MA-004.252-1

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Saad Dahlab Blida

جامعة البليدة - ١ -
مكتبة الكلية
كلية العلوم

Faculté des sciences

Département d'informatique

Mémoire Présenté par :

Zelazel Naima Lamara Khadidja

En vue d'obtenir le diplôme de master

Domaine : Mathématique et informatique

Filière : Informatique
Spécialité : Informatique
Option : Ingénierie de logiciel

Sujet :

Une Architecture orientée service et d'aide a la décision dédiée au processus de recrutement

Soutenu le :

MA-004-252-1

Président : Mme Farah
Examinateur : Mme Ouhyani
Examinateur : Mme Kherfan
Promotrice : Mme. Boumahdi.F
Encadreur : M. Louassi.M

Promotion 2014 / 2015

جامعة البليدة 1
مكتبة
الكلية
كلية العلوم

Résumé

L'Architecture Orientée Service est un style d'architecture logicielle pour lequel les processus métier de l'entreprise sont des composants logiciels paramétrables, orchestrant des tâches avec les acteurs de l'entreprise et des appels à des composants de services pour s'exécuter.

Différents modèles et méthodes sont utilisés pour soutenir le processus de conception de l'architecture SOA (Service Oriented Architecture), mais après de nombreuses années de pratique, beaucoup de questions et de problèmes non résolus provoquent l'échec des projets de développement SOA. Une des raisons est que les changements rapides dans l'environnement de l'entreprise nécessitent l'introduction de l'aspect décisionnel qui devrait soutenir efficacement le processus de conception de l'architecture SOA.

Vu l'importance de la décision, des outils et des méthodes ont été proposés dont le but est l'aide à la décision. Or les entreprises se sont retrouvées face à des systèmes d'information empilés d'applications hétérogènes au fil du temps.

Dans le cadre de notre PFE, nous nous proposons d'inclure les aspects décisionnels dans l'architecture SOA et l'enrichir avec des informations spécifiquement d'aide à la décision.

Nous avons retenu le processus de recrutement pour matérialiser la migration des aspects d'aide à la décision dans une architecture SOA.

Mots clés : SOA, SoaML, Service, Orchestration, Aide à la décision, Service Web, processus de recrutement, Méthode AHP.

ملخص

البنية الخدمية هي نوع من الهندسة البرمجية التي تصبح من أجلها المنظومة الوظيفية للمؤسسة مكوناتٍ برمجية تعطي مهامًا مع ممثلي المؤسسة ومركبات المصالح لأجل التنفيذ.

مختلف النماذج مستعملة لإسناد مناهج إعداد الهندسة الموجهة، لكن بعد عدة سنوات من التطبيق، كثير من التساؤلات بقيت مطروحة بدون حلول مما أدى لانهيار مشاريعها ، وهذا راجع للتغيير السريع لمحيط المؤسسة، مما استدعى تدخل هيئة اتخاذ القرار التي تستطيع مساندة منظومة إعداد الهندسة الموجهة.

نظرا لأهمية القرار، اقتُرحت وسائل وطرق مساعدة لاتخاذ القرار، في حين وجدت المؤسسات نفسها موجهة لأنظمة معلومات تراكمت مع مرور الزمن.

في إطار مشروعنا لنهاية الدراسة نقترح إدخال هيئات اتخاذ القرار في البنية الخدمية وإثرائها بمعلومات للمساعدة على اتخاذ القرار.

في النهاية اخترنا عملية التوظيف لتجسيد جوانب مساعدة اتخاذ القرار في البنية الخدمية.

Abstract

The service-oriented architecture is a style of software architecture for which the company's business processes are configurable software components, orchestrating tasks with the actors of the company and calls to service components to run.

Different models and methods are used to support SOA design process (Service Oriented Architecture), but after many years of practice, many unresolved issues and problems cause the failure of SOA development projects. One of the many reasons is that the quick changes in the business environment require the introduction of the decision-making aspect that should support effectively SOA architecture design process

Given the importance of the decision, tools and methods have been proposed with the aim of decision support. However companies found themselves facing stacked information systems of heterogeneous applications over time.

In the context of our PFE, we propose to include the policy aspects in SOA and full with specifically information for decision support.

We have retained the recruitment process to materialize migration aid aspects to the decision in an SOA.

REMERCIEMENTS

*Nous tenons, avant tout, à remercier **ALLAH**, tout clément, tout puissant, de nous avoir donné la force de réaliser ce travail.*

Notre sincère reconnaissance et notre sincère gratitude à Madame BOUMAHDI Fatima, notre promotrice, pour nous avoir proposé le sujet, pour sa disponibilité, pour ses précieux conseils, sa confiance qu'elle nous a toujours témoigné et sa sollicitude dont elle nous a entouré, et ce tout au long de l'élaboration du présent travail.

*Nous exprimons notre profonde reconnaissance aux membres du **jury** pour avoir accepté de juger notre travail.*

Que tous nos professeurs qui ont contribué à notre formation trouvent ici notre plus profonde gratitude.

Dédicaces

Je dédie ce travail

A mes très chers parents qui ont sacrifié leur vie pour ma réussite, et pour leurs encouragements. Que dieu les garde éternellement heureux.

A toute la famille Zelazel et la famille Misseraoui

A tous mes amis

A ma grande mère

A la mémoire de ma tante Misseraoui Fatima Azahra que dieu l'accueil dans son vaste paradis

Zelazel Naima

Dédicaces

Je dédie du plus profond de mon cœur ce modeste travail,

À ma très chère mère, pour toi je dédie toute ma vie,

À mon adorable père, merci pour ton soutien à toute épreuve,

À mes chers frères,

À mes chers grands parents,

À tout les membres de ma grande famille,

À ma binôme et sa famille,

À tous mes amis,

À tous les collègues de promotion,

À tous ceux qui m'ont soutenu.

Lamara Khadija

Liste des abréviations

SOA	Service Oriented Architecture
CHU	unité hospitalier-universitaire
WSOA	web Service Oriented Architecture
XML	eXtensible Markup Language
W3C	World Wide Consortium
WSDL	Web Services Description Language
SOAP	Simple Object Access Protocol
UDDI	Universal Description, Discovery and Integration
REST	Representational State Transfer
URI	Uniform Ressource Identifier
MIME	Multipurpose Internet Mail Extensions
TI	technologies de l'information
SI	système d'information
ESB	*Enterprise Service Bus*
SOMA	Service-Oriented Modeling and Architecture
AHP	Processus de hiérarchie analytique
SOAML	Service Oriented Architecture Modeling Language
DMN	Decision Model and Notation

SOAF	Service Oriented Architecture Framework
MDA	Model Driven Architecture
CIM	Computational Independent Models
PIM	Platform Independent Models
OI	Approche Outside In
MO	Approche Middle Out
SOA +d	Service Oriented Architecture with a Decision
OMG	Object Management Group
SM	services Métier
NLD	Niveau Logique de Décision de DMN
CRUD	Create, Read, Update, Delete
SOS	Systèmes Open Source
BPEL	Business Process Execution Language
SQL	Structured Query language
PHP	Personale Home Page
MLM	machine logique de la couche
MLO	machines logiques de la couche organisation
AJAX	Asynchronous Javascript And XML
BPD	Business Process Diagram

جامعة البليدة 1
مكتبة الكلية
كلية العلوم

BPEL	**Business Process Execution Language**
BPM	**Business Process Management**
BPEL4People	**Business Process Execution Language for People**
BPMN	**Business Process Modeling Notation**
BPMS	**Business Process Management System**
HTML	**Hyper Text Markup Language**
HTTP	**HyperText Transfer Protocol**
IBM	**International Business Machines**
SGBD	**Système de Gestion de Bases de Données Relationnelles**
UML	**Unified Modeling Language**
JBPM	**Jboss Business Process Management**
PAGRH	**Plan annuaire de gestion des ressources humaine**
DRH	**Directeur de ressource humaine**
DDF	**Direction des finances**
URL	***Uniform Resource Locator***
CSS	**Cascading Style Sheet**
SOAD	**Service Oriented Analysis and Design**
AD	**Approche Descendante**

Table de matière

1. Introduction ..... **Erreur ! Signet non défini.**

2. Présentation du sujet ..... **Erreur ! Signet non défini.**

2.1. L'entreprise concernée ..... **Erreur ! Signet non défini.**

2.2. Problématique ..... **Erreur ! Signet non défini.**

2.3. Objectifs du l'étude ..... **Erreur ! Signet non défini.**

3. Organisation du mémoire ..... **Erreur ! Signet non défini.**

Partie 1 Concept de base ..... **Erreur ! Signet non défini.**

Chapitre I. Architecture orientée service SOA ..... **Erreur ! Signet non défini.**

1. Introduction ..... **Erreur ! Signet non défini.**

2. Définition de la SOA ..... **Erreur ! Signet non défini.**

3. Objectif de l'architecture SOA ..... **Erreur ! Signet non défini.**

4. Les couches d'abstraction de SOA ..... **Erreur ! Signet non défini.**

4.1. La couche d'affaire ..... **Erreur ! Signet non défini.**

4.2. La couche applicative ..... **Erreur ! Signet non défini.**

4.3. La couche des services ..... **Erreur ! Signet non défini.**

5. Le concept de service ..... **Erreur ! Signet non défini.**

5.1. Service métier ..... **Erreur ! Signet non défini.**

5.2. Le service entité ..... **Erreur ! Signet non défini.**

5.3. Le service utilitaire ..... **Erreur ! Signet non défini.**

6. Les caractéristiques des services ..... **Erreur ! Signet non défini.**

6.1. Couplage faible ..... **Erreur ! Signet non défini.**

6.2. Découverte, sélection et consommation ..... **Erreur ! Signet non défini.**

6.3. Réutilisation ..... **Erreur ! Signet non défini.**

6.4. Le contrat des services ..... **Erreur ! Signet non défini.**

6.5. Asynchrone/ sans état ..... **Erreur ! Signet non défini.**

6.6. Autonomie ..... **Erreur ! Signet non défini.**

7. Les services Web et l'architecture SOA ..... **Erreur ! Signet non défini.**

7.1. Définition des services Web ..... **Erreur ! Signet non défini.**

7.2. Les standards utilisés par les services Web ..... **Erreur ! Signet non défini.**

8. Un exemple de SOA : REST ..... **Erreur ! Signet non défini.**

9. La gouvernance SOA ..... **Erreur ! Signet non défini.**

Les dimensions multiples à considérer ..... **Erreur ! Signet non défini.**

10. L'Enterprise Service Bus (ESB) ..... **Erreur ! Signet non défini.**

11. Avantage et inconvénient d'une SOA ..... **Erreur ! Signet non défini.**

11.1. .....Avantage

Erreur ! Signet non défini.

11.2. .....Inconvénient

Erreur ! Signet non défini.

12. Conclusion ..... **Erreur ! Signet non défini.**

Chapitre II. Méthodologie de développement SOA ..... **Erreur ! Signet non défini.**

1. Introduction ..... **Erreur ! Signet non défini.**

2. Stratégie de la conception des architectures SOA .... **Erreur ! Signet non défini.**

2.1. Top Down ..... **Erreur ! Signet non défini.**

2.2. Bottom up ..... **Erreur ! Signet non défini.**

2.3. Outside In ..... **Erreur ! Signet non défini.**

جامعة البليدة
مكتبة
الكلية
كلية العلوم

2.4. Middle Out..... **Erreur ! Signet non défini.**

3. Les méthodologies de développement des SOA..... **Erreur ! Signet non défini.**

3.1. SOMA (Service-Oriented Modeling and Architecture) d'IBM..... **Erreur ! Signet non défini.**

3.2. La méthodologie SOAD ..... **Erreur ! Signet non défini.**

3.3. La méthodologie SOAF ..... **Erreur ! Signet non défini.**

3.4. La méthodologie SODM..... **Erreur ! Signet non défini.**

3.5. La méthodologie CBDI-SAE..... **Erreur ! Signet non défini.**

3.6. La méthodologie de Thomas Erl..... **Erreur ! Signet non défini.**

3.7. La méthodologie de S. Jones ..... **Erreur ! Signet non défini.**

4. Comparaison des méthodologies de développement des SOA**Erreur ! Signet non défini.**

5. Conclusion ..... **Erreur ! Signet non défini.**

Chapitre III. La méthode soa+d ..... **Erreur ! Signet non défini.**

1. Introduction..... **Erreur ! Signet non défini.**

2. Description détaillée de la méthode Soa+d..... **Erreur ! Signet non défini.**

2.1. Phase 1 : La phase d'analyse ..... **Erreur ! Signet non défini.**

2.2. Phase 2 : Identifier et catégoriser les services.... **Erreur ! Signet non défini.**

2.3. Phase 3 : Modélisation de Services ..... **Erreur ! Signet non défini.**

2.4. Phase 4 : La réalisation..... **Erreur ! Signet non défini.**

3. Conclusion ..... **Erreur ! Signet non défini.**

Chapitre IV. Processus de hiérarchie analytique ..... **Erreur ! Signet non défini.**

1. Introduction..... **Erreur ! Signet non défini.**

2. Définition ..... **Erreur ! Signet non défini.**

5. Conclusion 34
Chapitre III. La méthode soa+d 35
1. Introduction 36
2. Description détaillée de la méthode Soa+d 37
2.1. Phase 1 : La phase d'analyse 38
2.2. Phase 2 : Identifier et catégoriser les services 41
2.3. Phase 3 : Modélisation de Services 44
2.4. Phase 4 : La réalisation 46
3. Conclusion 49
Chapitre IV. Processus de hiérarchie analytique 50
1. Introduction 51
2. Définition 51
3. Hiérarchie de décision : 51
4. Points forts de la méthode AHP 52
5. Trois principes de la méthode AHP 53
6. Différentes étapes de la méthode AHP 53
6.1. Etape 1 : Décomposer le problème en une structure hiérarchique 53
6.2. Etape 2 : Effectuer les combinaisons binaires niveau par niveau 53
6.3. Etape 3 : Déterminer les priorités 53
6.4. Etape 4 : Synthétiser les priorités 54
7. Exemple d'application de la méthode AHP 54
8. Avantage de la méthode AHP : 59
9. Conclusion 59
Partie 2 la conception de l'architecture orientée service 60
Chapitre V. La démarche de développement 61
1. Introduction 62
2. Phase1 : analyse 62

مكتبة الكلية

2.1. analyse du système d'information 62
2.2. Analyses métier 63
2.3. Analyse de la décision 66
3. Phase 2 : Identifier et classer les services 72
4. Phase 3 : spécification des services 74
5. Phase 4 : réalisation 79
5.1. Les choix techniques 79
5.1.2. Serveur http 80
5.1.3. Moteur d'orchestration 80
5.1.4. Formulaire 81
5.1.5. Portail client (candidat) 81
5.1.6. Environnement de développement (WampServer) 82
6. Conclusion 82
Chapitre VI. Réalisation et démonstratio 83
1. Introduction 84
2. Réalisation 84
2.1. La couche métier 84
2.2. La couche réalisation 85
2.3. Couche présentation 88
3. Les interfaces 89
4. La console d'administration des processus : 94
5. Conclusion 94
Conclusion Générale 95
Bibliographie 99

مكتبة

Liste des figures

Figure I.1 Les concepts de SOA 7
Figure I.2 Les couches d'une SOA (COUTURIER, 2011) 8
Figure I.3 La classification des services SOA (LEMRABET, 2012) 10
Figure I.4 Modèle opérationnel de l'architecture orientée services 15
FigureI.5 La gouvernance SOA (Nassar, 2012) 20
Figure I.6 La virtualisation de service avec un ESB 20
Figure II.1 Stratégie d'intégration (Xebia IT Architect, 2007) 25
Figure.II.2 La méthodologie SOMA (Arsanjani, et al., 2008) 27
Figure.II.3 La méthodologie SOAD (Zemmermann, et al., 2004) 28
Figure II.4 La méthodologie SOAF (Arradi, et al., 2006) 29
Figure II.5 La méthode SOD-M (De Castro, 2011) 30
Figure III.1 L'utilisation des standards par SOA+d (Boumahdi, et al., 2015) 36
Figure III.2 La description des phases de SOA+davec la notation BPMN (Boumahdi, et al., 2015) 37
Figure III.3 Les choix technique pour implémenter une architecture SOA (Boumahdi, et al., 2015) 47
Figure IV.1 Hiérarchie de décision. (Ernest Forman, 2001) 52
Figure IV.2 Structure hiérarchique de la méthode AHP (ZOUGGARI, 2011). 54
Figure IV.3 Hiérarchie de la structure du problème de sélection d'un projet (PANNETON, 2010). 55
Figure IV.4 Matrice de comparaisons par paires des critères en regard de l'objectif (PANNETON, 2010). 56
Figure IV.5 Matrice de comparaisons considérant la vétusté (PANNETON, 2010)..... 56
Figure IV.6 Matrice de comparaisons considérant la sécurité (PANNETON, 2010). ... 57
Figure IV.7 Matrice de comparaisons considérant la capacité (PANNETON, 2010).... 57
Figure IV.8 Performance relative de chaque critère de la hiérarchie (PANNETON, 2010). 58
Figure IV.9 Agrégation des projets (PANNETON, 2010). 58
Figure IV.10 Performance relative des critères et des alternatives (PANNETON, 2010). 59
Figure V.1 Diagramme de cas d'utilisation pour le recrutement 62
Figure V.2 Processus de recrutement BPMN 65

Figure V.3 Trier les candidats DMN 67
Figure V.4 diagramme étape 1. 68
Figure V.5 Les services de la vision Métier 72
Figure V.6 Les services de la vision «Système Information» 73
Figure V.7 Les services de la vision «Décision» 73
Figure V.8 Architecture générale. 77
Figure V.9 Le contrat de service Transmettre PAGRH Réception 77
Figure V.10 Le contrat de service Inscription Candidat 78
Figure V.11 Le contrat de service Sélection Candidat 78
Figure V.12 Le contrat de service Installation Candidat 79
Figure V.13 Choix techniques 79
Figure V.14 une vue globale de la plateforme de destination 82
Figure VI.1 Service CRUD. 84
Figure VI.2 Exemple d'un fichier WSDL. 86
Figure VI.3 Les étapes de transposition des processus métiers. 87
Figure VI.4 Définitions des rôles 87
Figure VI.5 Partie du processus principale 87
Figure VI.6 Intégration des formulaires 88
Figure VI.7 Intégration des Services Web 88
Figure VI.8 Formulaire AJAX « Transmettre PAGRH » 89
Figure VI.9 La fenêtre Accueil 89
Figure VI.10 Formulaire d'inscription du candidat 90
Figure VI.11 services CRUD 91
Figure VI.12 Liste des candidats triée. 91
Figure VI.13 Intalio|Workflow «Authentification» 92
Figure VI.13 Intalio|Workflow «Liste des tâches» 92
Figure VI.14 Formulaire de transmettre un PAGRH 93
Figure VI.15 Consulter poste 93
Figure VI.16 Intalio console. 94

Liste des tableaux

Le tableau II.1 résume les principales méthodes de mise en place d'une SOA au sein de l'entreprise. Ces méthodes sont synthétisées à l'aide des cinq critères déjà présentés. . 33
Tableau II.2 La comparaison des méthodologies selon ces critères (Boumahdi, et al., 2015) 33
Tableau III.1 Règle de Mapping 1 (Boumahdi, et al., 2015) 41
Tableau III.2 Règle de Mapping 2 (Boumahdi, et al., 2015) 42
Tableau III.3 Règle de Mapping 3 (Boumahdi, et al., 2015) 42
Tableau III.4 Règle de Mapping 4 (Boumahdi, et al., 2015) 42
Tableau III.5 Règle de Mapping 5 (Boumahdi, et al., 2015) 43
Tableau III.6 Règle de Mapping 6 (Boumahdi, et al., 2015) 43
Tableau III.7 Règle de Mapping 7 (Boumahdi, et al., 2015) 43
Tableau III.8 Règle de Mapping 8 (Boumahdi, et al., 2015) 44
Tableau III.9 Règle de Mapping 9 (Boumahdi, et al., 2015) 44
Tableau IV.1 Liste de projets déposés (PANNETON, 2010) 54
Tableau V.1 Liste finale de candidats pour le poste de directeur. 68
Tableau V.2 Matrice des critères de décisions. 69
Tableau V.3 Matrice des critères de décisions en pourcentage relatifs 69
Tableau V.4 Matrice de comparaison "âge" 70
Tableau V.5 Matrice avec les pourcentages relatifs "âge" 70
Tableau V.6 Matrice de comparaison "expérience" 70
Tableau V.7 Matrice avec les pourcentages relatifs "expérience" 70
Tableau V.8 Matrice de comparaison "éducation" 71
Tableau V.9 Matrice avec les pourcentages relatifs "éducation" 71
Tableau V.10 Matrice de comparaison "appui" 71
Tableau V.11 Matrice avec les pourcentages relatifs "appui" 71
Tableau V.12 Matrice de solutions 71
Tableau V.13 solution finale 72
Tableau V.14 Tache à l'action UML (Brian Elvesæter, 2010). 74
Tableau V.15 Sous-Processus à l'architecture de services (Brian Elvesæter, 2010) 74
Tableau V.16 Pool au Participant (Community-level) (Brian Elvesæter, 2010). 75
Tableau V.17 Ligne (Lane) au participant (Brian Elvesæter, 2010). 75

Tableau V.18 Le début de Flux de message au service (Brian Elvesæter, 2010)..... 75
Tableau V.19 La fin de Flux de message à la demande (Brian Elvesæter, 2010)..... 76
Tableau V.20 fragment de processus (modèle) au contrat de service (Brian Elvesæter, 2010)..... 76

Introduction Générale

جامعة البليدة - 1
مكتبة
كلية العلوم

1. Introduction

Une architecture orienté service est une architecture logicielle s'appuyant sur un ensemble de services simples, qui sont développés en s'inspirant des processus métier de l'entreprise. Un service est un composant fonctionnant de manière autonome et offrant des fonctionnalités métiers à d'autres applications ou d'autres services.

À l'heure actuelle, l'Architecture Orientée Services est principalement destinée à intervenir au niveau de : système informatique et métier des entreprises, en présentant leurs différentes applications sous forme d'un ensemble de modules indépendants capables d'être composés appelés services. Néanmoins, en quête d'agilité, la SOA (Service Oriented Architecture) doit dépasser le cadre technique lié au niveau l'informatique et au niveau métier pour toucher le niveau décisionnel de l'entreprise. Dans ce sens, le vrai challenge consiste à étendre la SOA à l'aspect décisionnel de l'entreprise. Le but de ce sujet, est d'ouvrir la boite noire de la vision décisionnelle, et montrer ses composants selon une architecture SOA pour le processus « recrutement » au niveau du l'hôpital CHU de Blida Unité M'hamed Yazid .

2. Présentation du sujet

2.1. L'entreprise concernée

L'unité M'hamed Yazid est unité hospitalier-universitaire du CHU (centre hospitalier-universitaire) de Blida car le CHU de Blida est constitué de plusieurs unités.

