considération les informations fournies sur les différentes ressources utilisées (les systèmes informatiques, les services Web, les informations, leurs types d'usage, etc.)

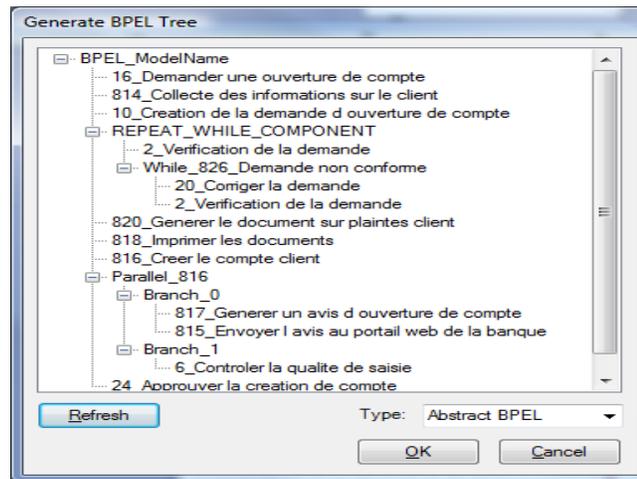


Figure 4.11 Arborescence de génération d'un code BPEL.

Le code généré peut être exporté directement dans un éditeur de BPEL, afin de le valider. Ce code présente le squelette de l'orchestration à déployer sur un moteur d'orchestration. En effet, le fichier BPEL généré présente un patron que les développeurs doivent respecter tout au long du processus de création de l'orchestration. Ce patron facilite la tâche aux développeurs en leur reproduisant toutes les règles métier et les informations sur les ressources utilisées. Cependant, il reste toujours des détails techniques à ajouter qui ne sont pas présents dans les processus d'affaires et qui sont nécessaires pour le déploiement d'une orchestration. La Figure 4.12 présente un aperçu de l'orchestration générée automatiquement à partir du MI illustrant le processus de création de compte client (nous utilisons comme éditeur BPEL, l'outil Visual Studio² de Microsoft³).

² Visual Studio : Ensemble d'outils de développement conçus par Microsoft

³ Microsoft : Une multinationale américaine spécialiste dans le développement de solutions informatiques.

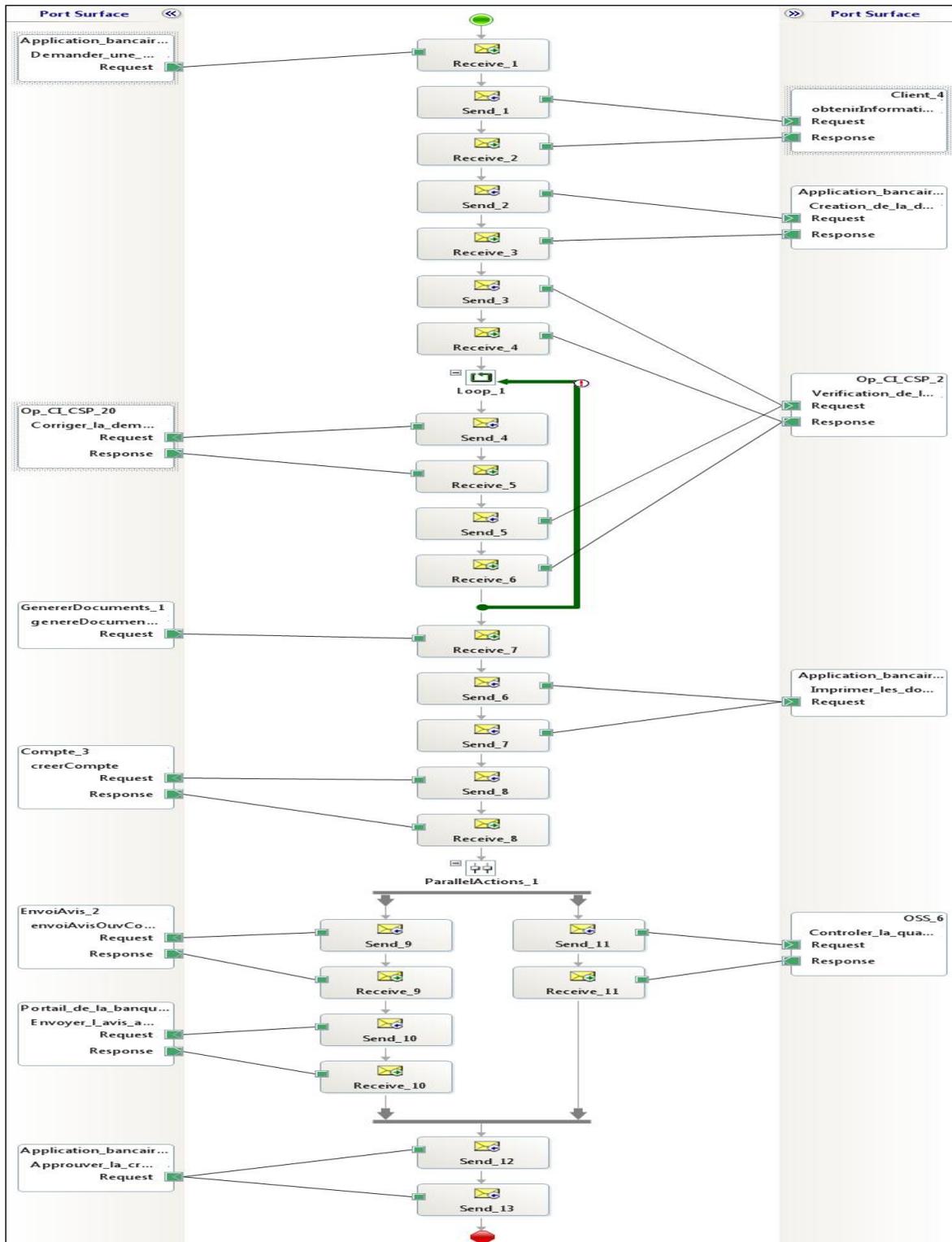


Figure 4.12 Orchestration du processus de création d'un compte bancaire.

Arrivons à ce stade, nous avons pu identifier les services les plus commodes aux besoins de la banque sujette de l'étude. Le Tableau 4.15 décrit la liste de tous les services que nous avons pu identifier par le biais de l'approche que nous proposons.

Tableau 4.15 Liste complète des services identifiés

	Service	Description
Services d'orchestration	OuvertureCompte	Description des règles métiers et des étapes à suivre lors de l'ouverture d'un compte client.
	FermetureCompte	Fermeture d'un compte client suite à une demande de ce dernier.
	OrdreAchatTitre	Traitement d'un ordre d'achat d'un titre de placement lancé par le client.
	OrdreVenteTitre	Traitement d'un ordre de vente d'un titre de placement lancé par le client.
	TransactionAchatTitre	Traduire l'ordre d'achat en une transaction qui doit être traitée par le système bancaire.
	TransactionVenteTitre	Traitement d'une transaction de vente d'un titre suite à un ordre de vente d'un titre de placement
Services centrés-tâche	TraitementTransaction	Traiter une transaction bancaire due à une instruction du client ou une instruction de placement. Une transaction consiste à faire un retrait, ou un dépôt.
	PréparationPlanAction	Préparer un plan d'action pour un client en tenant compte de ses informations.
	VérificationCompte	Vérification complète de la situation du compte.
	TransAvisRefus	Transmettre un avis de refus sur les instructions de placement non admises.
	PriseRDV	Prendre un rendez-vous, et mettre à jour le plan d'action.
	PropositionProduits	Retourner une liste de produits à proposer au client en tenant compte du plan d'action réalisé avec ce dernier.
Services entité	Compte	Gestion des comptes clients de la banque.
	Placement	Gestion des placements et des fonds admissibles pour un placement donné.
	Client	Gestion des informations liées aux clients de la banque.
	PlanAction	Gestion et suivi des plans d'action pour un client donné.
	RDV	Gestion des rendez-vous avec les clients.
	InstructionPlacement	Gestion et vérification des instructions de placement.
Services d'application	GenererAvis	Génération d'avis (courriel, document électronique, etc.), qui seront échangés entre différents acteurs et systèmes informatiques. Ces avis seront générés à partir des données stockées sous un format native (XML, base de données, etc.).
	Gestion SystemeTelephonique	Gestion et traitement des transactions téléphoniques faites par le client.
	GenererDocuments	Génération de documents dont le contenu sera généré automatiquement à partir de données stockées sous un format native.