Avant de faire partie du CHU de Blida l'unité M'hamed Yazid a été réalisée durant l'époque coloniale et s'appelant la chimique « Farroudja » et était spécialisée dans les accouchements.

Durant les années quatre-vingt cette chimique a été rattachée au CHU de Blida avec deux spécialités « La chirurgie générale, la traumatologie et chirurgie orthopédique » avec une capacité de 80 lits plus un seul poste d'opération, un seul poste de réanimation et deux unités des urgences.

Les deux services sont dirigés par deux professeurs chefs de services sans l'égide d'une direction qui comprend des services administratifs, des services techniques, et des services pour la gestion médicale et paramédicales.

2.2. Problématique

Dans ce sujet, nous avons choisis le processus métier « Le recrutement», ce dernier inclut des problèmes de décisions comme : le classement et la sélection des candidats.

La motivation principale de ce sujet est d'offrir l'aide pour résoudre les problèmes de décision dans le processus de recrutement, en se basant sur une architecture orientée service.

Le processus de recrutement rencontre quelque problème au sein du système d'information comme :

- La difficulté de la sélection du meilleur candidat.
- La recherche des candidats faite par une annonce de façon manuelle.
- Le recrutement des candidats prend beaucoup de temps.
- L'équipe de recrutement contacte les candidats sélectionnés pour la phase de contractualisation.
- Le classement des candidats.
- la sélection finale.

2.3. Objectifs du l'étude

Le processus de recrutement peut engager dans le monde de SOA avec les objectifs suivants :

- Automatiser le classement des candidats.
- Aider le décideur à prendre des bonnes décissions.
- Apporter une sécurité dans le classement des candidats.
- Informatiser la contractualisation du service de recrutement.
- Avoir un classement fiable.
- Connaitre à tout moment tous les informations concernant le candidat.
- Eviter les arrêts de recrutement et minimiser les durées du temps dues par le classement manuel.

3. Organisation du mémoire

Ce mémoire est composé d'une introduction générale suivie de deux parties, une partie théorique et une autre pratique.

La première partie sera consacrée à l'étude théorique des grands concepts rencontrés lors de la réalisation de notre projet, structurée en trois chapitres.

Le premier chapitre « Introduction au architectures orientées services » dans lequel nous introduisons la notion de processus métier, l'architecture orientée services et les services web. Le second chapitre présente des méthodes de développement orientées service. Dans le troisième chapitre nous allons présenter la méthode Soa+d (Service Oriented Architecture with a Decision), en définissant les différents aspects de cette méthode.

Enfin le quatrième chapitre présentera la « méthode AHP (Processus de hiérarchie analytique) » c'est la solution adaptée pour facilité l'aide a la décision dans le recrutement.

La deuxième partie « La conception de l'architecture orientée service» consacrés d'abord à la conception de la solution pour notre problématique avec la méthode Soa+d, tout en définissant les différents aspects de cette méthode, en utilisant les langages de modélisation SOAML (Service Oriented Architecture Modeling Language), BPMN (Business Process Modeling Notation) et DMN (Decision Model and Notation). Ensuite, nous passerons à l'implémentation de la solution retenue au niveau de la conception après une description de l'environnement du travail. Nous achèverons cette partie avec la présentation du système réalisé.

A la fin une conclusion générale résumera les principales leçons tirées de ce travail.

Partie 1 Concept de base

Chapitre I

Architecture orientée service AOS

1. Introduction

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 (AOS) qui viennent pour contribuer à l'adaptation des systèmes d'information au nouvel environnement assez turbulent dans lequel évoluent les organisations.

Ce premier chapitre commence par une définition de l'architecture orientée service, ensuite nous présentons l'objectif de SOA et ces couches d'abstraction.

Dans un deuxième temps, nous abordons le concept de service des SOA et ces caractéristiques. Ensuite nous donnons un exemple d'architecture orientée service.

Enfin, nous abordons la gouvernance SOA qui représente un processus intéressant pouvant être à la base de la gestion d'un environnement de services, et on termine par les avantages et les inconvénients de l'architecture orienté service.

2. Définition de la SOA

Une architecture orientée service est une architecture logicielle s'appuyant sur un ensemble de services simples. Ils sont développés en s'inspirant des processus métier de l'entreprise. Un service est un composant fonctionnant de manière autonome et offrant des fonctionnalités métiers à d'autres applications ou d'autres services.

Ces services représentent les fonctions basiques des fonctionnalités des entreprises. Ils dialoguent entre eux au travers de bus ou par Internet, on parle alors de Web Service (WSOA) (Ulrich Duvent, 2009/2010).

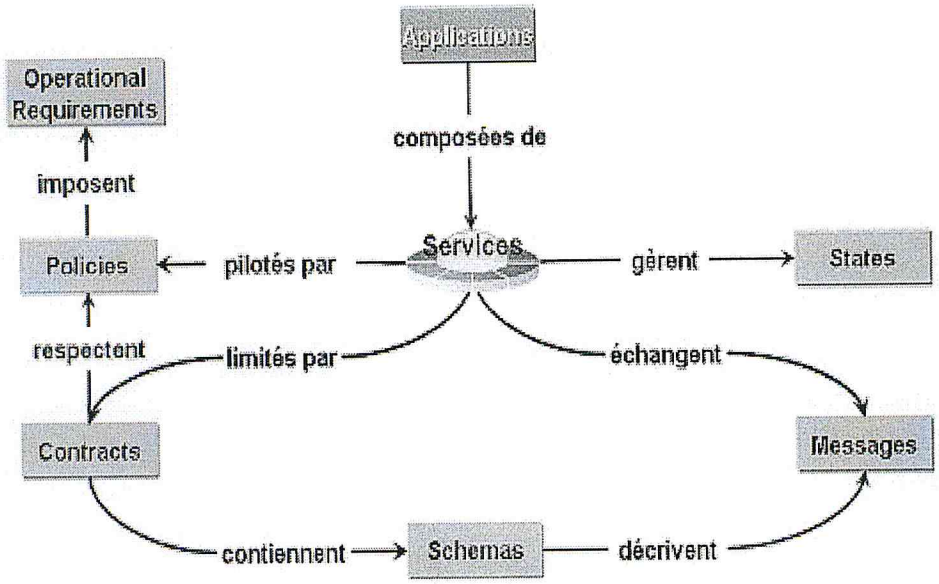


Figure I.1 Les concepts de SOA

Les services au centre de ce schéma sont les unités atomiques d'une architecture orientée services. Ces services communiquent entre eux par le biais de messages. Ces communications peuvent être synchrones ou asynchrones. Le langage de programmation du service n'a aucune importante, ainsi que la plateforme d'exécution, matérielle et logicielle (Ulrich Duvent, 2009/2010).

3. Objectif de l'architecture SOA

L'objectif des architectures orientées services ou SOA (Service Oriented Architecture) est de permettre aux entreprises de réaliser des progrès métiers et technologiques en combinant l'innovation des processus, une gouvernance efficace et une stratégie technologique centrée sur la définition et le réemploi des services.

Bien qu'impliquant une stratégie à long terme - qui amène à repenser la façon dont les technologies de l'information est fournie aux utilisateurs - les SOA doivent également répondre à des initiatives immédiates.

En d'autres termes, le bénéfice des SOA exige de maintenir l'équilibre entre objectifs à long terme et besoins opérationnels à cour terme en instituant, dès l'initialisation, un ensemble de pratiques d'excellence organisationnelles, financières, opérationnelles, de conception et de livraison.

Le modèle SOA en 6 domaines conçu par BEA « encapsule » ces pratiques d'excellence dans six domaines clés – qui doivent être traités avec la même diligence pour proposer un Framework SOA performant (BEA Systems,, 2005).

4. Les couches d'abstraction de SOA

La Figure I.2 illustre qu'une SOA est composée de plusieurs couches d'abstraction.

La première couche représente l'ensemble des systèmes patrimoniaux et des bases de données auxquelles on souhaite donner accès la seconde couche est l'abstraction des données.

Toutes les données doivent être ramenées dans un format standardisé, la troisième couche est la messagerie, cette couche assure le transport des données.

Les services composent la quatrième couche et permettent la consommation (l'utilisation) de ces données, une cinquième couche permet de regrouper plusieurs services entre eux par composition, orchestration ou chorégraphie pour réaliser une activité ou un processus d'affaires particulier (COUTURIER, 2011).

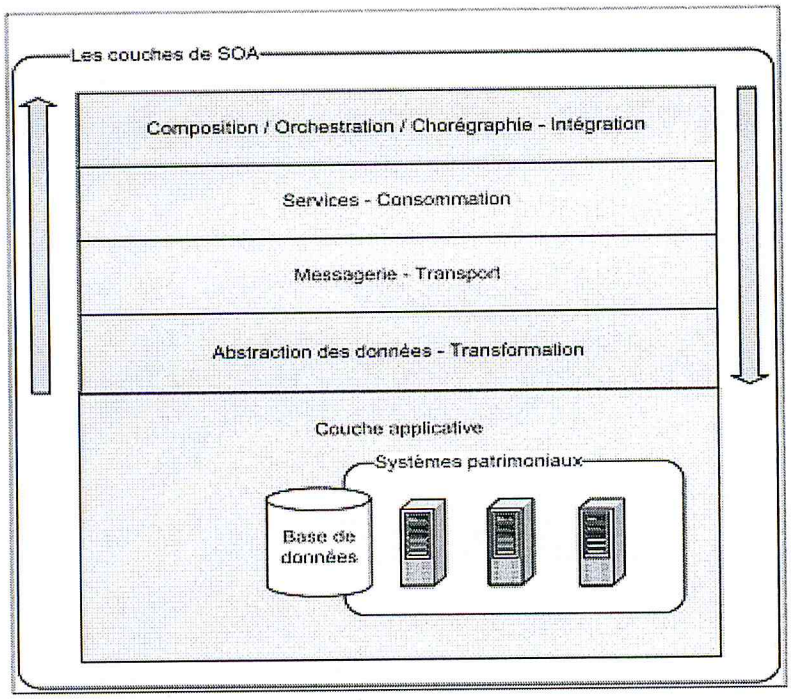


Figure I.2 Les couches d'une SOA (COUTURIER, 2011)

4.1. La couche d'affaire

La couche d'affaires représente les processus d'affaires et les activités qu'ils contiennent.

Ces processus et ces activités sont agnostiques de la technologie qui les supporte. Cette couche n'est pas représentée dans la Figure I.2 C'est la couche de plus haut niveau (COUTURIER, 2011).

4.2. La couche applicative

La couche applicative contient l'ensemble des applications et des bases de données qui supportent la mission de l'entreprise. Cette couche est située en dessous de la couche de « transformation » de la Figure I.2 C'est la couche de plus bas niveau (COUTURIER, 2011).

4.3. La couche des services

La couche des services se retrouve entre la couche d'affaires et la couche applicative.

Elle correspond aux couches de « messagerie » et de « service » de la Figure I.2

La couche des services peut elle-même être subdivisée en plusieurs sous-couches. Il y a tout d'abord la couche des services applicatifs. Les services applicatifs permettent d'interagir avec la couche applicative. Les services applicatifs seront utilisés par d'autres services de plus haut niveau. Ce sont les seuls services qui seront associés à une technologie particulière, celle de l'application avec laquelle ils interagissent.

La couche des services d'affaires permet de réaliser les activités automatisables des processus d'affaires. Ces services sont, grâce aux services applicatifs, indépendants de la technologie.

On peut subdiviser la couche des services d'affaires en deux autres sous-couches. La couche des services d'affaires entités représente généralement les entités du domaine d'affaires. Les services de tâches regroupent des opérations faisant interagir des services d'affaires entités ou d'autres services de tâches.

Il pourra finalement y avoir une couche d'orchestration, optionnelle, qui aura pour mission de diriger le fonctionnement de plusieurs services d'affaires afin de réaliser un

flux de travail plus complexe. Elle correspond à la couche « intégration » de la Figure I.2 (COUTURIER, 2011).

5. Le concept de service

Le service est la brique de base de l'architecture orientée service. 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.

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 (ABDELLATIF, 2011).

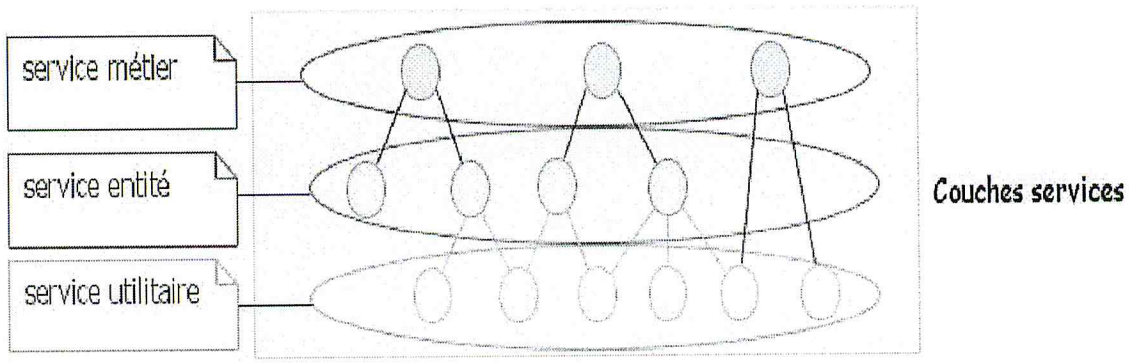


Figure I.3 La classification des services SOA (LEMRABET, 2012)

5.1. Service métier

(Marlon, et al., 2010) Considèrent que le terme service métier représente le résultat d'une opération dans une organisation. Du fait que les opérations d'une organisation peuvent être à différents niveaux de granularité, le service métier peut aussi être représenté à différents niveaux. D'autres auteurs considèrent qu'un service métier peut être compris comme

l'agrégation de services techniques, ou découvert directement au niveau de la modélisation des processus.

Le traitement offert par un service métier est un service de type particulier. En effet, le rôle du service métier est d'offrir un ensemble cohérent de traitements métiers. Il correspond à un périmètre fonctionnel exposé à des consommateurs indépendamment des choix d'architecture applicative. On estime que le service métier doit être défini du point de vue de l'utilisateur final, qui perçoit le résultat de SOA à travers le service métier qu'il consomme. Pour l'utilisateur, un service métier est une opération offrant une valeur ajoutée pour l'activité, favorisant son efficacité et améliorant sa performance (LEMRABET, 2012).

5.2. Le service entité

Les services entités sont découverts à partir des objets sémantiques principaux qui expriment les fondamentaux du métier. Ces services permettent de dégager un noyau stable composé des services hautement réutilisables. Ils représentent les domaines d'objets (objet métier) construits autour des objets centrales de l'organisation (LEMRABET, 2012).

5.3. Le service utilitaire

Ces services peuvent être utilisés pour enrober les ressources existantes de l'entreprise issues de systèmes patrimoniaux.

Contrairement aux deux autres types de services, les services utilitaires ne dépendent pas forcément du métier de l'organisation. Ils peuvent assurer des fonctionnalités utiles aux autres services, comme la journalisation, la sécurité, la gestion d'exception ou la transformation des données.

Ces services doivent être implémentés une seule fois ensuite ils doivent être considérés comme faisant partie de l'infrastructure SOA (LEMRABET, 2012).

6. Les caractéristiques des services

Afin de répondre au 'paradigme orienté services', un service doit présenter les caractéristiques détaillées dans les paragraphes suivants.

6.1. Couplage faible

Le couplage faible assure la flexibilité et réduit les dépendances entre les différents services.

Cette propriété permet d'avoir des services faiblement couplés pour simplifier la création et l'évolution des applications. Les aspects à considérer pour assurer le couplage faible sont les suivants :

- Un service ne peut pas appeler un autre service : Il délègue cette fonction à un traitement spécialisé dans l'enchaînement (fonction d'orchestration) .

Cette propriété signifie aussi que le service doit intégrer les fonctionnalités dont il est lui-même responsable.

- Une meilleure prise en compte de l'interopérabilité : Un service doit être indépendant des technologies d'implémentation. Il doit délivrer les fonctionnalités à travers son interface.

En effet, le service doit être accompagné d'une description qui explique aux consommateurs la manière de l'invoquer et d'utiliser ses fonctionnalités.

Il doit être activable indépendamment de sa technologie grâce à l'utilisation de standards. Par exemple, dans le monde des services web, l'envoi et de la réception des messages d'invocation se font en XML (Nassar, 2012).

6.2. Découverte, sélection et consommation

Les services devront être publiés de manière à garantir leur découverte, leur sélection et leur consommation sans l'intervention du fournisseur. Les consommateurs utilisent les mécanismes de découverte pour localiser les services d'une manière transparente et utilisent la description des services pour les sélectionner et les 'consommer'.

La description fonctionnelle et non fonctionnelle des services est à la base de leur découverte. Elle devra contenir des métadonnées concernant la localisation, les capacités et les exigences du service, et devra être réalisé avec précision à la conception des services.

La sélection peut se faire sur la base des propriétés fonctionnelles et non fonctionnelles.

À titre d'exemple, un consommateur peut rechercher la liste des services qui répondent au besoin fonctionnel et décider d'en sélectionner un particulier selon ses paramètres de qualité de service ou de sécurité (Nassar, 2012).

6.3. Réutilisation

Cette propriété devra être prise en compte dès les premières phases de conception des services.

En effet, les services doivent encapsuler une logique de traitement suffisamment générique pour être utilisés dans des contextes d'utilisation différents.

Cependant, le degré de réutilisation varie selon la portée du service et de sa granularité. Par exemple, un service de donnée a un potentiel fort de réutilisation.

En revanche, les services propres à des processus métier particuliers ont une réutilisation limitée au processus en question (Nassar, 2012).

6.4. Le contrat des services

Le contrat des services permet de décrire les services d'une façon normalisée. Il permet aux consommateurs de juger de l'utilisabilité du service pour le sélectionner.

De plus, il permet aux consommateurs de connaître les capacités du service, ses exigences et la manière dont il fonctionne.

Le contrat des services est définit comment un ensemble de documents contenant:

- La description de chaque opération du service.
- Les types de messages tels que les messages en entrée, les messages générés comme réponse et les messages d'exception ou d'erreur.
- La description de chaque type de données contenu dans les messages.
- La localisation physique du service et les protocoles de communication.
- Les informations et règles sur l'exploitation du service comme le temps de réponse, le temps pendant lequel le service doit être disponible, etc.

Dans le cas des services web, le contrat de service (intitulé 'interface de service') est décrit en deux parties. La partie abstraite déclare les messages en entrée et en réponse au traitement offert.

La partie concrète décrit les standards et protocoles techniques utilisés pour l'activation du service (Nassar, 2012).

6.5. Asynchrone/ sans état

Un service fonctionne de manière asynchrone. C'est-à-dire qu'il ne bloque pas le consommateur pendant qu'il s'exécute.

Ce principe est intéressant pour réduire les goulets d'étranglement (performance, robustesse).

En effet, les opérations d'un service s'exécutent sans maintenir un état pendant une longue période.

Une fois les opérations réalisées, le service ne retient aucune information sur l'état des messages ou du consommateur.

Toutefois, l'information du contexte (comme les données de session, les droits d'accès et les règles de validation) peut s'avérer très importante dans le cas de la composition de services.

C'est la raison pour laquelle la tâche de conservation et de passage de cette information est déléguée au service d'orchestration qui se charge de maintenir le bon déroulement des activités et de l'exécution de la composition (Nassar, 2012).

6.6. Autonomie

Lors d'un travail d'architecture applicative, les services sont identifiés en décomposant les traitements métier (décomposition des activités métier).

Ces services qui forment l'inventaire des services, devront être autonomes. L'autonomie hérite de la propriété 'couplage faible' et signifie que le service peut évoluer indépendamment des autres services de l'inventaire.

Nous distinguons l'autonomie dans la phase de la conception et celle de la phase d'exécution. Dans le premier cas, l'autonomie permet d'assurer des changements au niveau du service sans affecter les autres services de l'inventaire. Dans la phase de l'exécution, cette propriété signifie que le service peut contrôler le traitement métier qu'il expose et les ressources qu'il utilise d'une façon autonome (Nassar, 2012).

7. Les services Web et l'architecture 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 (ABDELLATIF, 2011).

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 reparties, sécurité, etc.). Ceci rend les Web services le choix par excellence pour la réalisation de SOA (ABDELLATIF, 2011).

La SOA représente la relation entre trois types de participants : le fournisseur de service, l'organisme de découverte de services et le demandeur du service (Papazoglou, et al., 2006). Les interactions comprennent les opérations « publish », « find » et « bind » (littéralement publier, trouver et lier) comme le montre la figure qui suit.

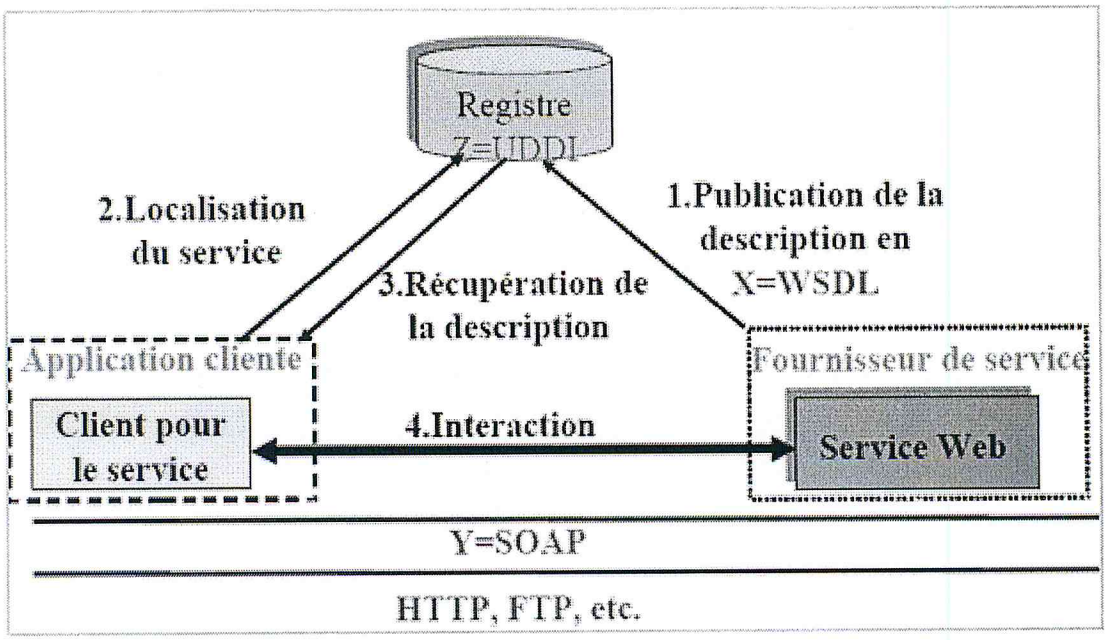


Figure I.4 Modèle opérationnel de l'architecture orientée services

Un client recherche auprès de l'annuaire les services qui répondent à ses besoins. Il obtient l'information de localisation du fournisseur et peut alors invoquer le service correspondant à son besoin. Si le client possède les informations nécessaires pour invoquer le service, il peut contacter directement le fournisseur.

7.1. Définition des services Web

Le point fort de la technologie des services web est qu'elle est indépendante des plateformes d'implémentation et des langages de programmation. Le groupe de travail « Web services Architecture » de la W3C (World Wide Consortium) a défini les services web comme: « Un service Web est un système destiné à soutenir l'interaction interopérable machine-à-machine sur un réseau. Il dispose d'une interface décrite dans une machine de format exploitable (spécifiquement WSDL). Autres systèmes interagissent avec le service Web dans la manière prescrite par sa description en utilisant SOAP-messages, généralement transportés en utilisant HTTP avec une sérialisation XML en conjonction avec d'autres standards Web » (Nassar, 2012).

Cette définition montre qu'un service Web peut être vu comme étant une application accessible par une interface spécifiée selon le standard Web Services Description Langage WSDL. De plus, l'interaction entre les services web se fait via le standard SOAP . Ainsi, les services web utilisent des standards pour décrire leur fonctionnement et la façon d'interagir avec eux (Nassar, 2012).

7.2. Les standards utilisés par les services Web

Les services web sont bâtis en se basant sur des standards ouverts et largement adoptés comme http et eXtensible Markup Language (XML).

Ces standards sont maintenus par des organisations indépendantes sans but lucratif. Il en résulte des standards ouverts et gratuits selon lesquels des applications et des outils compatibles avec les services web sont construits. (MAyyad, 2009).

Dans les paragraphes suivants, nous allons introduire les standards qui jouent un rôle vital dans l'environnement des services Web.

- **XML : eXtensible Markup Language**

XML concerne les spécifications liées à la customisation des types de documents, en utilisant des formats lisibles par l'Homme. Les spécifications de XML renforcent les règles de construction des documents, permet aux développeurs de créer leurs propres balises, permettant la définition, la transmission, la validation, et l'interprétation de données entre les applications et les organisations.

Les services web communiquent en utilisant XML (interfaces et messages de XML interprétables facilement par les applications) pour décrire leurs interfaces et encoder leurs messages et en s'appuyant sur les protocoles standards de web (MAyyad, 2009).

- **SOAP : Simple Object Access Protocol**

SOAP est un standard qui représente une « enveloppe » légère contenant la charge utile "payload" des messages échangés entre les producteurs et consommateurs de services. C'est un standard basé sur XML qui décrit le contenu du message échangé ainsi que la façon de traiter ce contenu. Ce protocole permet en outre une association avec les protocoles de transport. SOAP offre aussi un groupe de règles d'encodage pour exprimer les instances des types de données définies par les applications et une convention pour représenter les invocations à distance et les réponses des applications (MAyyad, 2009).

- **WSDL : Web Services Description Language**

WSDL est un langage au format de type XML utilisé pour la description des services offerts sur un réseau. Ce langage décrit les interfaces des messages contenants des information orientées documents ou orientées procédures. Les opérations et les messages sont décrits de manière abstraite. Ces descriptions sont ensuite couplées au système de transport (spécifiant le format des messages et les protocoles réseau) pour définir les points de connexion aux services afin de définir une extrémité. Les messages sont associés au protocole SOAP et au protocole de transport HTTP. La nature abstraite de WSDL (défini comme un outil de description pour les services) fournit la flexibilité nécessaire pour décrire des applications complexes des services web (MAyyad, 2009)

UDDI : Universal Description, Discovery and Integration

UDDI (UDDI, 2007) représente un groupe de protocoles et un répertoire public pour l'enregistrement et la recherche en temps réel des services web et autres processus métier. UDDI supporte un registre pour les services Web fournissant une liste des services disponibles sur un réseau. Un serveur UDDI permet aux organisations de : (MAyyad, 2009)

- Héberger plusieurs versions pour le même service.
- Créer des alias pour leurs services.
- Limiter l'accès à certains services.

8. Un exemple de SOA : REST

Pour montrer les performances des SOA, intéressons nous plus particulièrement à un exemple concret. REST est l'acronyme de "Representational State Transfer" inventé par Roy T. Fielding et dont la définition a été donnée dans sa thèse de doctorat sur les architectures Web en 2000. REST est un style architectural inspiré des SOA, mais basé sur le Web, il ne définit pas une spécification, mais explique un principe de développement à suivre. Il s'appuie sur l'utilisation des standards fondamentaux d'Internet. L'adressage des ressources est réalisé par les URI, Uniform Ressource Identifier, une chaine de caractères permettant d'identifier une ressource sur un réseau. Ceci permet donc d'identifier les différents services disponibles sur le réseau ainsi que des images, vidéos... Ensuite la communication avec le service est réalisée par protocole HTTP et la représentation des données est de type MIME (Multipurpose Internet Mail Extensions). Par exemple, si les données sont de type html, on aura « texte/html ». Il existe différents type : texte/css, texte/html, video/mpeg, image/png …

L'utilisation du protocole HTTP définit également l'interface du service grâce aux principaux verbes définit par celui-ci. Cette interface est standard et implémenté par tous les serveurs Web. Les services proposent des méthodes qui répondent aux requêtes http. On a « GET » qui permet de récupérer des données, « POST » qui envoie une commande, « PUT » pour mettre à jour une donnée et « DELETE » pour la suppression. Cependant, REST ne gère pas l'état du client (session), les données client doivent être stockées ailleurs, par exemple dans des bases de données, les cookies ...REST propose donc une architecture légère donnant de bonnes performances et tirant parti des fonctionnalités du protocole HTTP implémentées dans les serveurs (caching, etc.).

La conception de l'architecture d'une application en REST commence par l'identification des ressources disponibles. Ensuite, chaque ressource est définit par une URI permettant son accès de manière unique et sûre. Les scénarios de navigations sont définis grâce à des hyperliens auxquels on accède en « GET ». Et les clients récupèrent les informations des ressources consommées directement, il n'y a pas de sessions. Enfin, le filtrage du site est facilité par le biais des URI (Ulrich Duvent, 2009-2010).

9. La gouvernance SOA

La Gouvernance SOA est une discipline incontournable visant à accompagner et à sécuriser le déploiement d'une SOA de bout en bout, via l'identification, la spécification et la mise en œuvre de processus, de bonnes pratiques et de principes directeurs spécifiques au architectures de services (Nassar, 2012).

Les dimensions multiples à considérer

La gouvernance SOA recouvre un périmètre d'application très large, qui présente deux dimensions majeures : la stratégie et l'opérationnel.

- **Sur l'axe stratégique**, le programme de gouvernance SOA doit proposer un cadre pour la gestion de portefeuilles de solutions SOA (connues sous le nom d'applications composites).
- **Sur l'axe opérationnel**, il est nécessaire de contrôler et de sécuriser les trois temps forts de l'existence des solutions SOA et des services. Cela nécessite la mise en place de processus spécifiques de gestion des cycles de vie pour les solutions SOA et pour les services.

Sur chacun des deux axes, qu'il s'agisse des services ou des solutions SOA, le programme de Gouvernance SOA doit considérer les trois piliers fondamentaux d'un dispositif classique de gouvernance qui sont :

- L'organisation (acteurs de la gouvernance, acteurs des activités SOA et des tâches à réaliser, rôles et responsabilités…),
- Les processus (les processus de la gouvernance SOA, et les processus SOA à placer sous contrôle),
- La technologie (standards, outils, architecture, infrastructure…

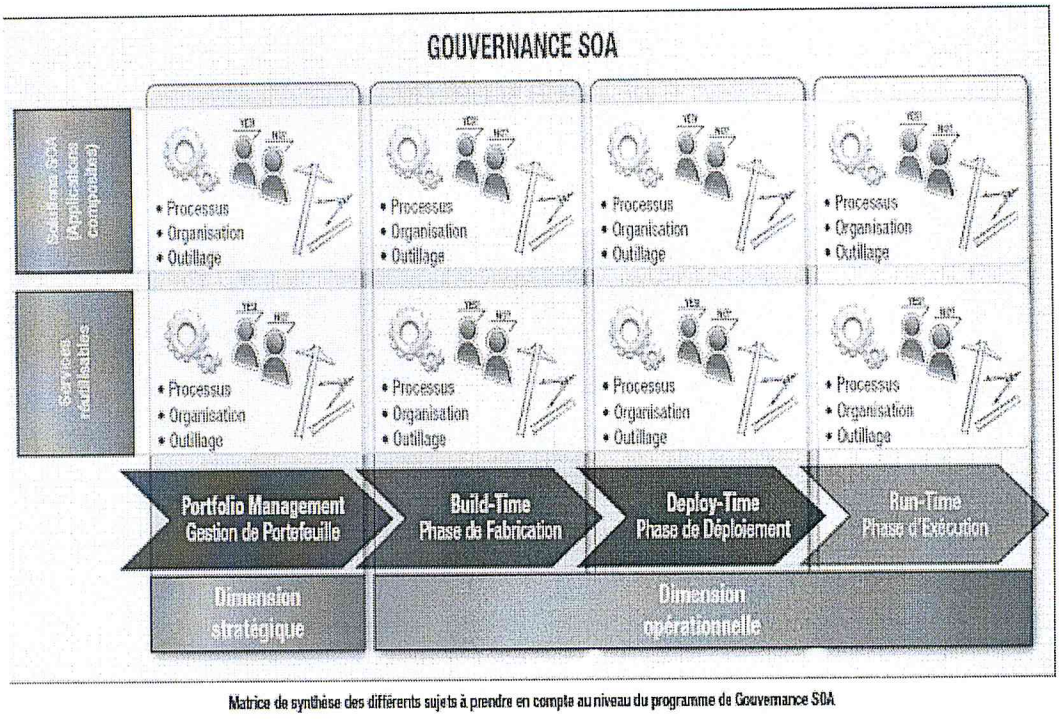


Figure I.5 La gouvernance SOA (Nassar, 2012)

10.L'Enterprise Service Bus (ESB)

Le concept de Bus de Service (Enterprise Service Bus - ESB) a été définit comme**:** « une nouvelle architecture qui exploite les web-services, le routage intelligent, la transformation et un middleware de messagerie ».

l'ESB doit aussi supporter plusieurs styles d'intégration : orientée service, orientée message et orientée événement. Dans le contexte d'une architecture orientée service, une définition restrictive considère l'ESB comme un garant du principe de couplage faible à travers la virtualisation des services : la transparence de la localité, du format de données et du protocole des services (LEMRABET, 2012).

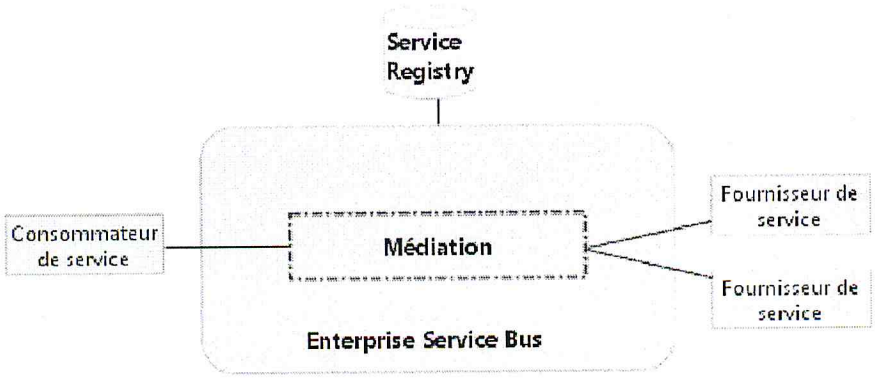


Figure I.6 La virtualisation de service avec un ESB

11. Avantage et inconvénient d'une SOA

11.1. Avantage

Les avantages de cette approche sont nombreux car elle permet de comptabiliser la contribution des technologies de l'information aux stratégies métiers et de quantifier le rapport coûts/avantages des fonctionnalités service par service.

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 quelques 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 (ABDELLATIF, 2011).

11.2. Inconvénient

La mise en place d'un SOA a un coût élevé à la fois financier et humain. Il faut former une équipe d'experts en conceptions ainsi que plusieurs équipes pour développer et administrer les différents services.

Dans le cas idéal, l'activité de l'entreprise doit être pensée autour des services. La conception du système d'information est donc une étape initiale critique. Si le fonctionnement de l'entreprise n'est pas organisé autour des services alors il est difficile d'utilisé un SOA et donc le coût de fonctionnement sera élevé. En effet, les SOA ont un intérêt limité si l'entreprise ne base pas ses processus sur l'exploitation des services, il faut donc concevoir des workflows adaptés.

De plus, il est difficile de migrer d'une architecture monolithique vers une architecture SOA sans étude préalable efficace (Ulrich Duvent, 2009-2010).

12.Conclusion

Une SOA couvre ainsi les modes de conception et d'organisation opérationnelle d'un système d'information pensé pour délivrer des services à tous les consommateurs qui en ont le besoin.

La notion d'annuaire devient centrale et incontournable. Elle permet aux consommateurs de trouver rapidement et efficacement leur bonheur dans un système qui va croître en richesse mais aussi en complexité avec le temps. SI le composant recherché n'existe pas, les consommateurs pourront en faire la demande auprès d'un Centre de Compétences dûment habilité, en suivant des procédures collaboratives de demande et d'approbation.

Au-delà de l'aspect purement technologique et des moyens utilisés, l'organisation collaborative d'une SOA et le dimensionnement d'un Centre de Compétences sont les clés de la réussite et de la pérennité de L'initiative dans l'entreprise. Mais, la question qui se pose: comment concevoir une architecture orientée service? La réponse de cette question est l'objectif du prochain chapitre.

Chapitre II Les méthodologies de développement SOA

1. Introduction

Il existe un certain nombre de méthodologies dédiées aux SOA. La plupart sont propriétaires et la description de leur contenu est souvent fragmentaire. Il existe également quelques méthodologies publiques.

Dans ce chapitre, nous nous intéressons aux méthodologies de conception des SOA pour trouver celles qui permettent de construire des SOA en assurant d'une part l'alignement métier et d'autre part l'aspect décisionnel. Nous présentons les différentes approches et méthodologies de conception qui font référence dans le domaine avant de définir des critères de comparaison pour établir un tableau comparatif de ces différentes méthodologies.

2. Stratégie de la conception des architectures SOA

Dans la l'littérature, les méthodes orientées service sont classifiées en :

2.1. Top Down

L'approche « top down » suit une logique descendante. Elle consiste à commencer l'intégration par la définition des processus métiers puits à la définition des services métiers nécessaire au fonctionnement des processus.

Cette logique est poursuivie dans le but d'aligner le système d'information sur le métier de l'entreprise (Erl, 2006). Cette approche permet de piloter l'architecture par les besoins métier tout en minimisant les redondances des services (Xebia IT Architect, 2007).

2.2. Bottom up

La seconde stratégie d'intégration, nommé « bottom up » opère dans le sens inverse de la précédente approche. Elle est réalisée dans les entreprises possédant déjà un système d'information. La conception se déroule de manière ascendante en commençant par identifier les fonctions existantes du SI que l'on peut réutiliser en tant que service (Erl, 2006).

En se basant sur un système déjà présent, l'entreprise peut récolter un gain sur les coûts d'implémentation. Ce type d'intégration est meilleur marché qu'une intégration top down (Xebia IT Architect, 2007).

مكتبة الكلية

2.3. Outside In

Cette approche préconise de mener en parallèle :

Un chantier Top Down pour définir les processus métiers et les services de plus haut niveau nécessaires à leur réalisation. Un chantier Bottom Up afin de cartographier l'existant applicatif dont dispose l'entreprise pour supporter les services métiers à forte valeur ajoutée.

De par sa nature, cette approche réunit les bénéfices des approches Top Down et Bottom Up. Elle permet de piloter la SOA par les besoins métiers tout en facilitant la réutilisation de services et la capitalisation sur l'existant (Xebia IT Architect, 2007).

2.4. Middle Out

Cette approche peut être introduite par la phrase « BPM et SOA sont les deux faces de la même pièce ». Assertion qui fait référence à un autre débat stérile ayant fait couler beaucoup d'encre et qui rappelle qu'il est impossible de dissocier la définition des processus métiers de l'établissement du catalogue de services. Par opposition à l'approche Outside In, cette méthode propose de commencer « In the middle », c'est-à-dire là où le métier et les IT parlent le même langage (ou en tout cas presque). Elle s'attaque donc d'emblée à ce qui reste un des principaux freins à l'adoption des SOA au sein des DSI françaises : La compréhension du métier de l'entreprise par les maîtrises d'œuvre et inversement, la compréhension des contraintes IT par les maîtrises d'ouvrage (Xebia IT Architect, 2007).

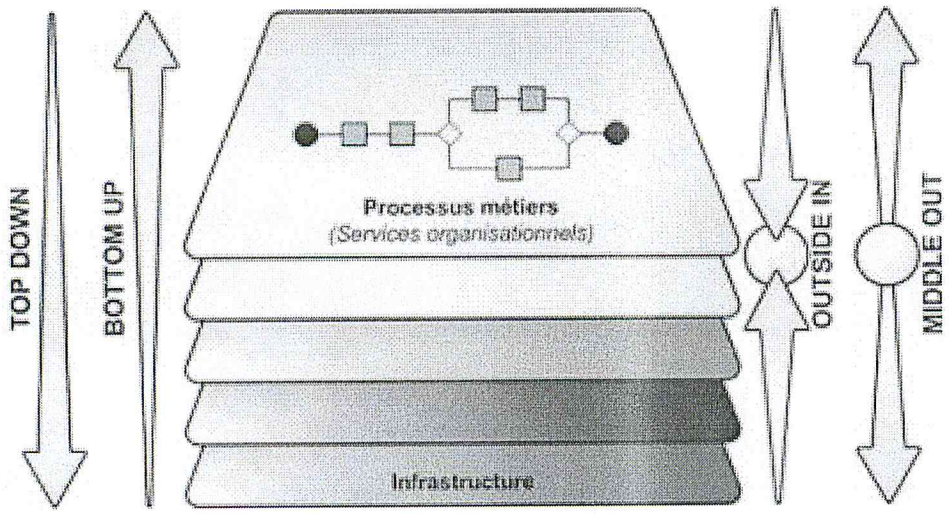


Figure II.1 Stratégie d'intégration (Xebia IT Architect, 2007)

3. Les méthodologies de développement des SOA

3.1. SOMA (Service-Oriented Modeling and Architecture) d'IBM

La méthode SOMA d'IBM (Arsanjani, et al., 2008) est une méthode pour l'identification, la modélisation et la spécification des services. Elle consiste en trois phases :

3.1.1. La phase d'identification des services

se base sur trois démarches : une démarche descendante (Top-Down) pour la décomposition des domaines. Elle offre une cartographie des cas d'utilisation du domaine concerné.

Une démarche ascendante (Bottom-Up) pour l'analyse du patrimoine existant. Alors que la démarche Bottom-Up, elle analyse les systèmes existants afin d'identifier les services de bas niveau.

Une démarche hybride (Middle-out) pour une modélisation du service selon le but. La démarche hybride (Middle-out) consiste en une modélisation de service basé sur les buts et vise à déterminer les services qui n'ont pas été identifiés dans les deux autres démarches.

3.1.2. La phase de spécification

commence par filtrer l'ensemble des services candidats en se basant sur des règles bien déterminées à savoir : l'alignement entre fonction et métier, l'élimination des redondances, la réutilisation des services par les différents processus métiers et la facilité de l'implémentation. Dans cette phase, les auteurs proposent d'attribuer une note pour les services sur une échelle de 1 à 5.

Les services qui auront les scores les plus élevés seront considérés comme des candidats pour la réalisation.

3.1.3. La phase de réalisation

Cette phase se charge d'implémenter l'ensemble des services. Le schéma si dessous illustre Toutes les étapes de SOMA .un livrable clé SOMA, qui couvre la syntaxe et les sémantiques de l'invocation de service, ainsi que d'autres caractéristiques opérationnelles tel que la possession de service, les dépendances et le visionnement (Zemmermann, et al., 2004).

La réalisation des services et des composants est assistée par des outils et des techniques telles que les patterns (par exemple les patterns d'IBM pour le e-business peuvent être utilisés).

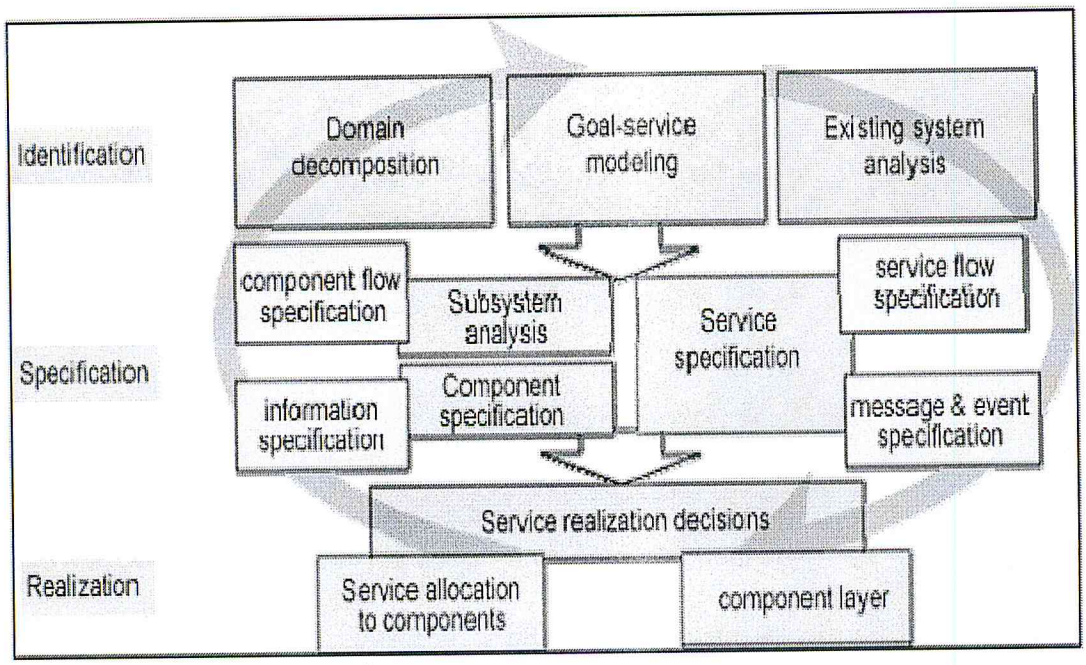


Figure II.2 La méthodologie SOMA (Arsanjani, et al., 2008)

3.2. La méthodologie SOAD

Dans la figure Figure.II.3, on voit comment SOAD (Zemmermann, et al., 2004) décompose un problème en plusieurs couches, la couche supérieure faisant l'abstraction de la couche inférieure. On fait clairement la distinction entre la couche d'affaires qui représente les processus et les services d'affaires, la couche de service qui offre le service logiciel, et finalement la couche composante qui implémente la logique métier dans les systèmes patrimoniaux.