4.8 Conclusion

Au cours de ce chapitre, nous avons pu valider notre approche en l'appliquant sur une banque offrant différents services financiers. L'analyse des processus d'affaires de cette dernière nous a permis d'identifier 21 services, constituant une architecture orientée services multiniveau, allant de la couche d'application à la couche d'orchestration.

Les services entités et les services applications, ainsi que leurs opérations sont identifiées à partir des informations clés et des activités les plus sollicitées dans la banque, assurant ainsi un haut degré de réutilisabilité. Ce critère est moins respecté pour les services centrés-tâche et les services d'orchestration, en vue de donner plus d'importance à la logique métier. Tout au long du processus d'identification de services, nous vérifions toujours qu'il n'existe pas de chevauchement fonctionnel entre les différentes opérations, ceci afin de respecter le principe d'autonomie de SOA. Les services et les opérations identifiées ont été le sujet d'une description détaillée des fonctionnalités qu'elles offrent. Ceci, afin de faciliter la découverte des services et permettre aux utilisateurs de choisir le service qui leur convient le plus.

Tout au long de ce chapitre, nous avons pu justifier tous nos choix et faire recours à des outils d'analyse pour l'aide à la décision telle que la méthode AHP d'analyse multicritères.

CONCLUSION GENERALE

L'adoption d'une architecture orientée services permet de réaliser le rêve de toute entreprise voulant accroître son agilité, argumenter sa productivité et réduire ses coûts. SOA est un concept qui commence à gagner du terrain et entrer dans la pratique des directions informatiques au sein des organisations, bien que les approches de mise en place de ce type d'architectures divergent encore.

La réussite d'un projet SOA passe par une identification justifiée des services à mettre en place. Une telle identification doit assurer à l'entreprise un retour sur l'investissement le plus rapidement possible. Pour atteindre cette fin, nous avons procédé à la conception d'une approche systématique d'identification des services les plus appropriés à une entreprise en nous basant sur l'analyse statistique de ses processus d'affaires. L'approche proposée, organisée en directives (étapes), représente une feuille de route pour les analystes SOA. Chaque étape de cette approche est marquée par un ensemble d'artefacts à valider avant de passer à l'étape suivante.

En vue de tester la totalité de l'approche proposée, nous avons procédé à appliquer cette dernière sur un cas réel de banque d'affaires, composée de plusieurs branches qui offrent différents types de produits et services financiers.

Enfin, notre travail a pu couvrir la majorité des critères d'identification des services les plus commodes à une entreprise tout en essayant d'automatiser la plupart des étapes amenant à cette identification, offrant ainsi un haut degré d'automatisation du processus d'identification de services.

L'approche proposée dans ce travail permet une identification justifiée et automatisée (dans la plus part de ces étapes) des services les plus appropriés aux besoins de l'entreprise.

L'application de la présente approche sur un cas réel nous permet de dégager quelques recommandations qui rentrent dans le cadre d'améliorations, et cela tout comme :

- La mise en place d'une liste exhaustive de critères à valider en vue de garantir un minimum de qualité de modélisation des processus d'affaires. En effet, la qualité des résultats retournés par la présente approche dépend principalement de la qualité et l'exactitude de la modélisation des processus d'affaires de l'entreprise en question. Ces résultats sont autant pertinents et exacts que les processus modélisés correspondent vraiment au processus réel.
- La mise en place d'un ensemble de recommandations qui aident l'entreprise à définir l'ordre de priorité de modélisation de ses processus d'affaires en cas où cette dernière n'a pas encore modélisé ses processus. Malheureusement, il existe plusieurs entreprises qui n'ont pas encore le niveau de maturité et la pratique de dresser une cartographie complète et globale de leurs processus d'affaires, sous prétexte que c'est très coûteux et sans valeur ajoutée.
- Une analyse statistique plus poussée pour le choix des seuils (à partir desquels une information ou une activité est considérée comme récurrente) et l'ajout d'autres types de graphiques pour améliorer la qualité des résultats trouvés.
- Une analyse ascendante de la couche applicative de l'entreprise afin compléter l'analyse descendante réalisée au cours de ce travail. Cette analyse permettrait une meilleure identification des services d'applications, qui constituent la couche de base d'une SOA.
- L'extension de cette approche de façon à ce qu'elle intervienne dans les phases de conception et de développement orienté-services du cycle de vie de SOA.

ANNEXE I

LISTE DES ACTIVITÉS AUTOMATIQUES ET SEMI-AUTOMATIQUES

Nom de l'activité	Nombre d'occurrences
Apporter les correctifs au besoin	2
Apporter les corrections	10
Approuver instruction placement	19
Attribuer le numéro de compte	1
Autoriser la nouvelle transaction	4
Autoriser le FOC	4
Autoriser le FOC ou MAJ	1
Autoriser le nouveau FOC	4
Autoriser le paiement	2
Autoriser l'émission du chèque	2
Autoriser l'ouverture ou MAJ	1
BO fait la correction de la transaction	2
Confirmer la transaction	9
Confirmer l'instruction avec le client	2
Confirmer ou modifier l'instruction avec le client	2
Consigner les attentes du membre et les résultats de la rencontre	2
Consigner les attentes et les résultats	1
Consigner les changements dans la situation du membre	3
Consigner les notes au dossier	1
Contrôler la qualité de saisie dans OSS	2
Convenir avec le membre de la répartition d'actifs cible	1
Créer l'investisseur et le compte	5
Déclencher l'impression du FOC et signer	2
Déposer le montant	5
Enregistrer le retrait et remarques	2
Établir le profil de l'investisseur	3
Faire une référence	1
Finaliser la transaction pour exécution	1
Finaliser le rapport de diversification de portefeuilles	1
Générer les formulaires	2
Générer les simulations	1
Imprimer la demande	1
Imprimer le FOC et les contrats	2
Imprimer les documents sur les risques et les plaintes	3
Imprimer l'état des actifs du compte	1
Indiquer le code de gel au compte	2
Initier la demande de paiement et confirmer caisse	2
Initier un suivi au plan d'action	3
Initier un suivi au plan d'instructions	1
Inscrire l'accusé de réception	2