Les auteurs recommandent d'intégrer les activités d'architecture d'entreprise avec les autres activités de modélisation incluse dans le cadriciel TOGAF. Par contre, aucune activité d'architecture d'entreprise n'est arrimée directement avec SOAD.

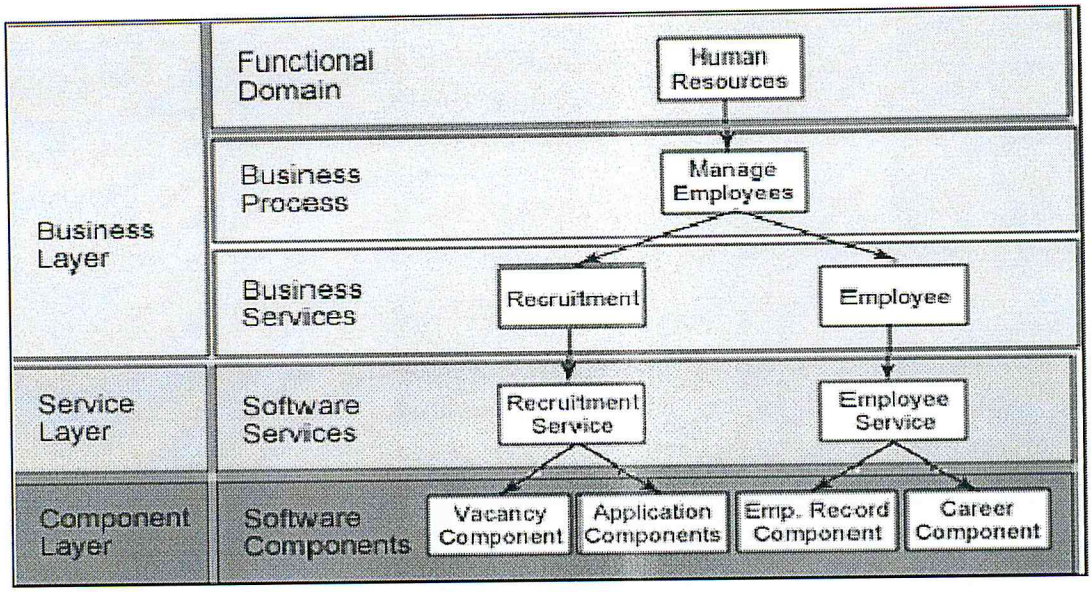


Figure.II.3 La méthodologie SOAD (Zemmermann, et al., 2004)

3.3. La méthodologie SOAF

La méthodologie SOAF (Service Oriented Architecture Framework) est une méthodologie réalisée par (Arradi, et al., 2006). Contrairement aux méthodes propriétaires précédentes, cette méthodologie détaille les artéfacts entrants, les activités et les artéfacts sortants pour chacune des phases. C'est à première vue la méthodologie disponible la plus complète. SOAF se décompose en cinq phases:

3.3.1. Extraction de l'information

Cette phase se concentre sur l'étude et la documentation de l'architecture d'affaires. On modélise les processus d'affaires, on fait l'inventaire des applications et on associe les processus d'affaires avec leurs applications respectives.

3.3.2. Identification des services

Cette phase décompose les processus d'affaires qui ont été préalablement modélisés afin d'identifier des services. Dans ce cas, on crée différents types de services; les services d'affaires sont associés à des tâches d'affaires alors que les services applicatifs communiquent avec les applications qui supportent les processus.

3.3.3. Définition des services

Cette phase modélise les services identifiés précédemment. On spécifie les interfaces des services, ainsi que les standards, qui seront utilisés. Les règles et politiques que devront respecter les futurs échanges de messages sont établies.

3.3.4. Réalisation des services

Cette phase définit la stratégie pour mettre en place les services identifiés à la phase d'identification. On identifie les transformations qui devront être faites sur les données ou sur les applications qui supportent le processus d'affaires. On développe les services selon la stratégie choisie.

3.3.5. Feuille de route et planification

Cette phase permet d'identifier les risques et la stratégie pour mettre en place les services qui ont été développés sans nuire au déroulement des activités de l'organisation. C'est également dans cette phase qu'est établie la gouvernance et que sont identifiés les propriétaires des services. Cette phase embrasse une vision plus large en analysant les risques et les plans de contingences de plusieurs projets.

La Figure II.4 SOAF se réfère à des artéfacts liés à des activités d'architecture d'entreprise (applications existantes, portefeuille, modèles et bonnes pratiques) sans pour autant donner davantage de détails sur la réalisation de ces activités. Cette méthodologie n'est enchâssée ni dans une méthodologie de développement complète ni dans un cycle incluant des disciplines d'entreprises.

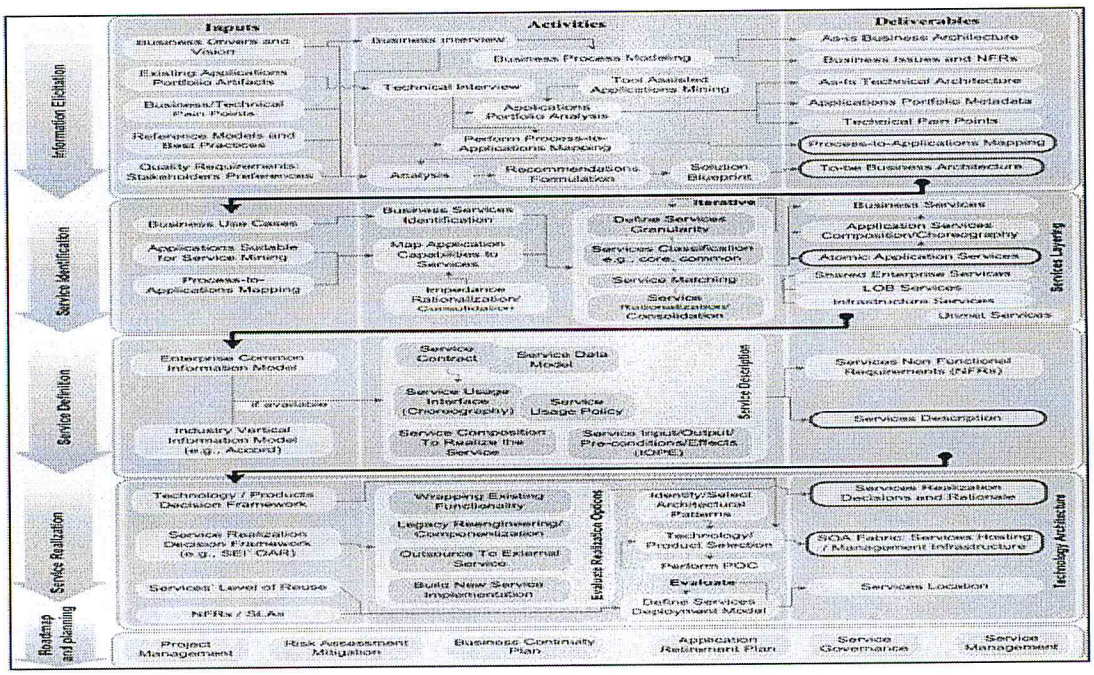


Figure II.4 La méthodologie SOAF (Arradi, et al., 2006)

3.4. La méthodologie SODM

(De Castro, 2011) propose une approche orientée service et le développement d'un modèle de base pour les systèmes web intitulée SOD-M, cette approche présente des modèles et des standards qui se focalisent principalement sur le développement des caractéristiques comportementales des systèmes, et l'établissement des standards pour la construction des modèles métier à un haut niveau d'abstraction.

L'approche SOD-M décrit trois méta-modèles organisés selon l'architecture MDA (Model Driven Architecture) (De Castro, 2011) et les niveaux d'architecture

- Les modèles au niveau CIM (Computational Independent Models) sont : le modèle valeur et le modèle BPMN.
- Les modèles niveau PIM (Platform Independent Models) sont : cas d'utilisation, cas d'utilisation étendu, le service processus et le service composition.

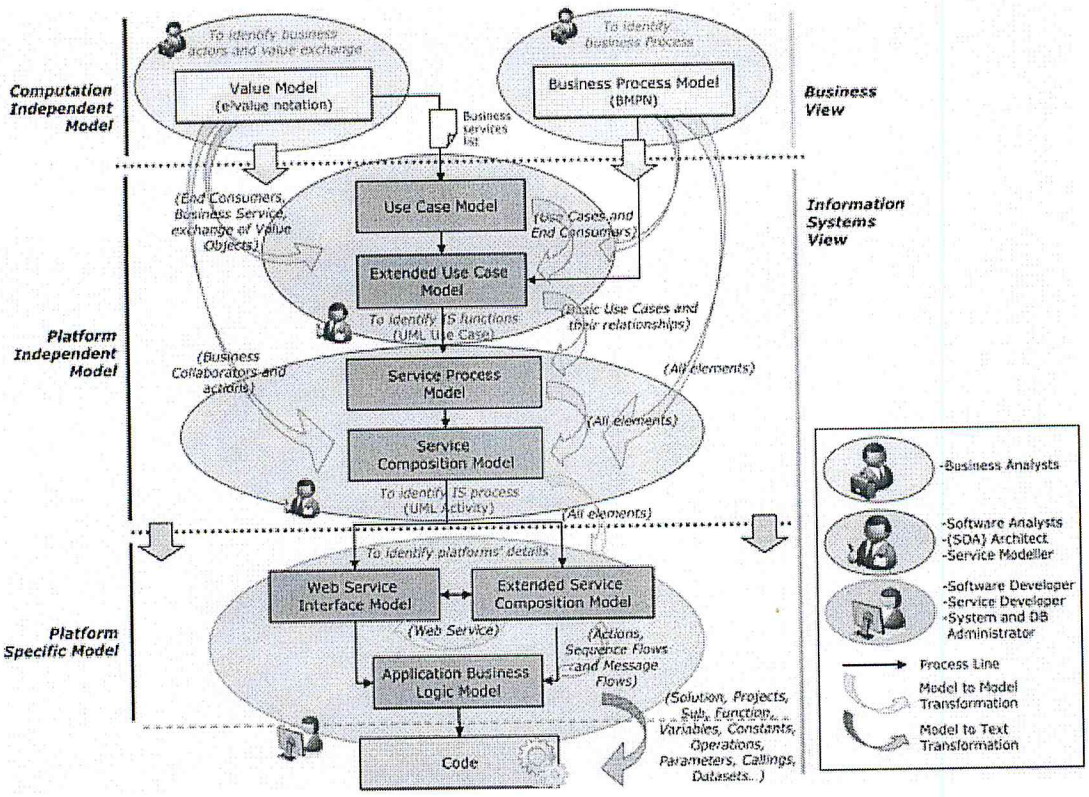


Figure II.5 La méthode SOD-M (De Castro, 2011)

3.5. La méthodologie CBDI-SAE

A partir de son cadre de référence (SOA Reference Framework), CBDI-SAE a développé une méthodologie SOA de conception et un modèle de service (Allen, 2007). La méthodologie vise à optimiser l'alignement métier en suivant une approche descendante. Le cycle de vie de services est couvert dans sa totalité, y compris les activités de déploiement, de surveillance et de gouvernance. Toutefois, l'accès à cette méthode est restreint : si la version 2 est en diffusion publique, la version la plus récente (version 3) est encore à accès réservé (CBDI-SAE, 2007).

3.6. La méthodologie de Thomas Erl

La méthodologie SOA de Thomas (Erl, et al., 2005) propose six phases dans le cycle de vie d'un développement SOA, Selon l'auteur, un projet SOA diffère des méthodes de développement standard uniquement dans les phases d'analyse et de conception alors que les phases subséquentes sont très similaires, Les six phases de ce cycle de vie sont :

3.6.1. Analyse orientée service

Ensemble des activités d'analyse permettant d'identifier les processus qui peuvent être représentés comme des services. On identifie les services candidats.

3.6.2. Conception orientée services

Ensemble des activités regroupant les services identifiés précédemment en un ensemble cohérent. Association avec les processus d'affaires existants. Séparation par couches selon que le service agisse au niveau de la coordination globale des services, du processus d'affaires ou de l'infrastructure.

3.6.3. Développement du service

Reprend globalement les activités de développement des méthodologies classiques. Utilisation d'un cadriciel de développement quelconque afin d'implémenter le service.

3.6.4. Test du service

Reprend globalement les activités de développement des méthodologies classiques.

Utilisation des techniques de test afin de vérifier et valider le fonctionnement des services. Il faut noter que la nature distribuée des services complique les activités de test.

3.6.5. Déploiement du service

Reprend globalement les activités de transition des méthodologies classiques. Tes d'intégration et intégration du service dans l'environnement de production.

3.6.6. Administration du service

Reprend globalement les activités de maintenance des méthodologies classiques. Gestion de l'évolution du service, gestion des changements et des demandes de modification.

3.7. La méthodologie de S. Jones

Dans (Jones, et al., 2005),S. Jones présente une méthodologie de conception qui met en évidence les propriétés métier d'une architecture orientée services. Il souligne l'importance des concepts métier et se focalise sur les éléments {Qui, Quoi, Comment et Pourquoi} dans l'identification des services métier selon une approche descendante. Toutefois, l'auteur n'aborde pas les aspects technologiques des architectures orientées services.

4. Comparaison des méthodologies de développement des SOA

La comparaison des méthodologies de développement des SOA elle se fait selon cinq critères (Ramollari, et al., 2007) :

- **Démarche de développement** : Il existe trois démarches de développement d'une approche SOA. La première est l'approche descendante (D) dans laquelle la logique métier des processus existants est examinée a fin d'identifier les services. À l'inverse, l'approche ascendante (A) débute par l'analyse du patrimoine applicatif a fin de déterminer les fonctions existantes du système d'information et tracer ainsi une cartographie applicative. À partir de cette cartographie, il est possible d'identifier les fonctions qui sont éligibles au rang de service. Enfin, l'approche hybride (H) préconise de mener en parallèle une approche descendante et une approche ascendante.
- **Documentation** : les approches se différencient dans leur documentation. Ce critère est utilisé pour analyser quelles sont les approches qui ont déjà été appliquées dans la pratique et quelles sont celles qui ont été purement dérivées des bases théoriques.
- **Langages et notations de modélisation** : ce critère permet de renseigner si la méthode propose de réutiliser des langages et les notations déjà existantes.

Comme exemples de ces langages et les notations, nous pouvons noter la conception orientée objet (UML :), le développement basé sur le langage SoaML, les modèles de processus (BPMN : Business Process Modelling Notation) et le nouveau standard DMN (decision modeling and notation).

- **Prise en compte de la décision**: la décision est une caractéristique importante et largement recherchée par les entreprises. Dans cette perspective, la décision se manifeste à travers la prise en compte de la solution (i.e., usage des services) au moment de leur modélisation. En outre, ce critère permet d'indiquer si les méthodes proposées considèrent les spécificités et les solutions des problèmes de décision. Nous avons choisi d'examiner les méthodes proposées par rapport à cette caractéristique.

Le tableau figure II.1 résume les principales méthodes de mise en place d'une SOA au sein de l'entreprise. Ces méthodes sont synthétisées à l'aide des cinq critères déjà présentés.

	Démarche	Les Standards utilisés	Documentation	Proprietary	Decision
SOA Decision Modeling (SOAD) 2009 (Zimmermann, 2009)	H	Non	Exemple	Oui	Oui
Service-oriented modeling and architecture (SOMA) 2008 (Arsanjani, et al., 2008).	H	Non	Exemple	Oui	Non
A Platform Independent Model for SOA (PIM4SOA) (2006) (Benguria & Larrucea, 2006)	D	Non	Exemple	Non	Non
CBDI-SAE Meta Model (2011) (Everware-CBDI, 2011)	D	SoaML + UML profile)	cas d'étude	Oui	Non
Modelling of Service Oriented Architectures With UML (2008) (Lopez-Sanz, et al., 2008)	D	UML profile	cas d'étude	Non	Non
A Service Oriented Information System: A Model Driven Approach (SOIS) (2012) (Fazziki, et al., 2012)	A	SoaML	Cas d'étude	Non	Non
SOD-M (the Service-Oriented Development Method) (2011) (De Castro, et al., 2011)	D	Non	Cas d'étude	Non	Non
Dynamic Homecare Service Provisioning Architecture (2012) (Zarghami, et al., 2012)	D	Non	Cas d'étude	Non	Oui
SOA^{ad}: the Service-Oriented architecture with a decisional aspect (Boumahdi, et al., 2014a)	D	SoaML DMN UML	Cas d'étude	Non	Oui

Tableau II.1 La comparaison des méthodologies selon ces critères (Boumahdi, et al., 2015)

5. Conclusion

La modélisation de service et ses approches de conception sont considérées comme un domaine de recherche actif, qui vise à définir des systèmes à un niveau d'abstraction élevé, indépendamment des technologies de mise en œuvre.

Dans ce chapitre, nous avons intéressé aux méthodologies de conception des SOA pour trouver celles qui permettent de construire des SOA en assurant d'une part l'alignement métier et d'autre part la décision des services développés, pour cela nous avons présenté les différentes approches et méthodologies de conception qui font référence dans le domaine.

D'après les résultats obtenus dans ce chapitre, aucune méthode de développement SOA prend en considération les problèmes de décisions dans un processus métier.

Pour cette raison, nous allons présenter dans le prochain chapitre, la nouvelle méthode SOA+d, cette dernière permet de résoudre les problèmes de décision.

Chapitre III La méthode soa+d

1. Introduction

Comme nous l'avons souligné dans la première partie, les architectures orientées services contient différentes approches de modélisation et d'identification de services, mais la majorité de ces approches ne sont pas flexibles du fait qu'elles ne répondent pas aux besoins décisionnels.

Dans ce chapitre nous allons présenter la nouvelle méthodologique qui est dénommée SOA +d (Service Oriented Architecture with a Decision).

SOA +d s'adresse aux petites et moyennes entreprises pour lesquelles, il est indispensable de proposer une solution basée sur des standards. L'utilisation des standards constitue une boîte à outils à laquelle s'ajoute une démarche guidant la construction de l'architecture de services à chaque étape. En effet les standards UML (Unified Modeling Language), BPMN (Business Process Modeling Notation), DMN (Decision Model and Notation) et SoaML (Service Oriented Architecture Modeling Language) sont utilisés dans le cadre de la méthode SOA +d.

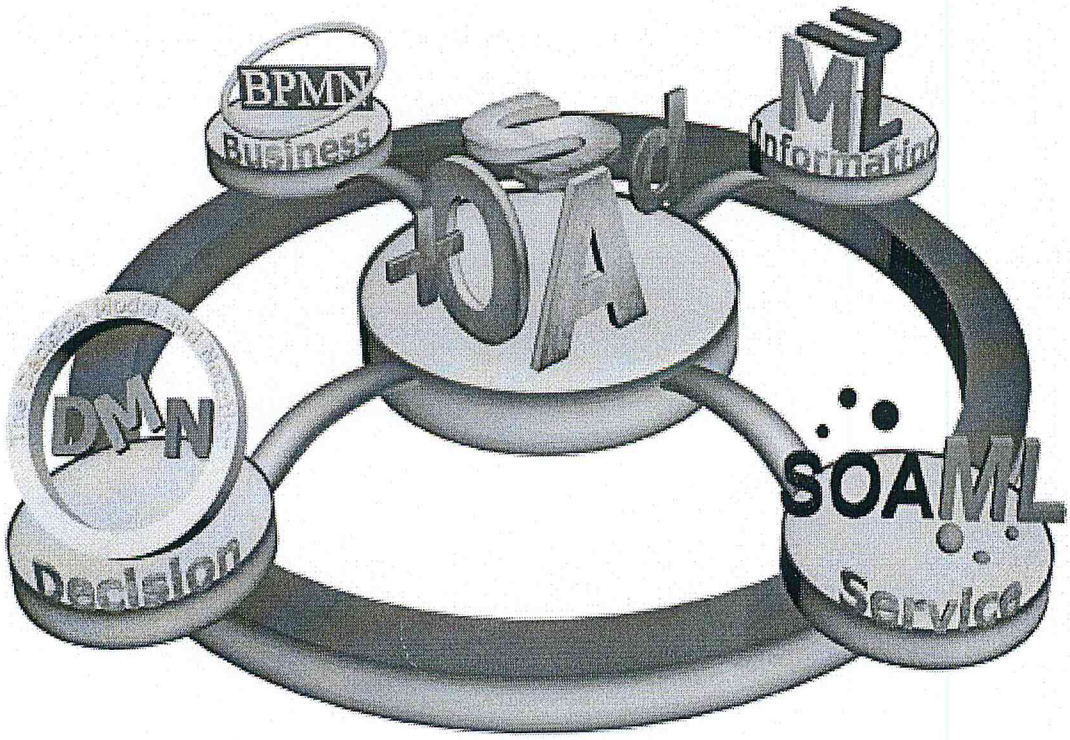


Figure III.1 L'utilisation des standards par SOA+d (Boumahdi, et al., 2015)

2. Description détaillée de la méthode Soa+d

SOA +d est une démarche méthodologique à quatre phases mettant en exergue une démarche de construction de l'architecture de services au sein de l'entreprise : la construction de la SOA métier, la construction de la SOA Informationnelle et la construction de la SOA Décisionnelle.

Chaque phase de la méthode SOA +d peut comporter une ou plusieurs étapes et peut être réalisée soit par un intervenant métier, un intervenant d'information, ou un intervenant de décision soit par les trois au même temps.

La méthode proposée (Boumahdi, et al., 2015) suit une approche descendante pour découvrir les services intervenant dans SOA +d.

Comme nous montre la Figure III.2, SOA+d s'articule au tourde quatre phases: analyse, identification et catégorisation des services, modélisation des services et la réalisation.