Inscrire les remarques au cte	2
Inscrire une indication de gel au compte	1
Mettre à jour Expert - Connaître son client	1
Mettre à jour les informations	3
Mettre en place les suivis du plan d'action	1
Modifier le plan d'action ou rapport Connaître son client	3
Modifier le statut de la transaction à Approuver	2
Modifier OSS	1
Ouvrir le compte	2
Prendre R-V	4
Préparer le dossier en vue de la rencontre avec le membre	2
Préparer un plan d'action	1
Présenter la solution et l'offre	1
Produire le rapport de transactions manuelles	2
Produire les formulaires	4
Produire un avis au conseiller	3
Proposer produits	2
Rechercher le numéro de code du titre	8
Récupérer les documents de l'offre	3
Récupérer les informations de l'offre	1
Refaire la convenance et documenter	1
Refuser le FOC	4
Refuser la MAJ du compte	1
Remettre le formulaire sur l'utilisation de l'effet levier	2
Remplir et expliquer le formulaire C-104	1
Remplir le formulaire d'adhésion	2
Remplir manuellement et signer la Convention d'épargnant	1
Remplir nouveau FOC	1
Retrait somme	6
Réviser le profil d'investisseur	1
Réviser les informations	1
Saisir et imprimer les instructions de placement dans FD	1
Saisir la transaction	11
Saisir le retrait et documenter	2
Saisir le retrait REEE	1
Saisir les changements FOC	2
Saisir les informations de l'adhésion dans Fonds Desjardins	4
Saisir les informations de l'ouverture de compte	2
Saisir les instructions de placement dans FD	3
S'approprier la référence en vue de la rencontre	1
Saisir la répartition par défaut au CIQ	2
Soumettre la demande d'ouverture de compte	1
Traitement des dépôts	2
Traitement monétaire par SIC	2
Traitement de l'instruction de placement	6
Traiter la demande gel ou renversement	2

Traiter le dépôt manuellement	2
Traiter l'instruction	1
Traiter transaction	1
Transcrire le numéro de compte sur les formulaires	1
Transmettre un avis au conseiller	4
Transmission de la demande à Winfund	2
Transmission Winfund	1
Vérifier et analyser les informations du FOC	4
Vérifier et analyser les informations du FOC ou MAJ	1
Vérifier la convenance	2
Vérifier la convenance de la transaction	18
Vérifier la demande	1
Vérifier la disponibilité de l'agent	5
Vérifier la situation du compte	2
Vérifier la transaction	4
Vérifier la transaction corrigée	2
Vérifier les instructions téléphoniques	5
Vérifier l'admissibilité du titre	6
Vérifier le FOC	2
Vérifier le formulaire et le solde à l'encaisse	2
Vérifier le formulaire et transaction SIPC	2
Vérifier les données du membre et le compte	1
Vérifier les éléments de l'offre au besoin	9
Vérifier les infos socio-démographiques du membre	1
Vérifier l'offre Expert	8
Vérifier sommairement le FOC	3

ANNEXE II

OCCURRENCES DES INFORMATIONS MÉTIERS

Informations	Occurrences
Acceptation de l'offre	5
Analyse besoins GA	6
Avis de Mise à jour de cpte	8
Avis de refus	12
Avis de révision FOC	1
Avis d'ouv. de cpte	32
C-104	1
C-107	2
C-124, 137, 139, 162	2
Caract. rais. repar actif	1
Chèque	2
Code de titre	8
Compte	1
Confirmation (nom et hre)	6
Contrat d'adhésion	13
Convention d'épargnant	1
Décision Winfund	1
Decl op non sollicité	1
Demande de correction	4
Demande de corrections	3
Demande de paiement	10
Demande gel de compte	3
Demande nouveau FOC	5
Demande ouverture compte	13
Dépôt direct	4
Dépôt SIPC	4
Détail des retraits REE	1
Détails du retrait	2
Documents légaux	1
Dossier du membre	9
Dossier épargne collect.	10
enregistrement de DD	2
Enregistrement vocal	6
FOC NUM	4
Form instr. Placement	28
Form. d'adhésion coll.	4
Formulaire de transf REID	12
Formulaire MAJ	1
Formulaire ouv. compte	55
inscription r-v	4

Instruct. à l'agent	2
Instructions de placement	79
Instructions téléphonique	18
Liste de numéro de compte	1
Liste des fonds adm.	10
Mise à jour FOC	7
Modèle Notes	4
Notes au dossier client	15
Numéro de compte	11
Numéro de confirmation	3
Numéro d'investisseur	5
Offre serv. div. PF	3
Photo pièce d'identité	7
Plan d'action	17
Plan d'instructions	5
Profil d'investisseur	4
Rap de trx de fonds place	2
Rap div portefeuille	2
Rap. connaître son client	26
Rapp. T+1 CI ON	4
Rapport SIPC	4
Référence	2
Relevé des Fonds Desj.	3
Relevés pension et autres	1
Rens. sur plaintes client	10
Risques liés à un emprunt	12
SIPC	2
Suivi au plan d'action	8
Suivi plan d'instruction	8
Transaction	10

ANNEXE III

LISTE DES ACTIVITÉS LES PLUS SOLLICITÉES

Activités métiers	Nombres d'occurrence
Apporter les corrections	10
Approuver instruction placement	19
Autoriser la nouvelle transaction	4
Autoriser le FOC	4
Autoriser le nouveau FOC	4
Confirmer la transaction	9
Créer l'investisseur et le compte	5
Déposer le montant	5
Prendre R-V	4
Produire les formulaires	4
Rechercher le numéro de code du titre	8
Refuser le FOC	4
Retrait somme	6
Saisir la transaction	11
Saisir les informations de l'adhésion dans Fonds Desjardins	4
Traitement de l'instruction de placement	6
Transmettre un avis au conseiller	4
Vérifier et analyser les informations du FOC	4
Vérifier la convenance de la transaction	18
Vérifier la disponibilité de l'agent	5
Vérifier la transaction	4
Vérifier les instructions téléphoniques	5
Vérifier l'admissibilité du titre	6
Vérifier les éléments de l'offre au besoin	9
Vérifier l'offre Expert	8

ANNEXE IV

DESCRIPTION DES OPÉRATIONS BASIQUES DES SERVICES ENTITÉ

Service entité	Opération	Description
Compte	creerCompte	Créer un compte bancaire en spécifiant toutes les informations qui le concerne : type de compte, produits liés, le ou les détenteurs du compte, etc.
	modifierCompte	Modifier les informations concernant un compte donné
	obtenirDetailCompte	Obtenir les informations sur un compte : son ou ses détenteurs, numéro de compte, etc.
Placement	creerPlacement	Créer un placement en spécifiant toutes les informations qui le concernent : type de placement, taux d'intérêt, termes, etc.
	modifierPlacement	Modifier les informations liées à un placement donné
	obtenirDetailPlacement	Obtenir les informations liées à un placement
	obtenirListeFondsAdmissibles	Obtenir la liste des fonds de placement admissibles de la banque
Client	creerInformationsClient	Créer un client en spécifiant toutes les informations qui lui sont liées
	modifierInformationsClient	Modifier les informations liées à un client
	obtenirInformationsClient	Retourner les informations liées à un client
	inscrireNotes	Ajouter des notes dans le dossier d'un client
	modifierNotes	Modifier les notes affectées à un client
	obtenirNotes	Retourner les notes concernant un client donné
	creerDossierEpargneCollective	Créer un dossier d'épargne collective pour un client
modifierDossierEpargneCollective	Modifier les informations liées au dossier d'épargne collective d'un client	
Plan d'action	créerPlanAction	Créer un plan d'action pour un client
	modifierPlanAction	Modifier les informations qui portent sur un plan d'action
	obtenirPlanAction	Obtenir le détail d'un plan d'action
	initierSuivi	Initier un suivi d'un plan d'action
	modifierSuivi	Modifier les informations liées à un suivi d'un plan d'action
	obtenirDetailSuivi	Obtenir les informations sur un suivi d'un plan d'action

Rendez-vous	fixerRDV	Fixer un rendez-vous avec le client en spécifiant l'heure, la succursale, l'agent avec lequel il y aura le rendez-vous, etc.
	modifierRDV	Modifier les informations liées à un rendez-vous donné
	obtenirDetailRDV	Obtenir le détail d'un rendez-vous
Instruction de placement	creerInstructionPlacement	Créer une instruction de placement lancée par un investisseur (ou client)
	modifierInstructionsPlacement	Modifier une instruction de placement
	obtenirInstructionsPlacement	Obtenir le détail d'une instruction de placement

ANNEXE V

DESCRIPTION DÉTAILLÉE DES OPÉRATIONS DES SERVICES D'APPLICATION

Services application	Opération de service	Description
EnvoiAvis	envoiAvisRefus	Génération et envoi d'un courriel pour aviser un agent et/ou un client du refus d'une action (ouverture compte, placement, etc.).
	envoiAvisOuvMAJCompte	Génération et envoi d'un message électronique via le portail de la banque pour aviser un agent de l'ouverture ou la mise à jour d'un compte.
	envoiAvisMajFOC	Génération et envoi d'un message électronique via le portail de la banque pour aviser un agent d'un changement dans les informations d'un formulaire d'ouverture de compte
GestionSystemeTelephonique	creerConfirmationTransaction	Créer et encoder une confirmation vocale d'une instruction téléphonique en incluant le nom de l'agent et l'heure de confirmation
	lireConfirmationTransaction	Décoder et retourner confirmation vocale afin de pouvoir la consulter
	enregistreCommunicationTel	Encoder et enregistrer une communication téléphonique
	lireCommunicationTel	Décoder et retourner une communication téléphonique enregistrée afin de pouvoir la consulter
	enregistrerInstructionTel	Encoder et enregistrer une instruction téléphonique effectuée par le client
	lireInstructionTel	Décoder et retourner une instruction téléphonique
GenererDocuments	genereDocumentPlaintesClientPDF	Générer un PDF à partir des données stockées sous un format natif qui décrit le processus à suivre par un client s'il veut faire une plainte auprès de la banque. Ce PDF va être par la suite imprimé
	genererRisquesEmpruntPDF	Générer un PDF à partir des données stockées sous un format natif qui explique les risques liés à un emprunt bancaire. Ce PDF va être par la suite imprimé.

ANNEXE VI

DIAGRAMMES DE DEPENDENCE POUR LES SERVICES CENTRÉS-TÂCHE

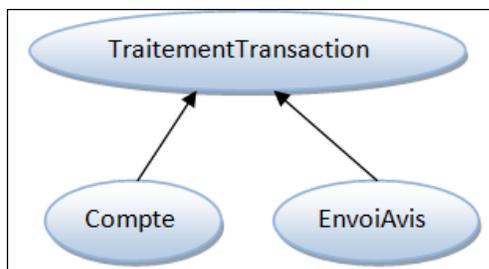


Figure-A 0-1 Diagramme de dépendance pour le service « TraitementTransaction ».

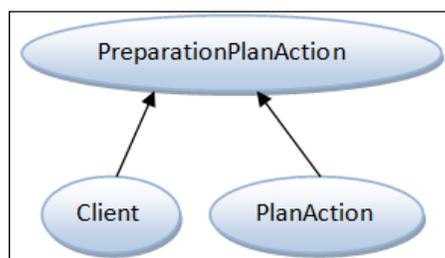


Figure-A 0-2 Diagramme de dépendance pour le service « PreparationPlanAction ».

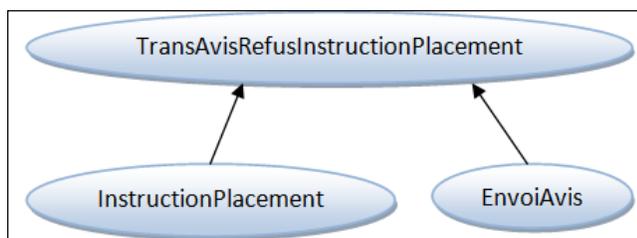


Figure-A I0-3 Diagramme de dépendance pour le service « TransAvisRefusInstructionPlacement ».

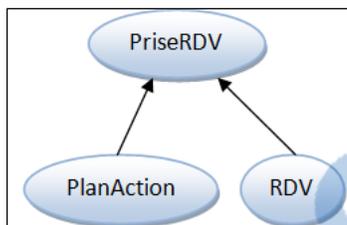


Figure-A 0-4 Diagramme de dépendance pour le service « PriseRDV ».

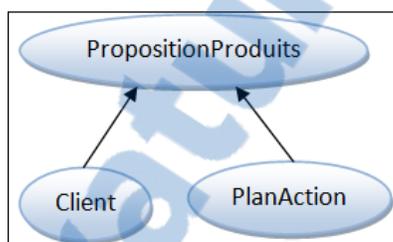


Figure-A 0-5 Diagramme de dépendance pour le service « PropositionProduits ».

ANNEXE VII

DIAGRAMMES DE SÉQUENCE DES SERVICES CENTRÉS-TÂCHE

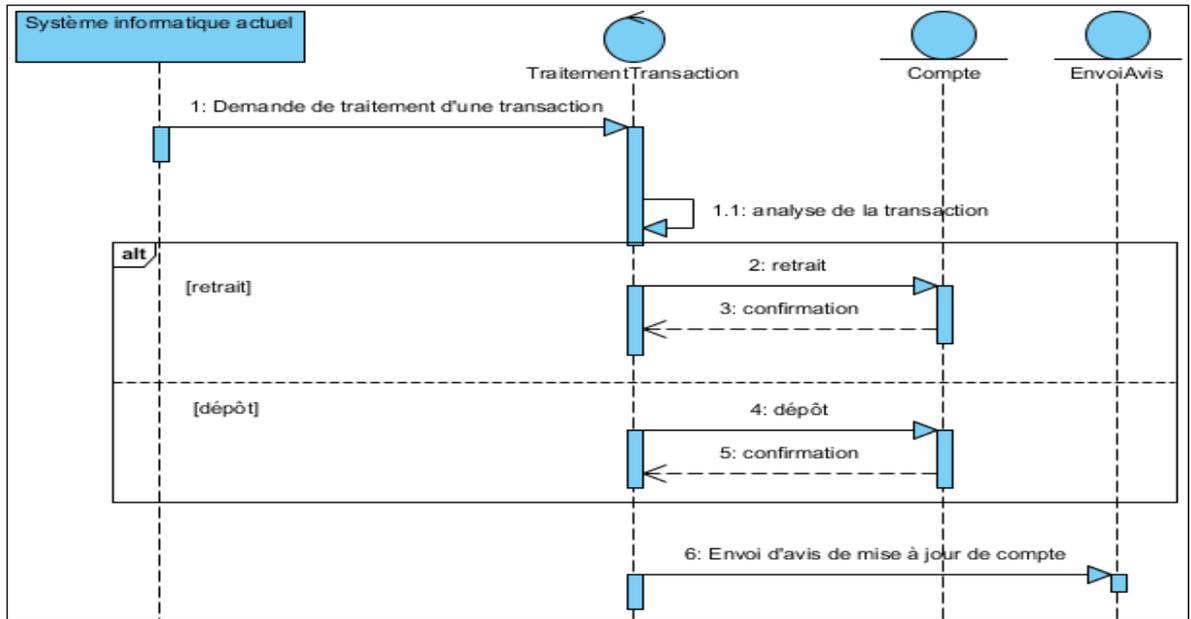


Figure-A VII-1 Diagramme de séquences pour le service « TraitementTransaction ».