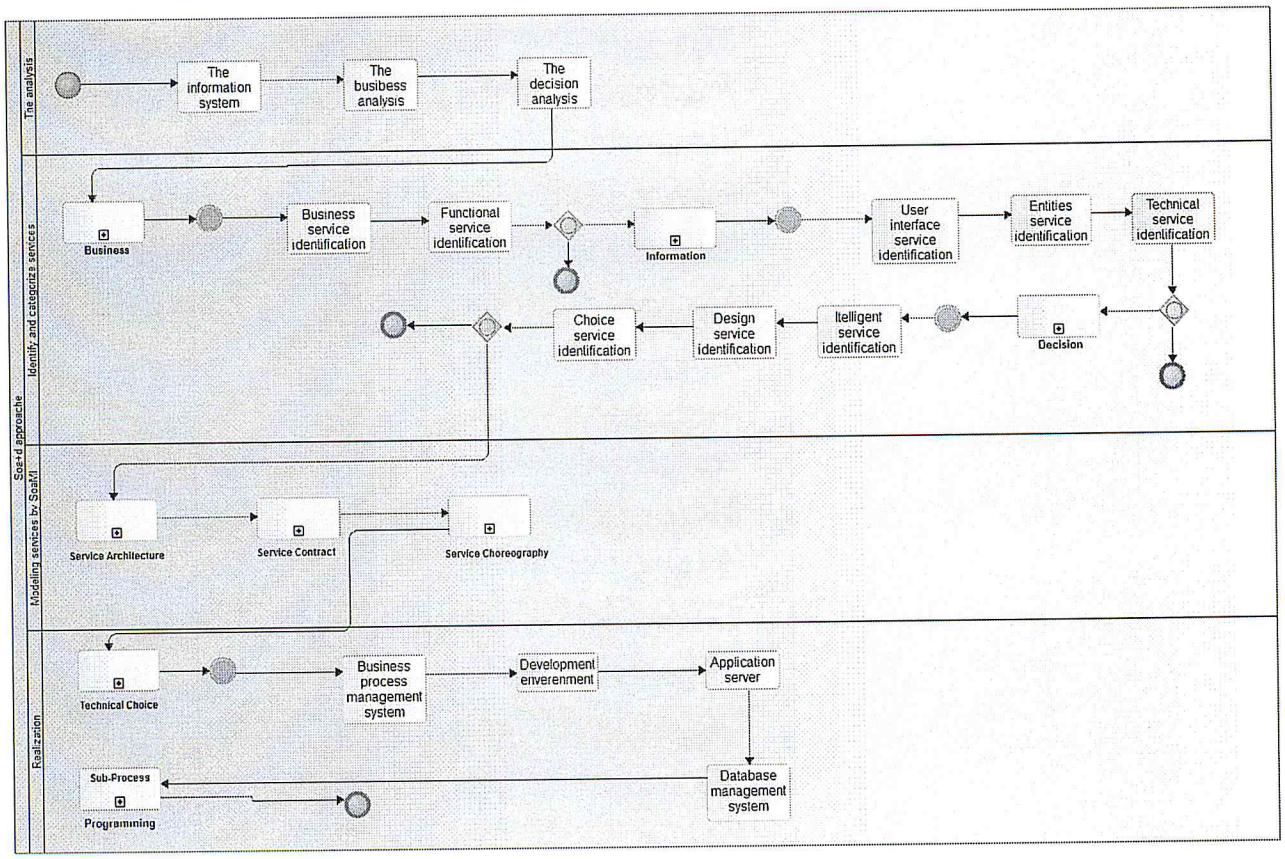


Figure III.2 La description des phases de SOA+davec la notation BPMN (Boumahdi, et al., 2015)

2.1. Phase 1 : La phase d'analyse

La première phase de SOA +d est réalisée conjointement par les intervenants métier, les intervenants d'information et les décideurs de l'entreprise. Elle comporte trois étapes majeures : l'élaboration du modèle métier, l'étude du système d'information et la construction d'une ou des solutions aux problèmes décisionnels.

La phase d'analyse est définie en se basant sur un ensemble de modèles permettant de décrire les exigences métiers. Principalement, cette phase se compose des modèles des cas d'utilisation, BPMN et des notations DMN. Les modèles des cas d'utilisation permettent la formalisation des besoins des utilisateurs associés aux différents acteurs interagissant avec le système. Les modèles BPMN permettent la description des différents scénarios de réalisation des cas d'utilisations. A la fin, le DMN décrit la conception des solutions des décisions.

Cette phase comprend trois étapes: l'analyse du système d'information, l'analyse métier et l'analyse de décision.

2.1.1. Activité 1 : L'analyse du système d'information

L'analyse des systèmes d'information est une étude détaillée des différentes opérations effectuées par un système et leurs relations au sein et en dehors du système. Une question clé est : que faut-il faire pour résoudre le problème? Le but de l'analyse est de définir les limites du système et déterminer si un système candidat devrait envisager d'autres systèmes liés.

La première phase de notre processus de développement est l'élaboration des modèles des cas d'utilisation. L'objectif de cette phase est la capture et la spécification des besoins des utilisateurs ainsi que la description des processus métiers. Dans cette phase, SOA+d détermine, en premier lieu, les différents acteurs du système ainsi que les besoins de chaque acteur via les diagrammes des cas d'utilisations. En effet, Les diagrammes des cas d'utilisation jouent un rôle très important pour l'identification des besoins des utilisateurs. Ils décrivent de manière exhaustive les exigences fonctionnelles du système.

- **Diagramme de cas d'utilisation**

Un cas d'Utilisation (« use case ») représente un ensemble de séquences d'actions qui sont réalisées par le système et qui produisent un résultat observable intéressant pour un acteur particulier.

Chaque cas d'utilisation spécifie un comportement attendu du système considéré comme un tout, sans imposer le mode de réalisation de ce comportement. Il permet de décrire ce que le futur système devra faire, sans spécifier comment il le fera (Roques, 2006).

2.1.2. Activité 2 : L'analyse métier

Le modèle de processus métier est utilisé pour comprendre et décrire les processus métier, ce dernier est lié à l'environnement dans lequel le système à construire sera utilisé. Cette phase comporte deux sous-phases :

- **Analyse des processus métier existants (As-Is) et cible (To- Be)**

La situation existante avant la prise de l'aspect décision est représentée avec les modèles As-Is (ceux qui existent actuellement); alors que la nouvelle situation est représentée avec les modèles To-Be (ceux qui doivent exister).

Les modèles As-Is décrivent les propriétés possédées par le système (propriétés indicatives) alors que les modèles To-Be décrivent les propriétés que le nouveau système possédera (propriétés opératives).

Cette étape examine l'ensemble des services existants dans le but de savoir quels sont les services qui sont déjà en place et quels sont ceux qui doivent être développés. Dans le cas où les analystes métier n'auraient jamais traité les problèmes de décision, il faut dresser une cartographie de l'existant (As-Is) et de la cible (To-Be). Sinon, si les participants prennent des décisions déjà il faut prendre en considération les modèles existants.

- **La modélisation des processus métier**

SOA+d adopte le langage BPMN 2.0 dont la sémantique est proposée par l'OMG.

Cette activité est l'une des activités clés dans le développement de services Métier (SM), car elle est l'entrée principale nécessaire pour comprendre et décrire le processus métier dans l'organisation. Nous utilisons BPMN pour les spécifier, qui fournit des éléments telsque les couloirs (pool, voie), objets de flux (activités, passerelles), élément de connexion (séquence, message) et des artefacts (groupes, données) pour modéliser le processus métier.

- **BPMN 2.0**

BPMN 2.0 est un langage de modélisation des processus dont la sémantique est supportée par un standard proposé par l'OMG.

La modélisation des processus métier avec une notation adaptée au métier comme BPMN contribue à faciliter l'alignement entre le métier et l'IT. En effet, BPMN fournit une notation intuitive et compréhensible à ses utilisateurs dans l'entreprise, depuis les analystes métier qui créent les ébauches initiales des processus jusqu'aux développeurs responsables de la mise en place de processus exécutables (LEMRABET, 2012).

2.1.3. Activité 3 : L'analyse de décision

SOA+d définit trois étapes dans l'analyse de décision :

- **Identifier les Décisions**

SOA +d propose d'identifier les problèmes de décision, de la manière suivante : à partir des cas d'utilisation et le BPMN,SOA+d détermine les mots ou les indices de décisions tels que : calculer, choisir, prévoir, déterminer, valider, évaluer la sélection et, bien sûr, décider. Cette tâche ou sous-processus est considéré comme un service de décisions candidat (Boumahdi, et al., 2015).

- **Spécifier les besoins de la décision**

Aller au-delà des simples descriptions des décisions pour spécifier les exigences détaillées de la décision. Le but de cette étape est la spécification des informations et des connaissances nécessaires pour prendre des décisions et de les combiner dans un graphe des besoins de la décision. Le diagramme des besoins de la décision de DMN est une notation représentant des éléments importants de la prise de décision et leurs dépendances (UDDI, 2007). Les éléments importants sont les décisions, connaissance métier, source de connaissance métier, et les données d'entrées (Boumahdi, et al., 2015).

- **Spécifier le niveau logique de la décision**

Le Niveau Logique de Décision de DMN (NLD) est l'étape pour spécifier la logique ou le raisonnement utilisé pour trouver une solution au problème de décision. Chaque décision dans un diagramme des besoins de décision de DMN peut avoir une expression de valeur, des règles métier, modèle de connaissance. Un modèle de connaissance métier peut être les spécifications de tables de décision, des modèles analytiques ou un algorithme (Boumahdi, et al., 2015).

- **DMN**

Malgré l'ancienneté du domaine de décision, les chercheurs n'ont pas pensé à un formalisme de modélisation pour ce domaine.

Un nouveau standard proposé par l'OMG (Object Management Group), dont la version 1.0 bêta de la spécification a été publiée le 01/02/2014 : DMN (Decision Model and Notation) (DMN, 2014).

L'objectif principal de DMN est de fournir une notation commune facile à comprendre par tous les utilisateurs du métier, à partir des besoins des analystes métier pour créer les exigences de la décision initiale, puis des modèles de décision plus détaillées, pour les développeurs techniques chargés d'automatiser les décisions dans les processus, et enfin, pour les personnels de métier qui permettront de gérer et de contrôler ces décisions.

2.2. Phase 2 : Identifier et catégoriser les services

Cette phase est basée sur les cartographies déjà étudiées par (Boumahdi, et al., 2015) pour identifier les services (métier, information et décision). Cette activité vise à identifier les services nécessaires à l'exécution du processus métier en cours de développement et c'est la première clé dans notre méthode. Les principales entrées de cette phase sont le BPMN, les cas d'utilisation et le DMN, qui sont spécifiés précédemment.

La découverte des services vise à apporter un soutien dans l'identification des services qui seront utilisés par les processus métier. Afin d'identifier et de décrire les services à partir des informations fournies par les cas d'utilisation, BPMN et le DMN. Nous proposons des règles de passage comme un guide pour identifier les services. Les règles proposées par (Boumahdi, et al., 2015) sont:

2.2.1. Règle de Mapping 1 : Cas d'utilisation Global vers le service métier

Chaque cas d'utilisation global correspond à un service métier Tableau III.1

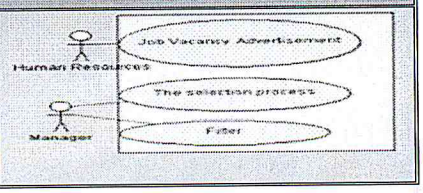
Cas d'utilisation	Service	Exemple
Cas d'utilisation Global	Métier	

Tableau III.1 Règle de Mapping 1 (Boumahdi, et al., 2015)

2.2.2. Règle de Mapping 2 : Cas d'utilisation CRUD vers le service CRUD

Dans le diagramme cas d'utilisation, les cas qui impliquent les objectifs : Créer, Lire, Mise à jour et Supprimer, seront considérés comme des services CRUD Tableau III.2

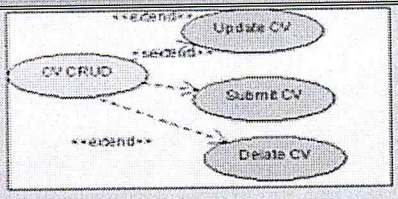
Cas d'utilisation	Service	Exemple
Cas d'utilisation CRUD	CRUD	

Tableau III.2 Règle de Mapping 2 (Boumahdi, et al., 2015)

2.2.3. Règle de Mapping 3 : Activité de BPMN vers le service Fonctionnel

Chaque activité identifiée dans le processus métier BPMN peut donner lieu à un service fonctionnel Tableau III.3

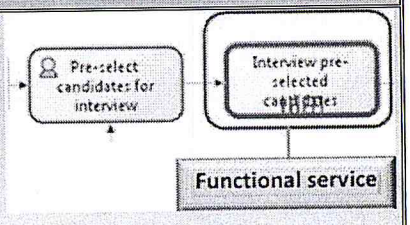
BPMN	Service	Exemple
Activité	Fonctionnel	

Tableau III.3 Règle de Mapping 3 (Boumahdi, et al., 2015)

2.2.4. Régle de Mapping 4 : Tâche de BPMN vers service Technique

Chaque tâche identifiée dans le modèle de processus métier peut donner lieu à un service technique Tableau III.4

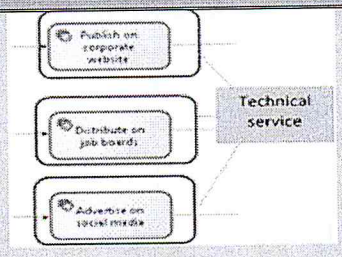
BPMN	Service	Exemple
Tache	Technique	

Tableau III.4 Règle de Mapping 4 (Boumahdi, et al., 2015)

2.2.5. Règle de Mapping 5 : Indice de Décision vers Service Décision

A partir du cas de d'utilisation et le BPMN, déterminer les indices des décisions (mots décision). Ces tâches ou sous-processus sont considérés comme des services de décisions Tableau III.5.

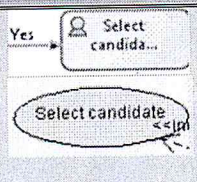
Cas d'utilisation +BPMN	Service	Exemple
Activité	Décision	

Tableau III.5 Règle de Mapping 5 (Boumahdi, et al., 2015)

2.2.6. Règle de Mapping 6 : Décisions de DMN vers Décision de service

Chaque décision du diagramme des exigences de décision correspond au service de décision Tableau III.6

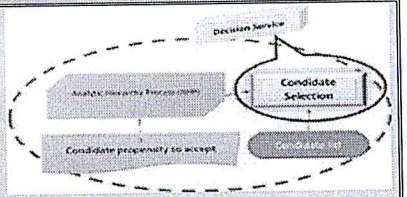
DMN	Service	Exemple
Décisions	Décision	

Tableau III.6 Règle de Mapping 6 (Boumahdi, et al., 2015)

2.2.7. Régle de Mapping 7 : Connaissance Métier DMN vers service de conception

Chaque connaissance métier à partir du diagramme des exigences de décision correspond au service de conception (Design Service) Tableau III.7.

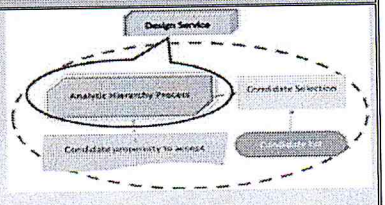
DMN	Service	Exemple
Connaissance métier	Conception	

Tableau III.7 Règle de Mapping 7 (Boumahdi, et al., 2015).

2.2.8. Règle de Mapping 8 : Source de Connaissance DMN vers service choix

Chaque source de connaissances du diagramme des exigences correspond à un service de choix Tableau III.8.

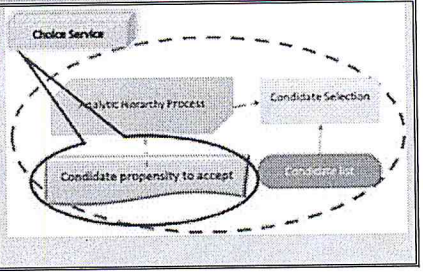
DMN	Service	Exemple
Source de connaissance	Choix	

Tableau III.8 Règle de Mapping 8 (Boumahdi, et al., 2015)

2.2.9. Règle de Mapping 9 : Données en entrée du DMN vers Service Intel-ligent

À partir du diagramme des exigences de décision, les données entrées seront considérées comme des services intelligents Tableau III.9

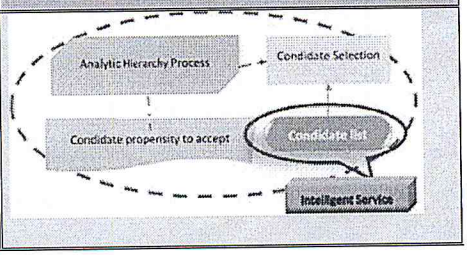
DMN	Service	Exemple
Donnée en entrée	Intelligent	

Tableau III.9 Règle de Mapping 9 (Boumahdi, et al., 2015)

2.3. Phase 3 : Modélisation de Services

La phase 3 se concentre sur la spécification des services (modèle de données, interfaces, etc.). Dans un projet SOA, il est possible de se retrouver confronté à la problématique suivante : comment modéliser les services et leurs interactions? La notation SoaML offre une solution standardisée et basée sur l'UML et centrée sur la notion d'architecture.

Dans cette phase, les services doivent être modélisés avec un formalisme, nous adoptons une spécification à la base du langage SoaML qui offre un haut niveau d'abstraction, il est alors nécessaire d'affiner les services pour les rendre propres à une plate-forme donnée. SoaML permet d'axer la modélisation sur :

- L'identification des services, leurs exigences et leurs dépendances.
- La spécification d'un service : capacité, protocoles, politiques et informations transitant entre un service client et un service fournisseur.

À ce niveau le problème consiste à identifier les services de la solution SOA. Le but est d'identifier les services métiers publics à partir des exigences exprimées avec les processus. La solution la plus simple consiste à créer un seul service et d'y mettre toutes les opérations découvertes. Dans ce cas tous les participants deviennent dépendent de ce même service. Cette solution entraîne un couplage fort entre les participants. Tout changement par le fournisseur du service affectera l'ensemble des participants qui le consomment.

2.3.1. SoaML

En proposant SoaML (Service Oriented Architecture Modeling Language) l'OMG a adapté UML qui est fortement imprégné des principes de l'orienté objet aux principes de l'orienté service (LEMRABET, 2012).

La spécification SoaML définit un langage de modélisation des services au sens SOA.

SoaML est avant tout destiné à modéliser des services dans des environnements distribués en prenant en compte les deux vues : métiers et IT. Le profil SoML permet de personnaliser les abstractions UML nécessaires à la modélisation des services. Il clarifie comment utiliser les métaclasses UML pour modéliser une solution SOA afin de garantir que les modèles des services reflètent les cinq principes SOA suivants : couplage faible, abstraction, réutilisation, autonomie et composition (LEMRABET, 2012).

2.3.2. La démarche de modélisation avec SoaML

D'abord, il faut commencer par identifier les participants à partir des processus BPMN.

Ensuite, il faut distinguer les capacités associées à chaque processus et à l'ensemble de la collaboration a fin de déterminer les rôles et les responsabilités de chaque participant.

Puis, il faut spécifier les types d'interactions de la collaboration (i.e., unidirectionnelle ou bidirectionnelle).

Ainsi, l'association des capacités aux interactions permet de déterminer les Consumers et les Providers. Enfin, il convient de concevoir les contrats de service qui sont nécessaires à la spécification de l'architecture globale des services publics.

➢ **Participants**

servent à définir les fournisseurs et les consommateurs dans un système. Un participant peut jouer le rôle de fournisseur de services, consommateurs, ou les deux au même temps.

➢ **Comment identifier les participants?**

Ces participants doivent être liés avec les pools et les voies dans les diagrammes BPMN pour assurer la cohérence entre les modèles BPMN et SoaML. Les pools identifiés dans le BPMN sont représentées comme participants en SoaML.

➢ **Architectures de services**

Est une description de la manière dont les participants (acteurs) s'échanges des services exprimés par les contrats de services.

Nous définissons les étapes de la modélisation des architectures des services comme suit:

- Créer une architecture de service. Créer une collaboration UML avec le stéréotype « Services Architecture » qui spécifie l'architecture services.
- Identifier les participants. Identifier les différents acteurs impliqués dans l'architecture de services. Cette étape est réalisée d'abord dans le cadre de la modélisation BPMN.
- Identifier les services contrats. Chaque service est représenté par un "Service Contract" qui définit le rôle joué par chaque participant du service (consommateur et fournisseur) ainsi que les interfaces qu'ils mettent en œuvre pour compléter ce rôle.

➢ **Des contrats de Service**

sont utilisés pour décrire les modes d'interaction entre des entités de services. Un contrat de service est utilisé pour modéliser un accord entre deux parties ou plus. Le rôle de chaque service dans un contrat de service est une interface qui représente généralement un fournisseur ou un consommateur

2.4. Phase 4 : La réalisation

Cette phase donne un aperçu des composants SOA et les options disponibles pour la mise en œuvre d'une architecture SOA à travers les Systèmes Open Source (SOS).

La Figure III.3 illustre les différents choix technique qui doivent être pris en considération pour implémenter une architecture SOA. Dans les sections ci-dessous, les éléments d'architecture sont mappés au Système Open Source (SOS) outils/produits proposés.

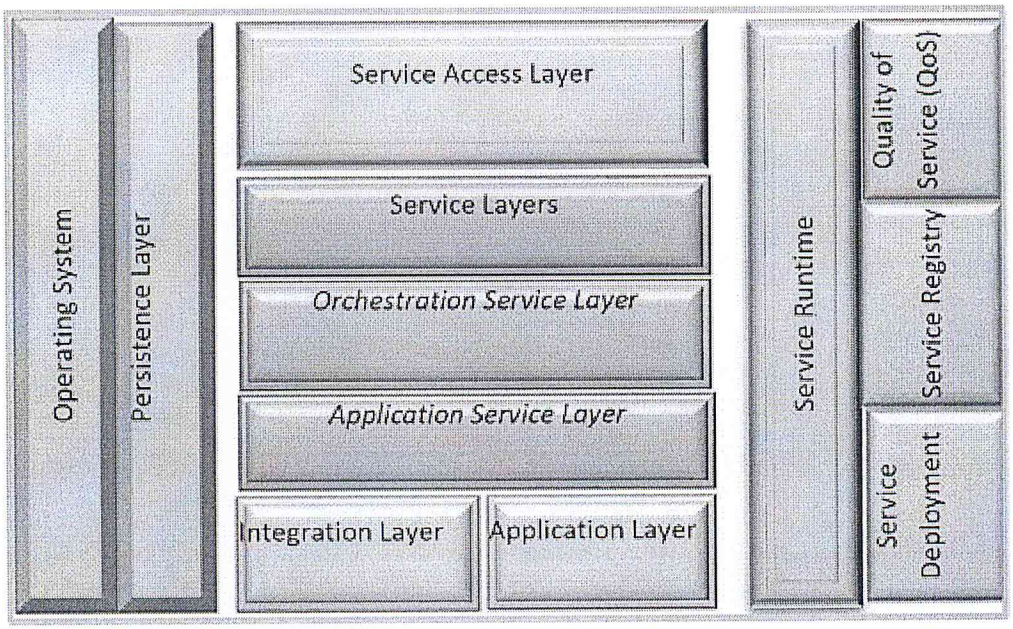


Figure III.3 Les choix technique pour implémenter une architecture SOA (Boumahdi, et al., 2015)

- **Couche Accès Service**

est la couche interface entre le client final et le système. Nous pouvons utiliser le Serveur Apache HTTP pour implémenter cette couche.

- **Couche Service**

cette couche définit et expose les fonctionnalités métier et application prises en charge par le système en tant que services. Sur la base de l'abstraction et de l'objectif, cette couche peut être encore divisée en :

- Couche service orchestration : Cette couche est fondamentale dans la réalisation de certains des principaux avantages de l'architecture SOA la composition, la réutilisation et l'abstraction. Les services de cette couche

définissent et exposent les processus métier du système. Comme outil nous pouvons utiliser :

- jBPM de JBoss permet la création de processus métier qui coordonnent entre les personnes, les applications et les services.
- Le Moteur Active BPEL est capable d'exécuter des définitions de processus créés pour le Langage Business Process Execution Language.

➢ **Couche Service Application**

Les services de cette couche sont les «travailleurs». Ils représentent l'application logique réelle nécessaire pour mettre en œuvre les processus métier. Nous pouvons implémenter cette couche en utilisant Apache Axis, qui est une implémentation de SOAP et permet le développement et le déploiement de services.

➢ **Couche Intégration**

Cette couche intègre le système orienté services pour les systèmes existants. C'est important lorsque les systèmes existants ne peuvent pas être exposés en tant que services, qui peuvent interopérer directement avec le système orienté services. ServiceMix est un exemple de cette couche d'outils de mise en œuvre.

➢ **Couche Application**

Cette couche accueille les implémentations des services.

➢ **Couche Persistance**

Il s'agit généralement de la couche de base de données, où les données fonctionnelles et non fonctionnelles sont conservées temporairement (cache) ou d'une manière permanente (durée déterminée par les exigences métier et techniques). Pour implémenter cette couche, nous pouvons utiliser : Apache Derby, PostgreSQL, MySQL, ou Hibernate.

➢ **Système d'exploitation**

C'est le système d'exploitation sur lequel les différentes piles du système orienté services sont hébergés. Comme logiciel Open Source, nous pouvons utiliser Linux.

➢ **Service Exécution**

Cette pile représente les composants nécessaires pour déployer et «exécuter» les services. Les éléments de base de cette pile sont :

- **Service Regitre (Service Registry)** : Le registre est le référentiel central où tous les services sont définis, exposés en utilisant les spécifications standards, et découverts par les deux systèmes internes et externes.

- **Qualité de Service** : cette composante représente la fourniture de capacités de transactions sécurisées et fiables à des niveaux de performance optimaux.
- **Déploiement de services** : Il s'agit de la plate-forme de déploiement de services. Cette plate-forme doit supporter les fonctionnalités d'une architecture orientée services.

3. Conclusion

Dans ce chapitre nous avons présenté la nouvelle méthode Soa+d, c'est une démarche top down, qui répond au manque de méthodes pour la construction de la SOA avec un aspect décisionnel. La démarche SOA +d a été décrite grâce à un processus de quatre phases qui guide les analystes dans le processus d'identification et de modélisation des services.

Dans la partie 2, nous utilisons cette méthode pour répondre aux problèmes de décision dans notre système de recrutement.

Le processus de recrutement inclut une décision principale : La sélection de (des) meilleur (s) candidat (s). Affin de répondre à ce problème nous nous intéressons à la méthode multicritère AHP, cette dernière sera présentée en détaille dans le prochain chapitre.

Chapitre IV

Processus de hiérarchie analytique

1. Introduction

Face à un développement de produits toujours plus rapides, les entreprises recherchent constamment à réutiliser leur connaissance, savoir-faire ou expertise. Les méthodes d'analyse se sont inscrites d'emblée dans ce cycle de capitalisation des connaissances et ont donné lieu à des résultats tangibles dans le monde industriel.

Dans ce qui suit, nous proposons un modèle d'aide à la décision basé sur une hybridation entre la méthode multicritère AHP et le Raisonnement à Partir de Cas. Ce modèle vise l'amélioration de la recherche de cas similaires suite à une décomposition et une hiérarchisation du système complexe en ses éléments constitutifs en tenant compte de ses relations essentielles.

L'application de la méthode AHP représente un outil capable d'arranger notre base de cas afin de faciliter l'aide à la décision.

2. Définition

à la base, il s'agit d'une méthode de prise de décision élaborée par Saaty (1980) (Saaty, 1980), destinée à prioriser les alternatives lorsque plusieurs critères doivent être pris en considération. Cette méthode permet au décideur de structurer des problèmes complexes, de les hiérarchiser.

Généralement la hiérarchisation comporte 3 niveaux, à savoir, le but, les critères et les alternatives. Selon Nydick et Hill, pour ce qui est de la sélection des fournisseurs, le but est la sélection du meilleur fournisseur. Les critères peuvent être la qualité, le prix, le service, la livraison, etc. Et les alternatives peuvent être propositions faites par les fournisseurs (JENN, 2008).

3. Hiérarchie de décision :

Les critères sélectionnés, constituant la hiérarchie, doivent être choisis en lien avec le but et également en regard de leurs influences marquées sur les alternatives. Ces dernières sont généralement mesurées par chaque critère. La hiérarchie est généralement construite sur au moins trois niveaux, comprenant le but, les critères et les alternatives. La nomenclature utilisée pour l'identification des niveaux peut être différente d'un auteur à l'autre, par exemple, elles peuvent être présentées comme suit : le but, objectifs et sous-objectifs et alternatives

(Ernest Forman, 2001). En somme, la dénomination utilisée peut varier d'un auteur à l'autre, mais le sens demeure semblable.

Voici à titre d'exemple une représentation schématisée de la structure de la hiérarchie généralement recherchée dans l'élaboration d'un problème de décision à résoudre: figure VI.1. (PANNETON, 2010)

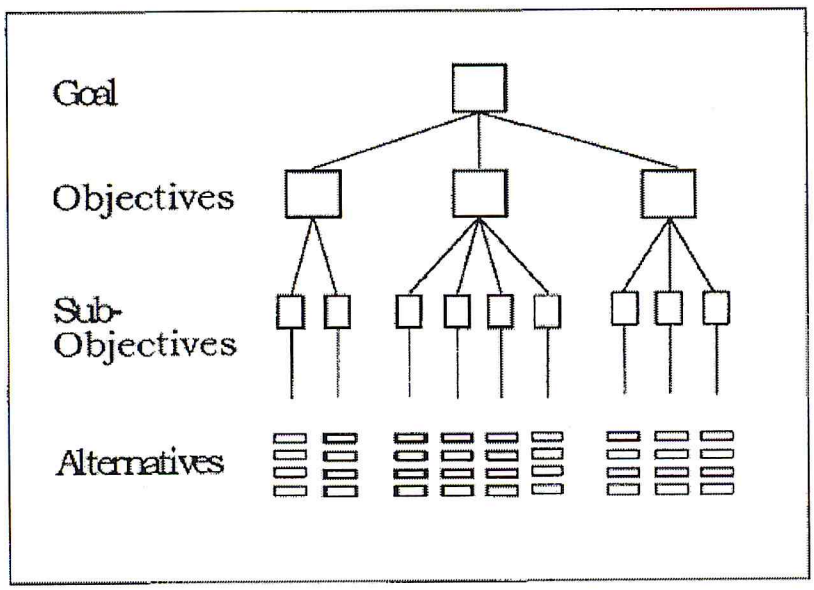


Figure IV.1 Hiérarchie de décision. (Ernest Forman, 2001)

4. Points forts de la méthode AHP

Les raisons de la popularité de la méthode AHP sont :

- **Structure hiérarchique** : tri les éléments d'un problème dans différents niveaux et dans des groupes à caractéristiques similaires.
- **Interdépendance** : permet de considérer l'interdépendance des éléments d'un problème sans insister sur le raisonnement linéaire (les éléments de chaque niveau de la structure hiérarchique sont dépendant).
- **Consistance** : permet de garder une consistance logique des jugements utilisés pour déterminer les priorités.
- **Synthèse**: permet d'obtenir une appréciation générale de la désirabilité de chaque Alternatif.
- **Identification des priorités**: permet de considérer la priorité relative de chaque critère pour ainsi obtenir le meilleur alternatif selon l'objectif identifié (ZOUGGARI, 2011).

5. Trois principes de la méthode AHP

cette méthode rigoureuse repose sur trois principes directeurs soit: la construction de hiérarchies, l'établissement de priorités et la vérification de la cohérence logique (Saaty, 1984).

6. Différentes étapes de la méthode AHP

6.1. Etape 1 : Décomposer le problème en une structure hiérarchique

Au début de la méthode, définissons une arborescence hiérarchique des critères, des sous critères et des alternatifs. Chaque élément de cette hiérarchie doit être défini Figure IV.2.

- Définir l'objectif (niveau 0).
- Définir les critères de décision ou de jugement (niveau 1).
- Le dernier niveau de la hiérarchie comprendra les différents alternatifs, choisies au préalable (niveau 2) (ZOUGGARI, 2011).

6.2. Etape 2 : Effectuer les combinaisons binaires niveau par niveau

- Comparer l'importance relative de tous les éléments appartenant à un même niveau de la hiérarchie deux par deux, par rapport à l'élément du niveau supérieur.
- Les matrices de comparaison ainsi obtenues auront la propriété suivante :
  - A = a_{ij}
  - a_{ii}=1 et a_{ji}=1/a_{ij} (ZOUGGARI, 2011).

6.3. Etape 3 : Déterminer les priorités

- Calculer l'importance relative de chacun des éléments de la hiérarchie à partir des évaluations obtenues de l'étape 2.
- La détermination des priorités des éléments de chaque matrice se fait par la résolution du problème de vecteur propre.

1. L'importance relative des différents critères est exprimée par les valeurs du vecteur normalisé.

2. Plus cette valeur est grande, plus le critère correspondant est important (ZOUGGARI, 2011).

6.4. Etape 4 : Synthétiser les priorités

- Une fois que les priorités pour tous les critères figurant dans la hiérarchie ont été déterminées, le poids de chaque alternatif est calculé et un classement des alternatifs est réalisé (ZOUGGARI, 2011).

La figure VI.2 illustre la structure hiérarchique de la méthode AHP.

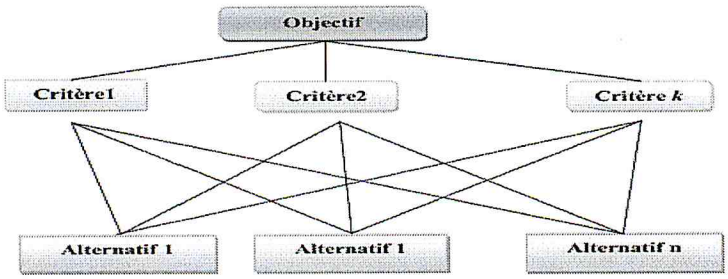


Figure IV.2 Structure hiérarchique de la méthode AHP (ZOUGGARI, 2011).

7. Exemple d'application de la méthode AHP

Nous présentons ici un exemple d'application de la méthode AHP montré sur la base de l'utilisation d'un tableur Excel. Également, dans le cadre de l'analyse multicritère, les jugements, c'est-à-dire les comparaisons par paires peuvent être faites soit de façon individuelle ou en groupe, c'est-à-dire par un décideur ou par un panel d'experts, par exemple. Pour ce dernier, un artifice mathématique permet de prendre en compte plusieurs avis, via une moyenne. Laquelle reflète l'importance attribuée aux critères considérant la variation de perception d'un individu à l'autre envers «ce qui est plus important» (Saaty, 1980).

Dans le cadre d'un programme d'attribution du budget régional complémentaire en maintien d'actifs immobilier, un gestionnaire doit procéder à l'analyse d'une série quatre projets qui lui sont présentés par un établissement. Les projets candidats déposés sont montrés au tableau VI.1.

Projets	Description	Valeur du projet	Participation demandée
1	Remplacement de la membrane de la toiture	780 000$	300 000$
2	Réparation de la maçonnerie des murs ouest et nord	1 250 000$	450 000$
3	Remplacement, mise aux normes des ascenseurs 3 et 4	275 000$	175000$
4	Mise aux normes du système d'alarme incendie	750 000$	195 000$

Tableau IV.1 Liste de projets déposés (PANNETON, 2010).

Cette analyse de priorisation des projets constitue le problème à résoudre. Lequel est décomposé en ses éléments sous la forme d'une structure hiérarchique.

Voici comment le gestionnaire présente la structure hiérarchique développée sur trois niveaux :

Le premier niveau comporte le but, c'est-à-dire de sélectionner un projet, parmi ceux présentés.

Le second niveau, montre les critères retenus pour cette analyse qui se présentent comme suit: l'état de vétusté de l'installation, l'aspect relatif à la sécurité de l'intervention et la capacité financière de l'établissement à supporter cette intervention. Le troisième niveau comprend les alternatives, représentées par les quatre projets, parmi lesquels le gestionnaire doit opérer le choix figure VI.3.

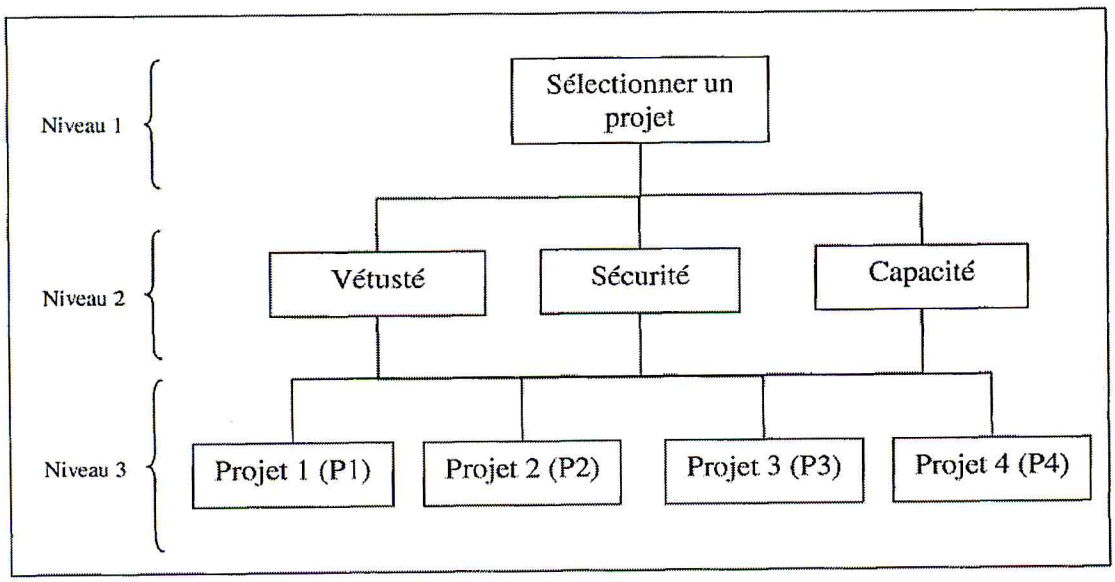


Figure IV.3 Hiérarchie de la structure du problème de sélection d'un projet (PANNETON, 2010).

Une fois la structure du problème développée, les comparaisons par paires sont réalisées pour chaque niveau. Les niveaux deux et trois sont traités dans cet ordre successivement. Les jugements sont d'abord exercés parmi les critères du deuxième niveau, c'est-à-dire entre les trois critères suivants : vétusté, sécurité et capacité. Suite à son analyse, le gestionnaire traduit ses jugements par paires et introduit les valeurs ordinales dans la matrice. Par la suite, une fois la normalisation de la matrice réalisée, le vecteur propre (Eigenvector) de la matrice est ainsi calculé figure VI.4.

	Vétusté	Sécurité	Capacité	Vecteur propre
Vétusté	1	1/2	3	0,3202
Sécurité	2	1	4	0,5571
Capacité	1/3	1/4	1	0,1226

$\lambda_{max} = 3,02$ CI = 0,00916 RI = 0,52

CR = 0,02 CR < 0,10 = ok

Figure IV.4 Matrice de comparaisons par paires des critères en regard de l'objectif (PANNETON, 2010).

À la figure IV.4, le vecteur propre exprime l'agrégation des critères en fonction des jugements. Selon les résultats de cette analyse le critère sécurité présente une forte prédominance, le critère vétusté est le deuxième en importance alors que le critère capacité est le moins important avec les résultats respectifs suivants: 0.5571,0.3202 et 0.1226.

La prochaine étape consiste à effectuer les comparaisons par paires de chacun des 4 projets, situés au troisième niveau, en regard de chaque critère situé au niveau supérieur soit par rapport à la vétusté, la sécurité et la capacité. Par la suite, après normalisation de la matrice, le vecteur propre (Eigenvector) de chaque matrice est calculé. La cohérence de l'ensemble des jugements est vérifiée pour chaque matrice, en s'assurant que l'indice CR soit égal ou inférieur à la valeur de 0, 1 O.

La figure IV.5 montre la matrice de comparaisons complétées en regard du respect du critère vétusté. L'indice CR est vérifié avec un résultat de 0,02, en conséquence acceptable. Sous ce critère la priorité des projets se classe comme suit, en ordre décroissant d'importance: P2, Pl, P3 et P4 avec des valeurs respectives de 0,4896, 0,3054, 0,1264, et 0,0786. Nous constatons que le critère vétusté influence de façon significative les projets P2 et Pl.

Vétusté	P1	P2	P3	P4	Vecteur propre
P1	1	1/2	3	4	0,3054
P2	2	1	4	5	0,4896
P3	1/3	1/4	1	2	0,1264
P4	1/4	1/5	1/2	1	0,0786

$\lambda_{max} = 4,05$ CI = 0,01619 RI = 0,89

CR = 0,02 CR < 0,10 = ok

Figure IV.5 Matrice de comparaisons considérant la vétusté (PANNETON, 2010).

La figure IV.6 montre la matrice de comparaisons complétée en regard du respect du critère sécurité. L'indice CR est vérifié avec un résultat de 0,09, de ce fait acceptable. Sous ce critère la priorité des projets se classe comme suit, en ordre d'importance : P4, P2, Pl et P3 avec des valeurs respectives de 0,6012, 0,2349, 0,0970, et 0,0669. Nous constatons que le critère sécurité influence de façon très marquée le projet P4.

Sécurité	P1	P2	P3	P4	Vecteur propre
P1	1	1/4	2	1/6	0,0970
P2	4	1	4	1/5	0,2349
P3	1/2	1/4	1	1/6	0,0669
P4	6	5	6	1	0,6012

$\lambda_{max} = 4,25$ CI = 0,08421 RI = 0,89

CR = 0,09 CR < 0,10 = ok

Figure IV.6 Matrice de comparaisons considérant la sécurité (PANNETON, 2010).

La Figure IV.7 montre la matrice de comparaisons complétée en regard du respect du critère capacité. L'indice CR est vérifié avec un résultat de 0,07, donc acceptable. Sous ce critère la priorité des projets se classe comme suit, en ordre d'importance: Pl, P3, P2 et P4 avec des valeurs respectives de 0,4657, 0,3510, 0,1077, et 0,0756. Nous constatons que le critère vétusté influence de façon très significative les projets Pl et P3.

Capacité	P1	P2	P3	P4	Vecteur propre
P1	1	4	2	5	0,4657
P2	1/4	1	1/6	2	0,1077
P3	1/2	6	1	4	0,3510
P4	1/5	1/2	1/4	1	0,0756

$\lambda_{max} = 4,19$ CI = 0,06192 RI = 0,89

CR = 0,07 CR < 0,10 = ok

Figure IV.7 Matrice de comparaisons considérant la capacité (PANNETON, 2010).

La prochaine étape consiste à déterminer la performance de la valeur relative de chaque projet sous la considération des critères situés au niveau supérieur: Figure IV.8.

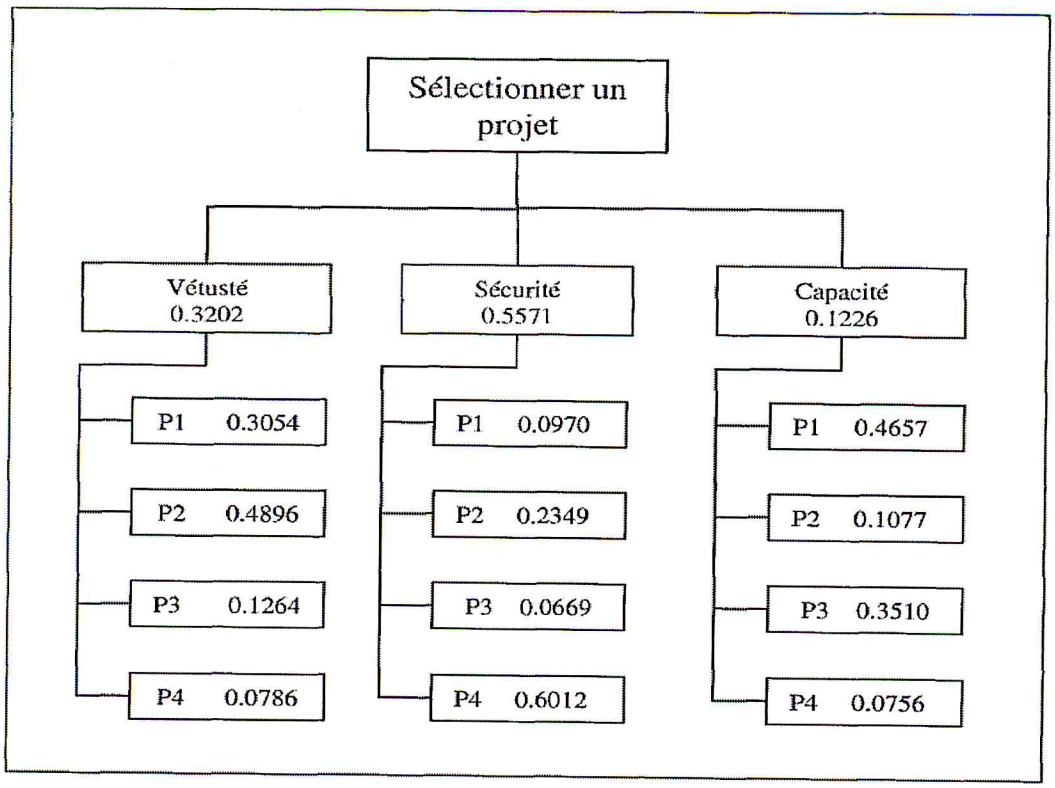


Figure IV.8 Performance relative de chaque critère de la hiérarchie (PANNETON, 2010).

En regard des étapes précédentes, et de 1'ensemble des résultats obtenus, voici 1'établissement de la priorité des projets : figure IV.9.

	vétusté	sécurité	capacité		priorités des critères		priorités des projets
P1	0,3054	0,0970	0,4657	x	0,3202	=	0,2089
P2	0,4896	0,2349	0,1077		0,5571		0,3008
P3	0,1264	0,0669	0,3510		0,1226		0,1208
P4	0,0786	0,6012	0,0756				0,3694

Figure IV.9 Agrégation des projets (PANNETON, 2010).

Le résultat final, exprimé par le vecteur des priorités des projets montré à la figure IV.9., indique que le projet P4 se révèle être le meilleur choix obtenant un résultat de 36,94 %. Il est suivi du projet P2, du projet Pl et finalement du projet P3 avec respectivement 30,08 %, 20,89 % et 12,08 %. Dans cet exemple le résultat de l'analyse suggère d'accorder, sans équivoque, la subvention au projet P4 puisque celui-ci récolte de l'ensemble des jugements du

décideur l'appréciation la plus élevée. Voici les résultats présentés sous le format de la hiérarchie développée initialement : figure IV.10.