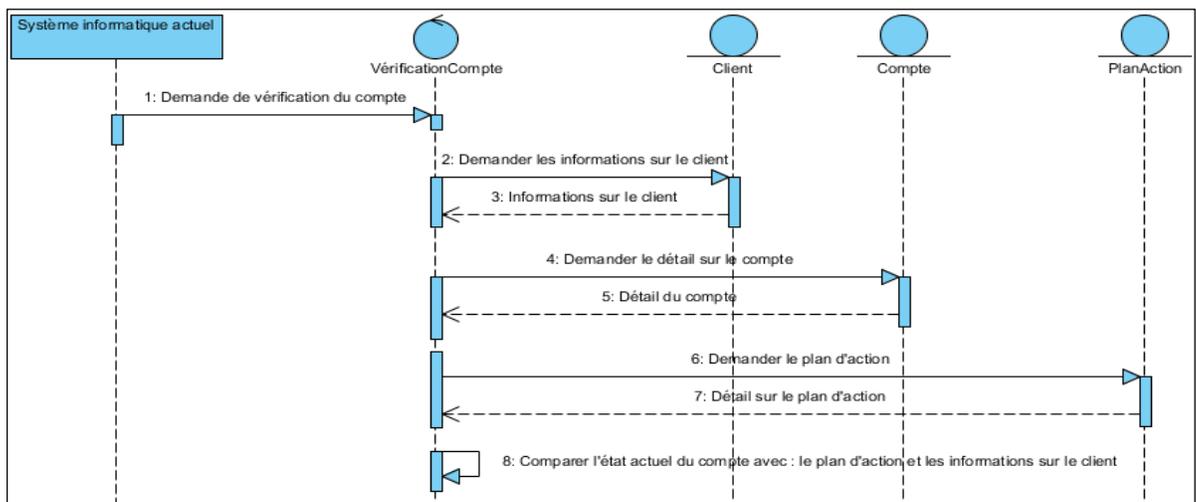


Figure-A VII-2 Diagramme de séquences pour le service « VérificationCompte ».

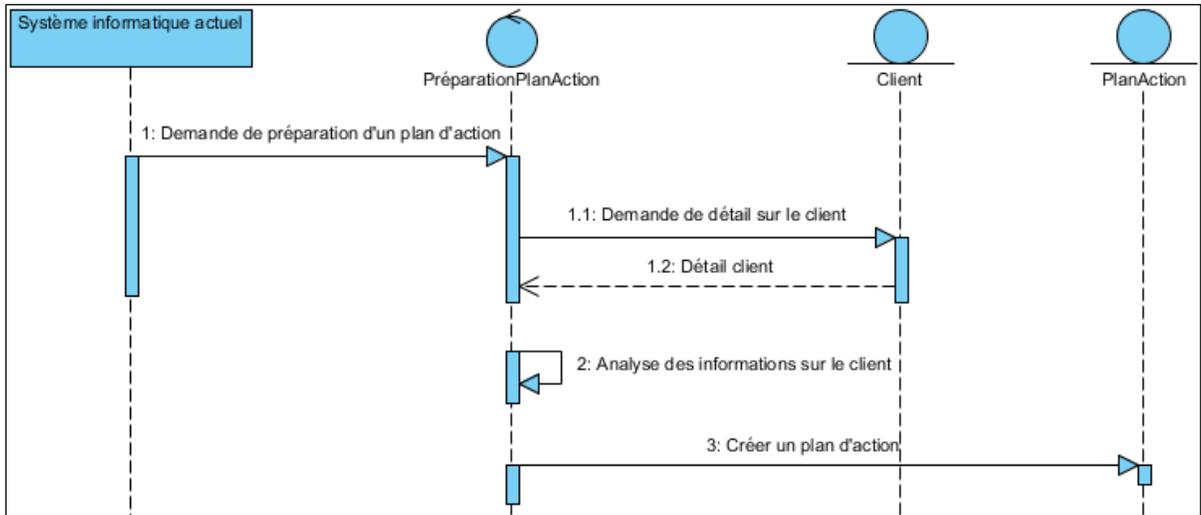


Figure-A VII-3 Diagramme de séquences pour le service « PréparationPlanAction ».

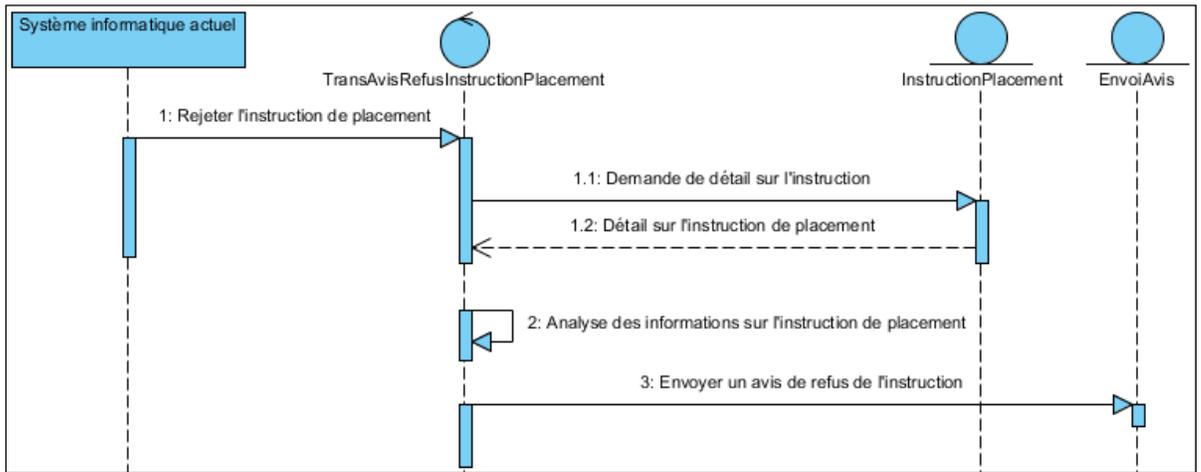


Figure-A VII-4 Diagramme de séquences pour le service « TransAvisRefusInstructionPlacement ».

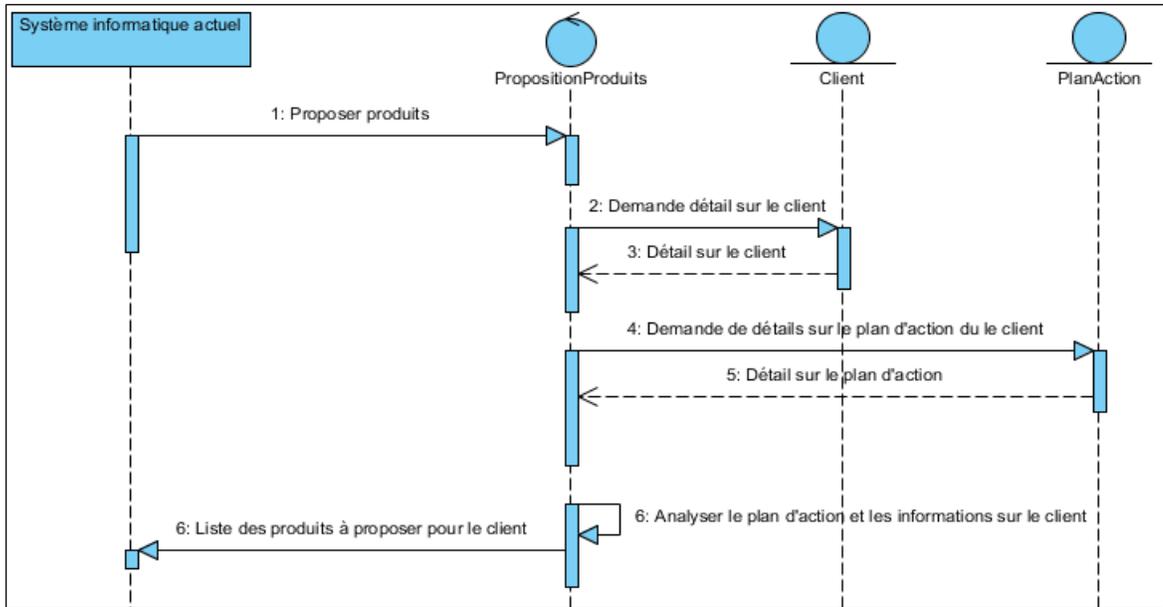


Figure-A VII-5 Diagramme de séquences pour le service « PropositionProduits ».