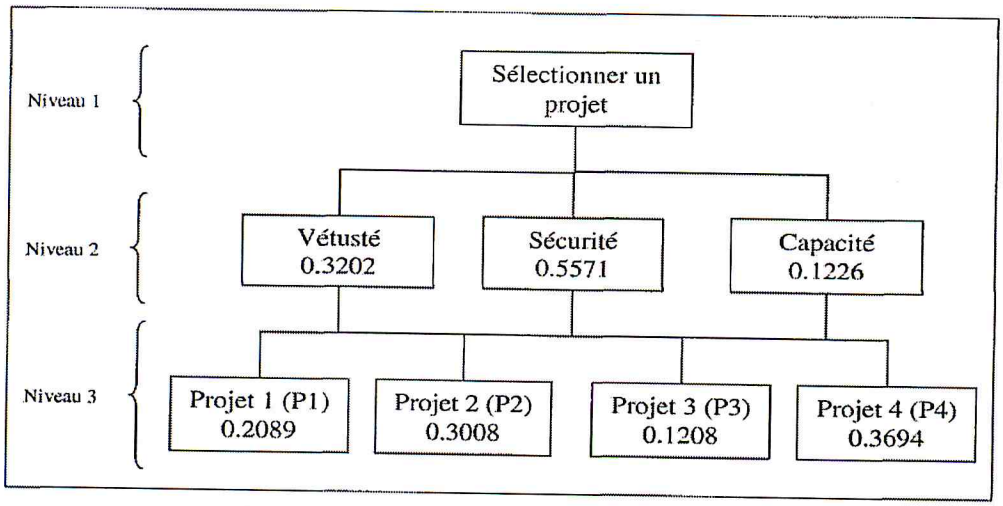


Figure IV.10 Performance relative des critères et des alternatives (PANNETON, 2010).

La Figure IV.10 indique qu'au niveau 2 les critères vétusté et sécurité ont prédominé et guidés de façon importante l'identification du projet à retenir situé au niveau 3. Nous en déduisons que le projet 4 rencontre ces caractéristiques d'où 1' obtention du poids relatif le plus élevé (PANNETON, 2010).

8. Avantage de la méthode AHP :

Soulignons que la méthode AHP revêt un avantage considérable permettant de traiter des critères de natures variées et hétéroclites, comme il nous est possible d'en rencontrer dans plusieurs aspects de la vie. En effet, la comparaison par paires, dite binaire, se prête à l'intégration logique de critères dont la teneur se situe autant au niveau qualitatif que quantitatif, au niveau subjectif qu'objectif, ainsi qu'au niveau intangible que tangible, (PANNETON, 2010).

9. Conclusion

Dans ce chapitre, nous avons présenter la méthode AHP qui permet de faciliter l'aide à la décision, en découvrant leurs différents étapes et leurs principes qui sont utile par la suite pour la phase d'aide à la décision.

Dans la partie 2, nous allons adapté la méthode AHP dans notre contexte, qui est la sélection du meilleur candidat

Partie 2 la conception de l'architecture orientée service

Chapitre V La démarche de développement

1. Introduction

Dans ce chapitre, nous présentons la démarche de développement de notre projet, en parcourant les différents aspects proposés par la méthode soa+d. Et à la fin nous terminons par la présentation de l'architecture technique de notre solution.

2. Phase1 : analyse

2.1. analyse du système d'information

SOA+d propose l'utilisation du diagramme de cas d'utilisation pour modéliser l'aspect système d'information. La Figure V.1 illustre le diagramme le cas d'utilisation du processus de recrutement.

Figure V.1 Diagramme de cas d'utilisation pour le recrutement

2.2. Analyses métier

Le processus de recrutement de l'hôpital CHU de Blida inclut les sous processus :

- **Sous processus 1: Etude et analyse des besoins**

Dans le cadre de l'ouverture des postes budgétaire un plan est confectionné dans ce domaine qui est le PAGRH (Plan annuaire de la gestion des ressources humaines).

Le PAGRH est le moyen d'exprimer le besoin de l'administration pour l'ouverture des multiples postes budgétaire en collaboration avec la direction des finances.

Donc par biais PAGRH ils ont étudié et ouvrir les postes budgétaire

Une fois que l'entreprise a analysé le besoin en recrutement, elle doit s'intéresser à la définition du poste.

La définition du poste va s'appuyer sur une grille d'analyse qui comprend différentes rubriques. Celles-ci sont :

» Identification du poste avec son intitulé

» Mission de l'emploi et les responsabilités.

» Inventaire des tâches de travail

» Description des activités

» Moyens et contraintes du poste

» Mode d'accès et évolution

» Compétences requises

à travers cette description de poste, on veut également spécifier les conditions spécifiques de travail.

La définition du profil se fait à partir de la description de poste en tenant compte des contraintes

Le DRH (Directeur des ressources humaines) va définir des compétences et des qualités personnelles que le futur candidat devra posséder. Il précisera également la formation souhaitée et l'expérience requise.

➢ **Sous processus 2: La recherche de candidats**

La rédaction de l'annonce de l'offre d'emploi est une retranscription de la définition du poste et du profil recherchés.

➢ **Sous processus 3: Sélectionner des candidats :**

faire un entretient pour vérifier en quoi et comment les compétences du candidat sont à même de répondre au poste à pourvoir.

La commission trie et choisie le meilleur candidats. Notre système va aider la commission pour le classement et le choix du (des) meilleur (s) candidat (s), pour ce faire nous adoptons la méthode AHP.

Le DRH informe le meilleur candidat pour compléter le dossier et signer le procès verbal d'installation

➢ **Sous processus 4: Recrutement**

Le DRH intègre le nouveau recrue dans le cadre organisationnel.

La Figure V.2 montre le processus métier « Recrutement » de l'hôpital CHU de Blida.

2.2.1.1. Processus de recrutement BPMN

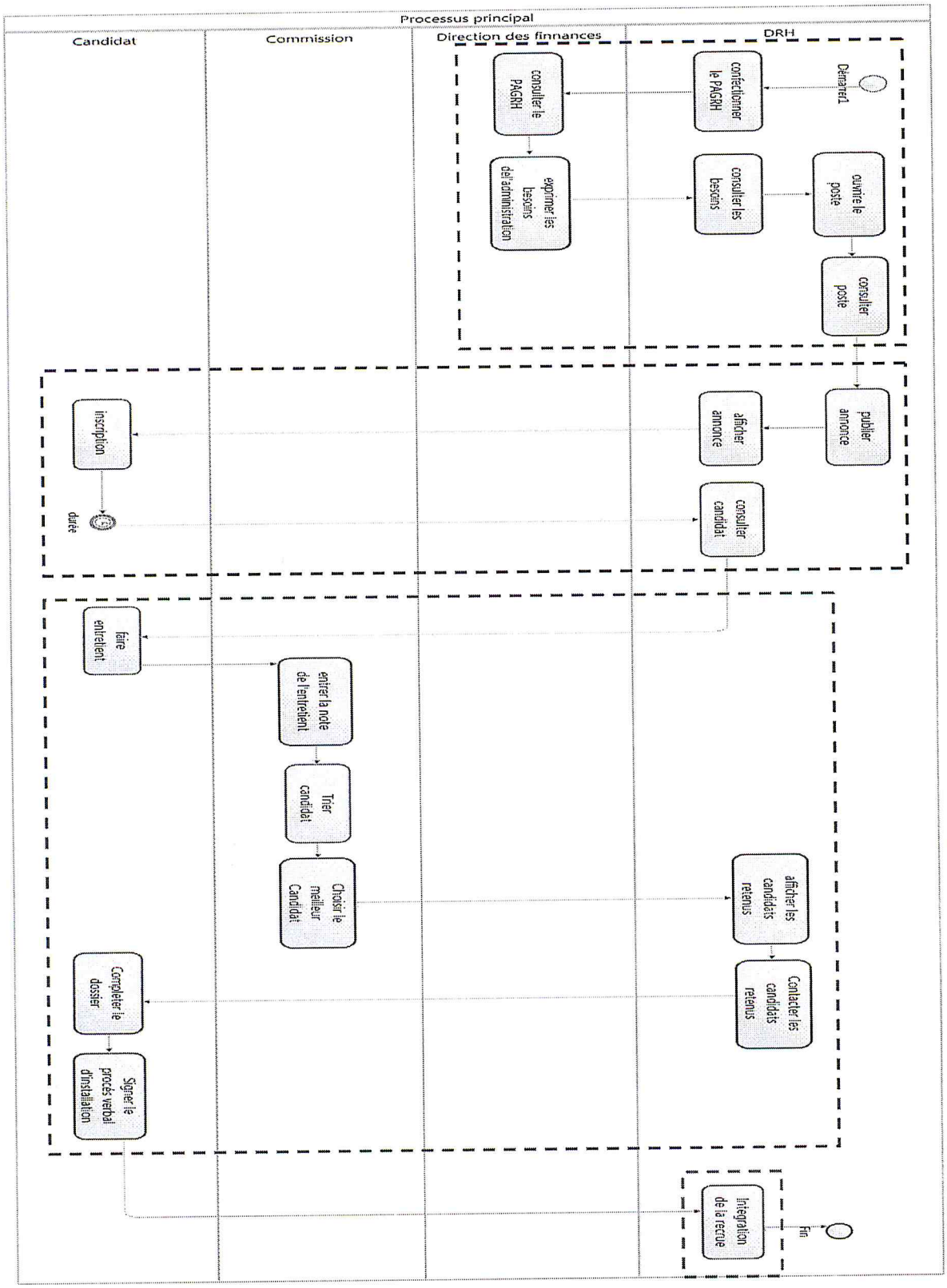


Figure V.2 Processus de recrutement BPMN

2.3. Analyse de la décision

D'après le cas d'utilisation et le BPM, nous avons un problème de décision qui est le trie et la sélection des candidats.

La sélection du candidat est un problème complexe impliquant des critères qualitatifs des candidats. La première étape de toute procédure de sélection des candidats est de trouver les critères appropriés à utiliser pour évaluer le candidat

Afin de sélectionner le bon candidat, l'approche analyse hiérarchique du processus (AHP) (Saaty, 1980) a été adoptée. Le AHP est une théorie de la mesure à travers des comparaisons par paires et s'appuie sur les jugements d'experts pour établir les échelles de priorité. Cette méthode d'analyse multicritère constitue en soi une démarche structurée complémentaire afin de traiter de manière organisée l'ensemble des informations pertinentes à l'évaluation de la problématique.

La Figure V.3 montre le DMN de la décision « trier les candidats ».

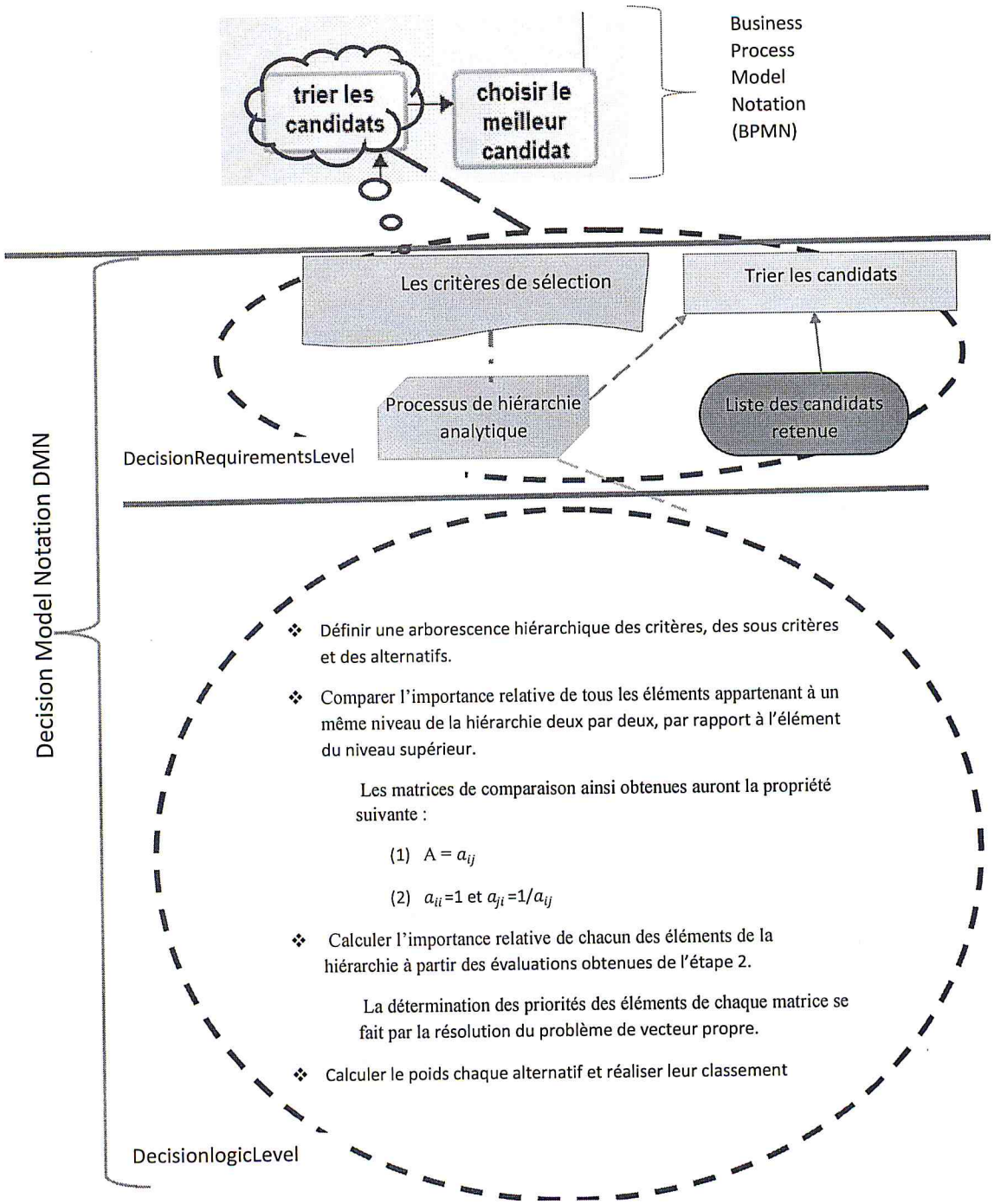


Figure V.3 Trier les candidats DMN

Exemple

Une organisation cherche un nouveau directeur pour l'un de ses départements. La décision sera basée sur quatre facteurs :

- L'âge du candidat.
- Les années d'expérience pertinentes du candidat.

- Le niveau d'éducation du candidat.
- L'appui de la direction pour ce candidat.

L'organisation considère l'expérience du candidat comme le facteur le plus important dans la décision, et donne moins d'importance au niveau d'éducation.

Le candidat idéal a le profil suivant :

- Le plus d'années d'expérience pertinentes possible.
- Un excellent appui de la direction.
- Agé de 40 ans.
- Maitrise universitaire

Après avoir considéré plusieurs employés dans différents département, trois candidats se distingues des autres.

Nom	Serge	Jean	Marc
Âge	40 ans	45 ans	55 ans
Expérience pertinente	15 années	20 années	30 années
Niveau d'éducation	Baccalauréat	Maîtrise	Baccalauréat
Appui de la direction	Très bon	Bon	Bon

Tableau V.1 Liste finale de candidats pour le poste de directeur.

Etape 1

Créer d'un diagramme à l'aide de l'objectif, les critères de décision et les solution alternatives.

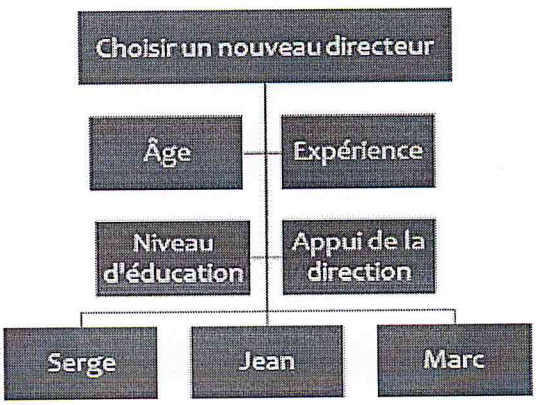


Figure V.4 diagramme étape 1.

Etape 2

- Création d'une matrice des critères de décisions afin de les comparer.

	Âge	Expérience	Éducation	Appui
Âge	1	1/4	2	1/2
Expérience	4	1	5	2
Éducation	1/2	1/5	1	1/3
Appui	2	1/2	3	1

Tableau V.2 Matrice des critères de décisions.

- Déterminer l'importance de chaque critère par rapport à un autre selon l'échelle ci-dessous :

1 = importance égale

2 = importance modérée

3 = importance forte

4 = importance très forte

5 = importance extrême

Ratio = importance du critère sur la ligne / importance du critère dans la colonne.

Par exemple, l'expérience du candidat a une très forte importance à comparer à l'âge. Le ration entre l'expérience (ligne) et l'âge (colonne) est de4/1 et celui entre l'âge et l'expérience est de ¼.

- Exprimer chaque ratio en pourcentage relatifs.

	Âge	Expérience	Éducation	Appui	Moyenne
Âge	0.133	0.128	0.182	0.130	0.143
Expérience	0.533	0.513	0.455	0.522	0.506
Éducation	0.067	0.103	0.091	0.087	0.087
Appui	0.267	0.256	0.273	0.261	0.264

Tableau V.3 Matrice des critères de décisions en pourcentage relatifs

Prendre les données de la « matrice des critères de décisions » pour faire le calcul du pourcentage relatif.

Exemple, pourcentage relatif de la cellule « âge/âge » : 1/(1+4+1/2+2) = 0.133.

- Calculer la moyenne de chaque critère de décision.

Exemple, calcul de la moyenne du critère « âge » : (0.133+0.128+0.182+0.130)/4 = 0.143.

Etape 3

- Pour chaque critère de décision, créer une matrice qui permet de comparer les différentes solutions.
- Utiliser l'échelle de l'étape 2 pour remplir la matrice.
- Pour chaque critère de décision, créer une seconde matrice en pourcentage relatif.
- Calculer la moyenne.

	Serge	Jean	Marc
Serge	1	2	3
Jean	1/2	1	2
Marc	1/3	1/2	1
Total	11/6	7/2	6

Tableau V.4 Matrice de comparaison "âge"

	Serge	Jean	Marc	Moyenne
Serge	0.545	0.571	0.500	0.539
Jean	0.273	0.286	0.333	0.297
Marc	0.182	0.143	0.167	0.164

TableauV.5 Matrice avec les pourcentages relatifs "âge"

	Serge	Jean	Marc
Serge	1	1/2	1/3
Jean	2	1	1/2
Marc	3	2	1
Total	6	7/2	11/6

Tableau V.6 Matrice de comparaison "expérience"

	Serge	Jean	Marc	Moyenne
Serge	0.167	0.143	0.182	0.164
Jean	0.333	0.286	0.273	0.297
Marc	0.500	0.571	0.545	0.539

Tableau V.7 Matrice avec les pourcentages relatifs "expérience"

	Serge	Jean	Marc
Serge	1	1/3	1
Jean	3	1	3
Marc	1	1/3	1
Total	5	5/3	5

Tableau V.8 Matrice de comparaison "éducation"

	Serge	Jean	Marc	Moyenne
Serge	0.2	0.2	0.2	0.2
Jean	0.6	0.6	0.6	0.6
Marc	0.2	0.2	0.2	0.2

Tableau V.9 Matrice avec les pourcentages relatifs "éducation"

	Serge	*Jean*	*Marc*
Serge	1	3	3
Jean	1/3	1	1
Marc	1/3	1	1
Total	5/3	5	5

Tableau V.10 Matrice de comparaison "appui"

	Serge	Jean	Marc	Moyenne
Serge	0.6	0.6	0.6	0.6
Jean	0.2	0.2	0.2	0.2
Marc	0.2	0.2	0.2	0.2

Tableau V.11 Matrice avec les pourcentages relatifs "appui"

Exemple de calculs pour la matrice « âge » :

- Calcul du pourcentage relatif de serge/serge = 1/(11/6)
- Calcul de la moyenne serge =(0.545 + 0.571 + 0.5)/3=0.539

Etape 4

- Créer une matrice de solutions en y insérant les solutions dans la première colonne et les critères de décision dans la première rangée.
- Utiliser les moyennes calculées à l'étape 3 pour remplir la matrice de solutions.

	Âge	Expérience	Éducation	Appui
Serge	0.539	0.164	0.2	0.6
Jean	0.297	0.297	0.6	0.2
Marc	0.164	0.539	0.2	0.2

Tableau V.12 Matrice de solutions

Etape 5

- Multiplier la matrice de solutions (étape 4) avec la matrice des moyennes des pourcentages relatifs (étape 2).
- Choisir la meilleure solution.

	Âge	Expérience	Éducation	Appui
Serge	0.539	0.164	0.2	0.6
Jean	0.297	0.297	0.6	0.2
Marc	0.164	0.539	0.2	0.2

*

	Moyenne
Âge	0.143
Expérience	0.506
Éducation	0.087
Appui	0.264

=

Candidats		
Serge	0.336	= 0.539*0.143+ 0.164*0.506 + 0.2*0.087 + 0.6*0.264
Jean	0.298	= 0.297*0.143 + 0.297*0.506 + 0.6*0.087 + 0.2*0.264
Marc	0.366	= 0.164*0.143 + 0.539*0.506 + 0.2*0.087 + 0.2*0.264

Tableau V.13 solution finale

Marc est le gestionnaire qui correspond le plus aux besoins de l'organisation, suivi de Serge et de Jean.

Choisir Marc pour le poste de directeur.

3. Phase 2 : Identifier et classer les services

SOA+d identifie trois types de services : services qui existent au niveau métier, système d'information et les services décisionnels. Les services de la vision Métier présentés dans la Figure V.5.