ANNEXE VIII

ANALYTICAL HIERARCHY PROCESS (AHP)

AHP (Analytical Hierarchy Process) est une méthode d'analyse qui rentre dans le cadre des méthodes d'aide à la décision multicritère. C'est une méthode qui se base sur des analyses mathématiques et psychologiques afin d'aider les décideurs à choisir l'alternative qui convient le plus à leurs besoins et à leur compréhension du problème. Développée en 1971 par Thomas L. Saaty, la méthode AHP est basée principalement sur quatre principes (Saaty, 2008) :

- **Décomposition** : Décomposer un problème décisionnel complexe et le structurer hiérarchiquement. Chaque niveau hiérarchique sera à son tour décomposé jusqu'à atteindre un niveau de complexité acceptable. Exemple, dans une hiérarchie de N niveaux, on trouvera l'objectif cible au (niveau 0), les critères au (niveau 1) jusqu'au (niveau N-1) et les alternatives occuperont le (niveau N)

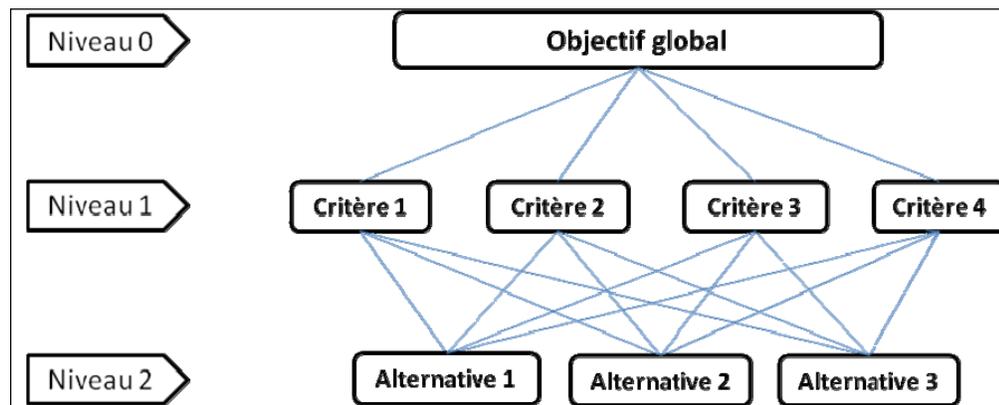


Figure-A VIII-1 Exemple de structure hiérarchique en AHP.

- **Prioritisation** : Comparer deux par deux les éléments appartenant au même niveau hiérarchique, par rapport à l'élément du niveau hiérarchique directement supérieur. Pour cette fin (Saaty, 2008) propose de réaliser une matrice carrée réciproque. Dans le cas où on a N éléments à comparer, nous aurons une matrice similaire à la matrice ci-dessous.

- **Analyse de sensibilité** : Varier les priorités attribuées aux critères et voir l'impact de ces variations sur la priorité du choix retenue, ceci afin de déterminer à quel point ce choix est robuste et fiable.
- **Cohérence des jugements** : Les jugements fournis par les décideurs présentent souvent des incohérences qui peuvent fausser les résultats d'analyse multicritère. Afin de remédier à cela, (Saaty, 2008) propose la formule ci-dessous pour calculer l'indice de consistance IC d'une matrice de jugement donnée :

$$IC = (\lambda_{\max} - N) / (N-1) \quad (4.1)$$

Avec N est le nombre d'éléments comparés et λ_{\max} la plus grande valeur propre de la matrice de comparaison. L'indice de consistance présenté par la formule ci-dessus est d'autant plus grand que les jugements sont incohérents et vis-versa. Pour vérifier si une matrice est cohérente ou non, IC est comparé à un ratio de cohérence (RC), défini par expérimentation, par (Saaty, 2008). Ce ratio est défini par l'équation (4.2)

$$RC = IC / IA \quad (4.2)$$

Avec

RC : le ratio de cohérence

IA : un indice aléatoire

IC : est l'indice de cohérence

IA est un indice aléatoire proposé par (Saaty, 2008), obtenu par simulation (Tableau-A VIII-2). Une valeur de RC inférieure à 10% est jugée comme acceptable pour dire qu'une matrice de comparaison est cohérente. Sinon les jugements effectués par les utilisateurs devront être le sujet d'une révision. La mesure de cohérence peut être utilisée pour évaluer la cohérence des décideurs ainsi que la cohérence de toute la hiérarchie.

Tableau-A VIII-2 Valeurs d'IA en fonction du nombre de critères

Nombre de critères	2	3	4	5	6	7	8	9	10	11
IA	0	0.58	0.9	1.12	1.24	1.32	1.41	1.45	1.49	1.51

ANNEXE IX

MATRICES DE PONDÉRATION

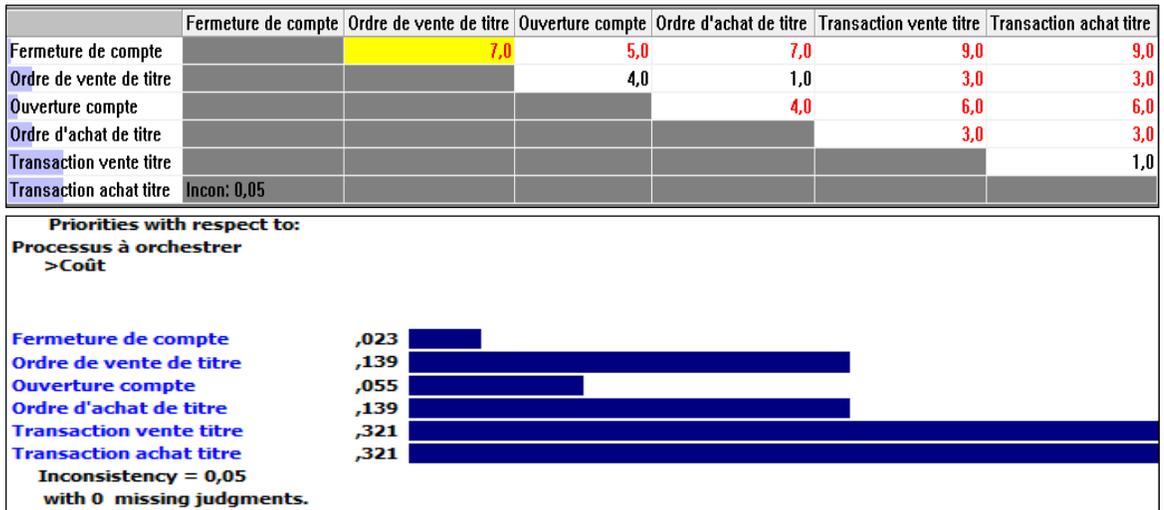


Figure-A IX-1 Matrice de pondération des alternatives par rapport au critère « Coût ».

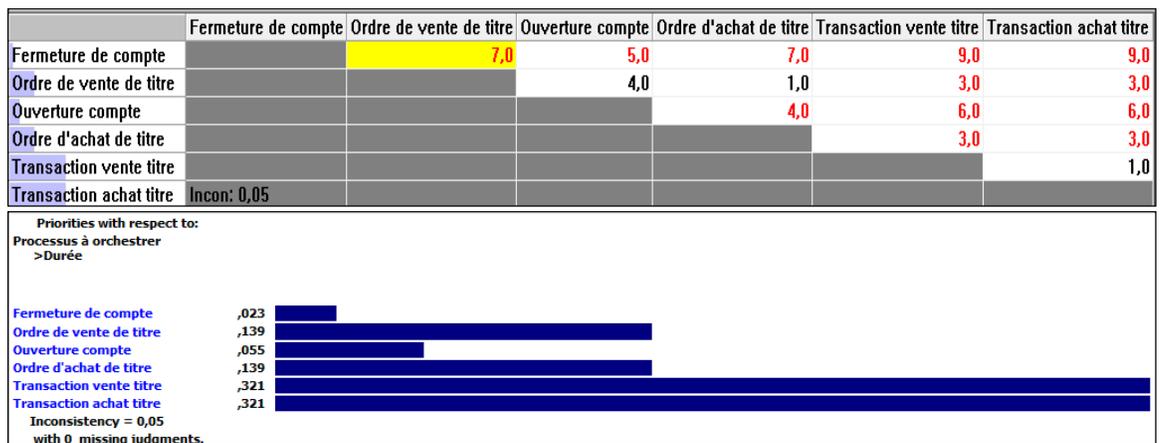


Figure-A IX-2 Matrice de pondération des alternatives par rapport au critère « Durée ».

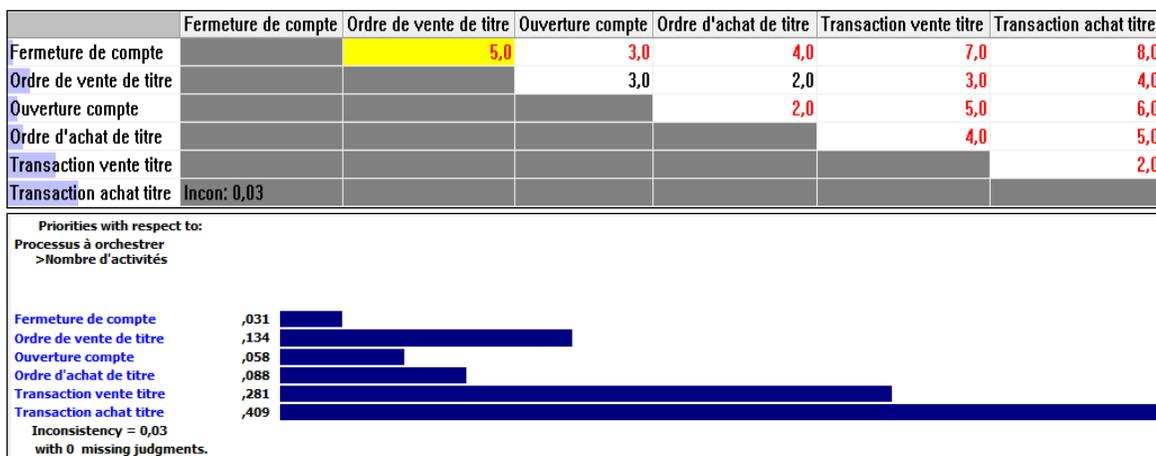


Figure-A IX-3 Matrice de pondération des alternatives par rapport au critère « Nombre d'activités ».

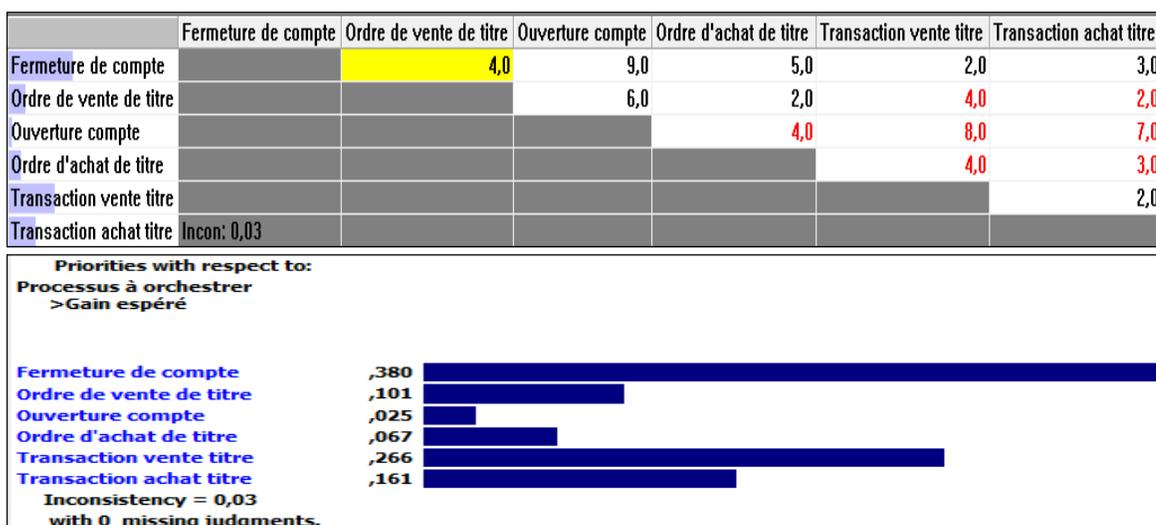


Figure-A IX-4 Matrice de pondération des alternatives par rapport au critère « Gain espéré ».

BIBLIOGRAPHIE

- Arsanjani, A., S. Ghosh, A. Allam, T. Abdollah, S. Ganapathy et K. Holley. 2008. « SOMA: a method for developing service-oriented solutions ». *IBM Systems Journal*, vol. 47, n° 3, p. 377-96.
- Azevedo, Leonardo Guerreiro, Flavia Santoro, Fernanda Baiao, Jairo Souza, Kate Revoredo, Vinicios Pereira et Isolda Herlain. 2009. « A method for service identification from business process models in a SOA approach ». *Lecture Notes in Business Information Processing*, vol. 29, p. 99-112.
- Boerner, R., et M. Goeken. 2009. « Service identification in SOA Governance literature review and implications for a new method ». In., p. 588-93. Coll. « 2009 3rd IEEE International Conference on Digital Ecosystems and Technologies (DEST) ». Piscataway, NJ, USA: IEEE. <<http://dx.doi.org/10.1109/DEST.2009.5276742>>.
- Bonnet, Pierre. 2005. « Cadre de Référence Architecture SOA (*) Meilleures Pratiques ». En ligne. Paris: Orchestra Networks. Consulté le 21 decembre 2009.
- Chun, Ouyang, M. Dumas, A. H. M. ter Hofstede et W. M. P. van der Aalst. 2008. « Pattern-based translation of BPMN process models to BPEL Web services ». *International Journal of Web Services Research*, vol. 5, n° 1, p. 42-62.
- Dan, Asit, Robert D. Johnson et Tony Carrato. 2008. « SOA service reuse by design ». In., p. 25-28. Coll. « Proceedings - International Conference on Software Engineering ». Leipzig, Germany: Inst. of Elec. and Elec. Eng. Computer Society. <<http://dx.doi.org/10.1145/1370916.1370923>>.
- DongSu, Kang, Song Chee-yang et Baik Doo-Kwon. 2008. « A method of service identification for product line ». In. Vol. vol.2, p. 1040-5. Coll. « 2008 Third International Conference on Convergence and Hybrid Information Technology (ICCIT) ». Los Alamitos, CA, USA: IEEE Computer Society. <<http://dx.doi.org/10.1109/ICCIT.2008.113>>.
- Dumas, Philippe, et Gilles Charbonnel. 1990. *La méthode OSSAD : Pour maîtriser les technologies de l'information*. Les éditions d'organisation, 161 p. <http://dumas.univ-tln.fr/Docs_PDF/OSSAD%20vol1%20020315.pdf>.
- Dwivedi, V., et N. Kulkarni. 2008. « A model driven service identification approach for process centric systems ». In., p. 65-72. Coll. « 2008 IEEE Congress on Services Part II (SERVICES-2) ». Piscataway, NJ, USA: IEEE. <<http://dx.doi.org/10.1109/SERVICES-2.2008.16>>.

- Erl, Thomas. 2005. *Service-Oriented Architecture: Concepts, Technology, and Design*. Prentice Hall PTR, 792 p.
- Erl, Thomas. 2008. *SOA : Principles of Service Design*. 573 p.
- Erradi, Abdelkarim, Naveen Kulkarni et Sriram Anand. 2006. « Service design principles: A case study in modeling services for the securities trading domain ». *Computer Systems Science and Engineering*, vol. 21, n° 4, p. 263-271.
- Girish, Juneja, Dournaee Blake, Natoli Joe et Steve Birkel. 2007. *Service Oriented Architecture Demystified*, 1st. Coll. « IT Best Practices Series ». Intel Press, 326 p.
- Harris, Torry. 2007. « SOA Test Methodology ». Torry Harris. http://www.thbs.com/pdfs/SOA_Test_Methodology.pdf.
- Heubès, Christophe. 16 août 2007. « Mise en œuvre d'une SOA : Les clés du succès ». In *Blog Xebia France : J2EE, Agilité et SOA*. <http://blog.xebia.fr/2007/08/16/Mise-en-oeuvre-dune-soa-les-cles-du-succes/>.
- Heubès, Christophe. 19 mars 2009. « SOA du composant au service ». In *Blog Xebia France : J2EE, Agilité et SOA*. <http://blog.xebia.fr/2009/03/19/soa-du-composant-au-service-le-couplage-lache/#more-1633>.
- Huysmans, Philip, Jan Verelst et Herwig Mannaert. 2007. « Towards systematic identification of services : A domain-specific approach ». In, WSEHS/-. Vol. ISDM, p. 267-277. Coll. « ICISOFT 2007 - 2nd International Conference on Software and Data Technologies, Proceedings ». Barcelona, Spain: INSTICC Press.
- Kapil, Pant, et Juric Matjaz. 2008. *Business Process Driven SOA using BPMN and BPEL: From Business Process Modeling to Orchestration and Service Oriented Architecture*. PACKT, 328 p.
- Kishore, Channabasavaiah, Holley Kerrie et Tuggle Edward. 2004. « Migration vers une architecture orienté services ». USA: IBM.
- Klose, Karsten , Ralf Knackstedt et Daniel Beverungen. 2007. « Identification Of Services - A Stakeholder-Based Approach To Soa Development And Its Application In The Area Of Production Planning* ». vol. 13.
- M. Gordon Hunter, et Felix B. Tan. 2007. *Strategic Use of Information Technology for Global Organizations*. IGI Publishing, 408 p.

- Matjaz, B. Juric, Mathew Benny et Sarang Poornachandra. 2006. *Business Process Execution Language for Web Services: An Architect and Developer's Guide to Orchestrating Web Services Using BPEL4WS*, Second Edition. Packt, 372 p.
- Papazoglou, M. P., et W. J. van den Heuvel. 2006. « Service-oriented design and development methodology ». *International Journal of Web Engineering and Technology*, vol. 2, n° 4, p. 412-42.
- Plouin, Guillaume, Bruno Penneç, Pascal Grojean et Cyril Rognon. 2007. « SOA Perception des entreprises françaises ». Paris: http://www.01net.com/clubs/resultat_etude_soa.pdf. Consulté le 20/Avril/2010.
- Ramudhin, Amar, Eric Chan et Abdelkader Mokadem. 2007. « A framework for the modelling, analysis and optimization of pathways in healthcare ». In. Vol. 1, p. 698-702. Coll. « Proceedings - ICSSSM'06: 2006 International Conference on Service Systems and Service Management ». Troyes, France: Inst. of Elec. and Elec. Eng. Computer Society. <http://dx.doi.org/10.1109/ICSSSM.2006.320547>.
- Reddy, V., A. Dubey, S. Lakshmanan, S. Sukumaran et R. Sisodia. 2009. « Evaluating legacy assets in the context of migration to SOA ». *Software Quality Journal*, vol. 17, n° 1, p. 51-63.
- Rouillard, José, Thomas Vantroys et Vincent Chevrin. 2007. *Architectures orientées services : une approche pragmatique des SOA* Coll. « Série génie logiciel ». Vuibert, 317 p.
- Saaty, T. L. 2008. « Decision making with the analytic hierarchy process ». *International Journal of Services Science*, vol. 1, n° 1, p. 83-98.
- Sandy, Carter. 2007. *The New Language of Business: SOA & Web 2.0*, 1st. United States IBM Press, 320 p.
- Senthil, Mani, V. S. Sinha, Sukaviriya Noi et T. Ramachandra. 2008. « Using user interface design to enhance service identification ». In., p. 78-87. Coll. « 2008 IEEE International Conference on Web Services (ICWS) ». Piscataway, NJ, USA: IEEE. <http://dx.doi.org/10.1109/ICWS.2008.86>.
- Sneed, H. M. 2006. « Integrating legacy software into a service oriented architecture ». In., p. 11 pp. Coll. « 10th European Conference on Software Maintenance and Reengineering ». Los Alamitos, CA, USA: IEEE Computer Society.
- Srikanth Inaganti, Gopala Krishna Behara. 2007. « Service Identification: BPM and SOA Handshake ». (Mars), p. 12.

- Strnadl, C. F. 2007. « Bridging architectural boundaries design and implementation of a semantic BPM and SOA governance tool ». In., p. 518-29. Coll. « Service-Oriented Computing - ICSOC 2007. Proceedings Fifth International Conference. (Lecture Notes in Computer Science vol. 4749) ». Berlin, Germany: Springer-Verlag.
- Suntae, Kim, Kim Minseong et Park Sooyong. 2008. « Service identification using goal and scenario in service oriented architecture ». In., p. 419-26. Coll. « 2008 15th Asia-Pacific Software Engineering Conference ». Piscataway, NJ, USA: IEEE. <<http://dx.doi.org/10.1109/APSEC.2008.31>>.
- Van Nuffel, D. 2007. « Towards a service-oriented methodology: business-driven guidelines for service identification ». In. Vol. pt.1, p. 294-303. Coll. « On the Move to Meaningful Internet Systems 2007: OTM 2007 Workshops. Proceedings OTM Confederated International Workshops and Posters AWeSOMe, CAMS, OTM Academy Doctoral Consortium, MONET, OnToContent, ORM, PerSys, PPN, RDDS, SSWS, and SWWS 2007. Part I ». Berlin, Germany: Springer-Verlag.
- Yousef, R., M. Odeh, D. Coward et A. Sharieh. 2009. « BPAOntoSOA: a generic framework to derive software service oriented models from business process architectures ». In., p. 50-5. Coll. « 2009 Second International Conference on the Applications of Digital Information and Web Technologies (ICADIWT) ». Piscataway, NJ, USA: IEEE. <<http://dx.doi.org/10.1109/ICADIWT.2009.5273939>>.