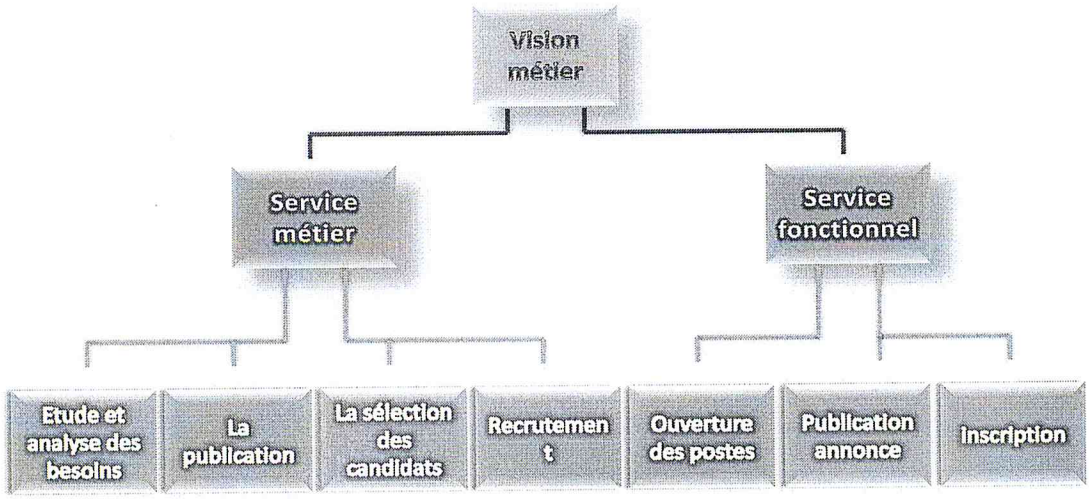


Figure V.5 Les services de la vision Métier

Après avoir détaillé le diagramme de cas d'utilisation, et appliquer les règles de passage définies dans le chapitre précédent, nous avons identifie les services de la dimension « Système d'information», qui sont présentés dans la Figure V.6

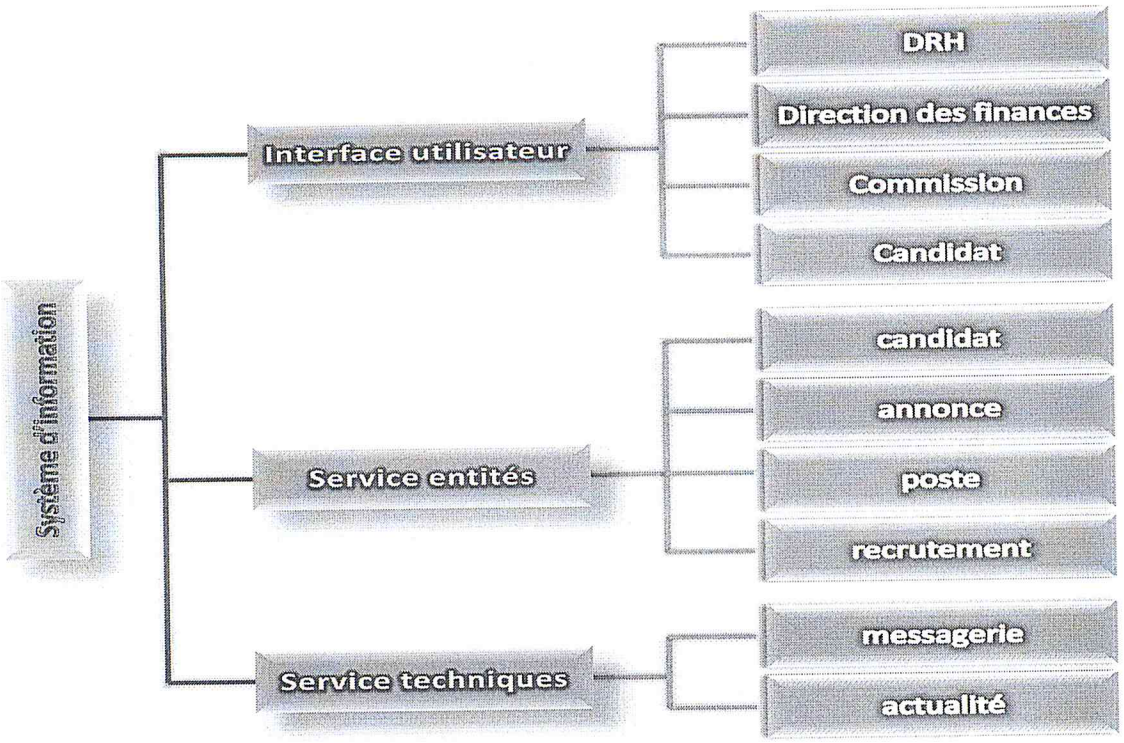


Figure V.6 Les services de la vision «Système Information»

Afin de répondre aux problèmes de décision identifiés dans la Phase 1, un ensemble de services est construit à la base des règles de passage définies dans le chapitre précédent. La vision décision est présentée dans la Figure V.7.

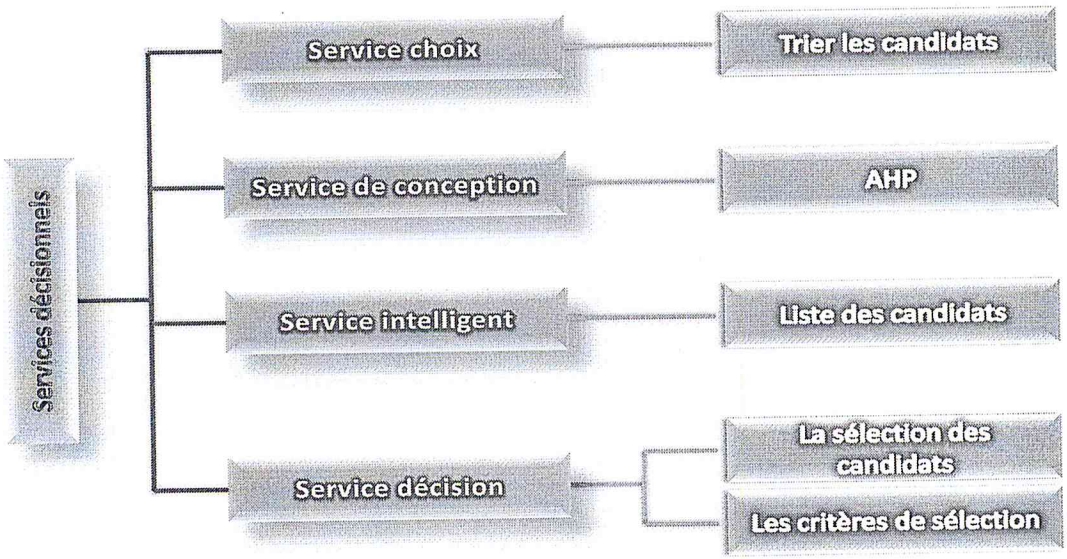


Figure V.7 Les services de la vision «Décision»

4. Phase 3 : spécification des services

Pour passer d'un modèle BPMN à un modèle SoaML, il faut respecter un ensemble de règles proposées par (Brian Elvesæter, 2010) , dont le but est de présenter la nouvelle version du langage de modélisation de l'architecture SOA, SoaML.

➢ **Règle 0**

Le processus à l'architecture de services : L'architecture de services est alignée avec le processus métier. Les participants sont dérivés des Pools ou bien des lignes et les contrats de services sont dérivés des activités ou bien des interactions entre participants.

➢ **Règle 1 : La Tâche à l'action UML**

Une tâche décrit une activité qui peut être consommée par les acteurs du processus. Elle représente une interface abstraite pour le travail accompli sans donner plus de précisions sur le flux de travail exécuté. Le Tableau V.14 illustre la correspondance des notations.

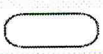
	BPMN	SoaML
Construct	Task	Action
Notation		

Tableau V.14 Tache à l'action UML (Brian Elvesæter, 2010).

➢ **Règle 2 : Sous-processus à l'architecture de services**

Un sous-processus représente une fonction plus complexe que la tâche. Il est à mentionner que l'architecture des services correspondant au sous-processus n'est pas de faible granularité. Le Tableau V.15 illustre la correspondance des notations.

	BPMN	SoaML
Construct	Sub-process	Services Architecture
Notation		

Tableau V.15 Sous-Processus à l'architecture de services (Brian Elvesæter, 2010).

➢ **Règle 3 : Les groupements (Pool) au participant Un Pool représente un participant important dans un processus en BPMN.**

Il peut également être structuré à l'égard des autres participants du processus, créant ainsi un des participants hiérarchiques. On représente le Pool dans le modèle SoaML par un rôle dans une architecture de services qui dispose d'un participant de type Pool.

Le Tableau V.16 illustre la correspondance des notations.

مكتبة الكلية

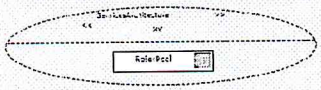
	BPMN	SoaML
Construct	Pool	Participant Role in a Community-level Services Architecture
Notation		

Tableau V.16 Pool au Participant (Community-level) (Brian Elvesæter, 2010).

- **Règle 4 : La Ligne (Lane) au participant**

Une ligne représente un participant ou un département en BPMN, elle est située dans un Pool, montrant ainsi la hiérarchie à deux niveaux. Afin de montrer la possibilité pour une nouvelle subdivision, la ligne est représentée par un rôle dans un participant au niveau de l'architecture des services de type ligne. L'architecture des participants du type ligne doit adhérer à l'architecture des services dont les participants du type Pool (les lignes sont contenus dans ces Pool).Le Tableau V.17 illustre la correspondance des notations.

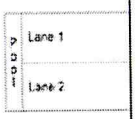
	BPMN	SoaML
Construct	Lane	Participant
Notation		

Tableau V.17 Ligne (Lane) au participant (Brian Elvesæter, 2010).

- **Règle 5 : Le début de Flux de message au service**

Le point de départ de chaque message dans BPMN représente un canal de transmission de données entre deux participants ou deux Pools. Dans le SoaML, c'est un port d'un service. Le Tableau V.18 illustre la correspondance des notations.

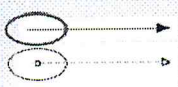
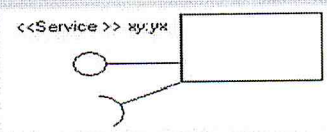
	BPMN	SoaML
Construct	Message "Begin"	Service
Notation		

Tableau V.18 Le début de Flux de message au service (Brian Elvesæter, 2010).

➢ **Règle 6 : La fin de Flux de message à la demande**

Le point de fin de chaque message dans BPMN représente un canal de transmission de données entre deux participants ou deux Pools. Dans le SoaML, c'est un port de demande d'un service. Le Tableau V.19 illustre la correspondance des notations.

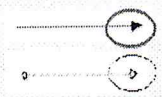
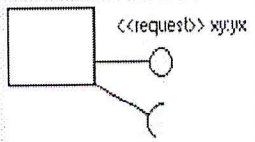
	BPMN	SoaML
Construct	Message "End"	Request
Notation		

Tableau V.19 La fin de Flux de message à la demande (Brian Elvesæter, 2010).

➢ **Règle 7 : fragment de processus (modèle) au contrat de service**

Il n'y a aucune construction unique en BPMN qui ressemble à un contrat de service. Il faut analyser les processus BPMN et identifier les fragments de processus qui peuvent être mappées à des contrats de service.

Un contrat de service définit un cahier des charges de service qui définit le rôle joué par chaque participant dans le service, et les interfaces qu'ils mettent en œuvre pour compléter ce rôle. On peut cependant, définir un modèle BPMN qui peut être associé à un contrat de service. Le Tableau V.20 illustre la correspondance des notations.

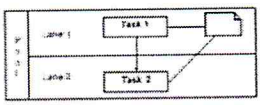
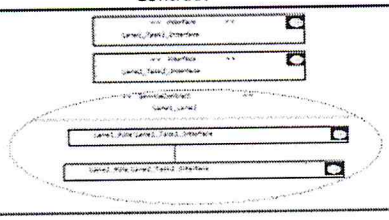
	BPMN	SoaML
Construct	Lane1 → Lane2	Service Contract
Notation		

Tableau V.20 fragment de processus (modèle) au contrat de service (Brian Elvesæter, 2010).

4.1.1. Application des règles de correspondance sur le processus de recrutement

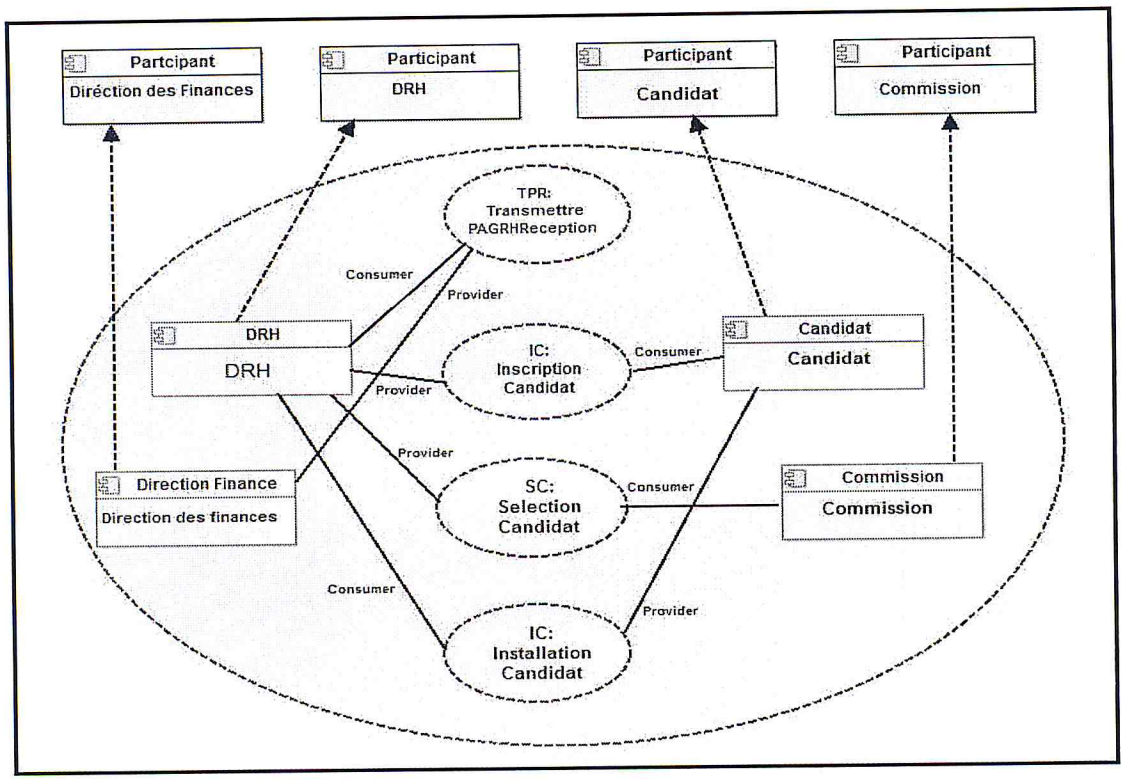


Figure V.8 Architecture générale.

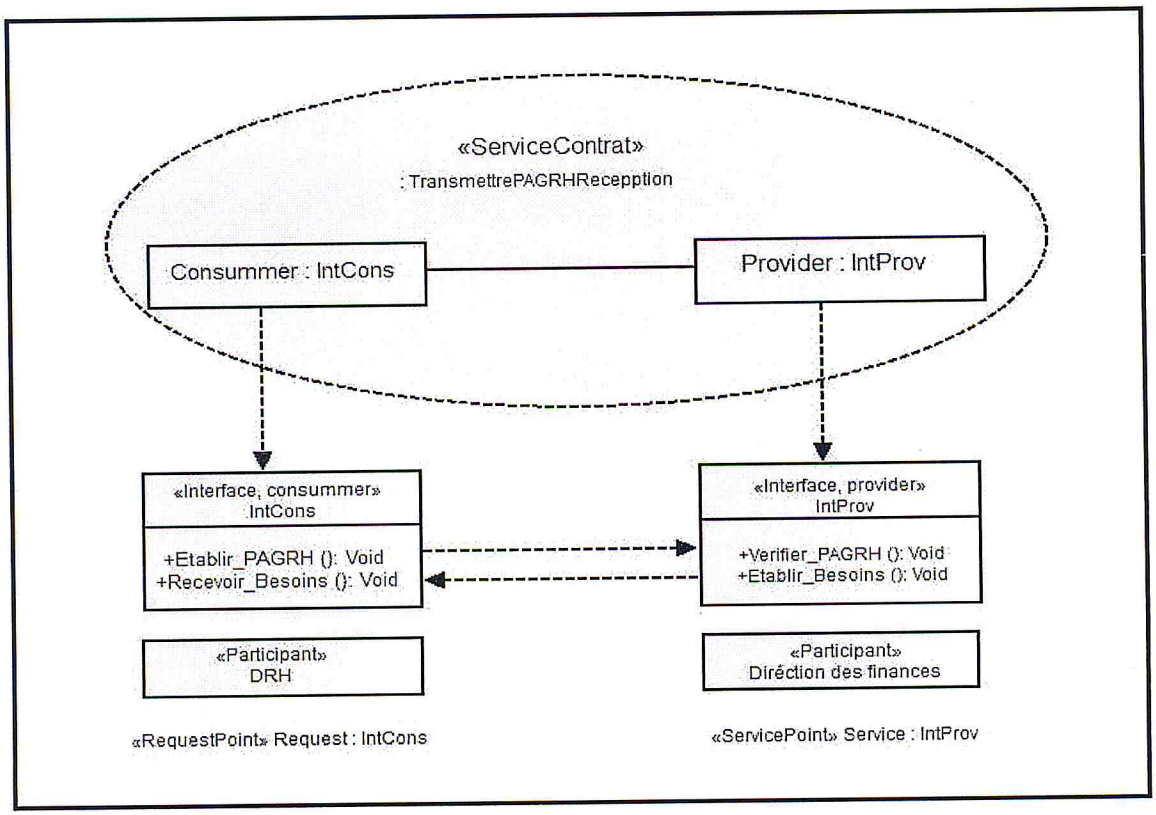


Figure V.9 Le contrat de service Transmettre PAGRH Réception

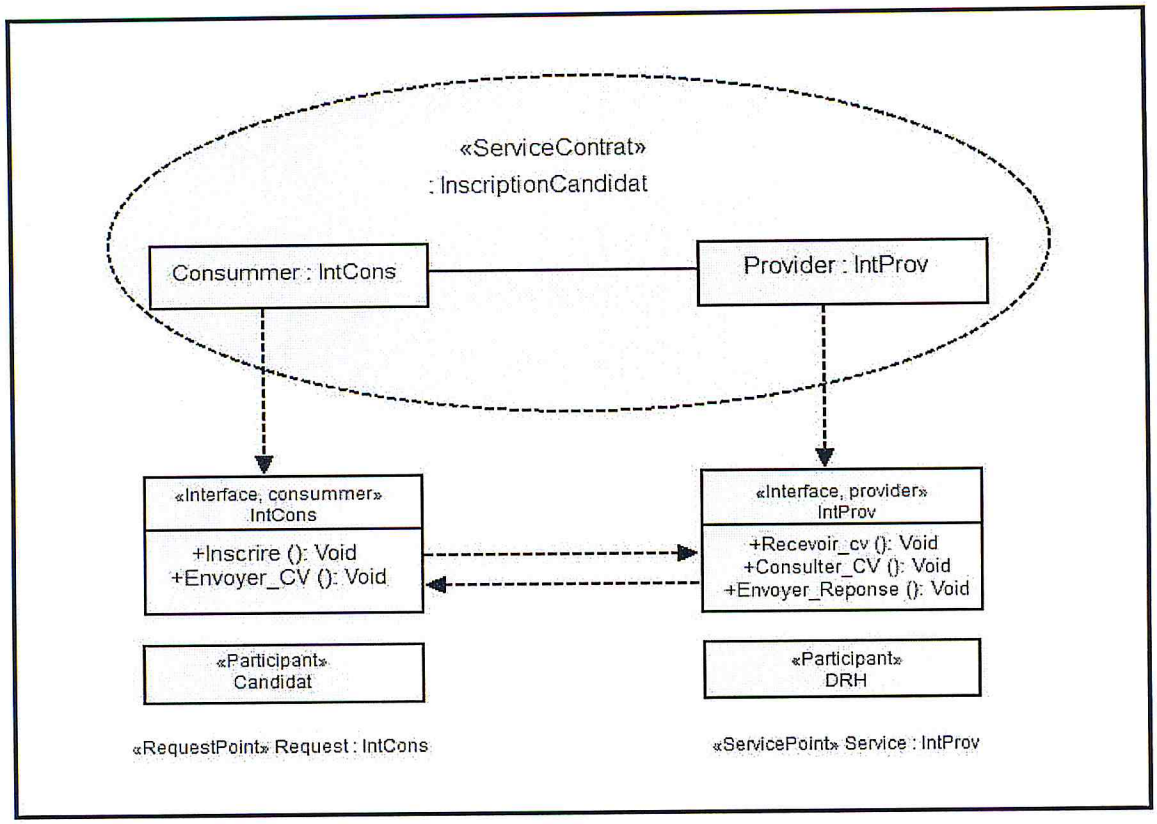


Figure V.10 Le contrat de service Inscription Candidat

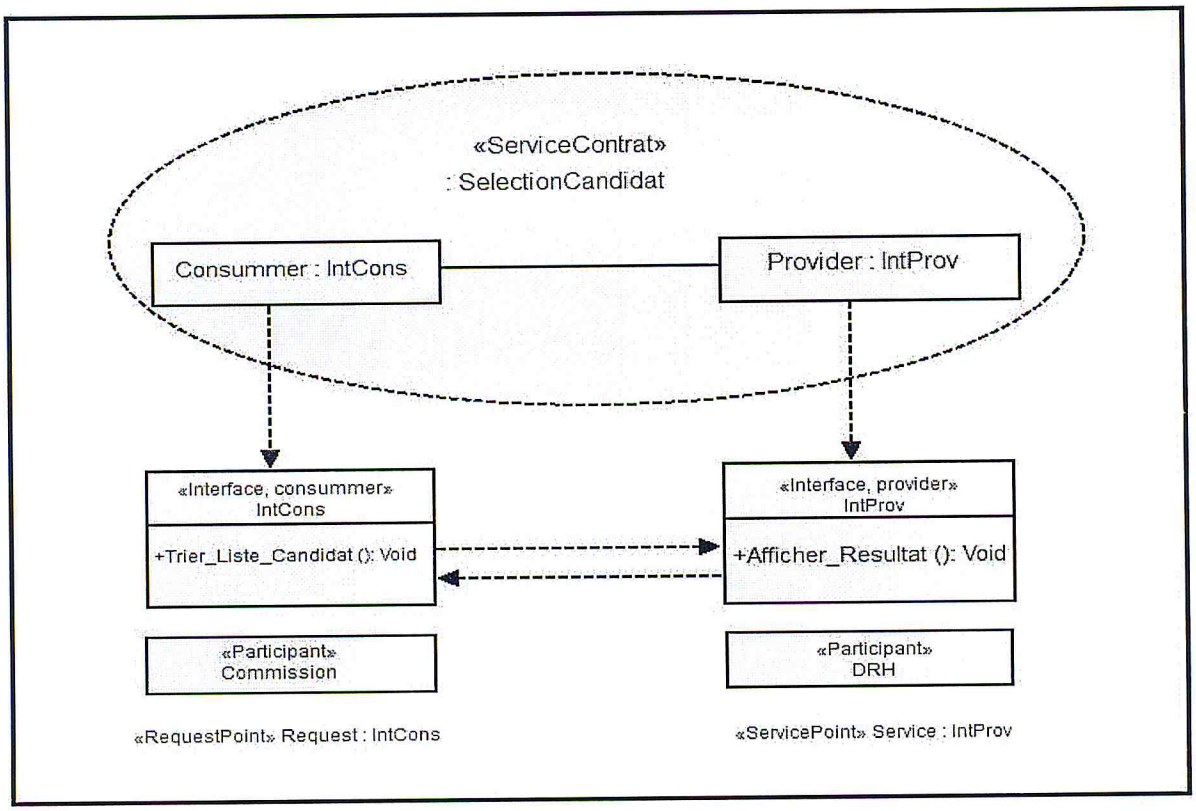


Figure V.11 Le contrat de service Sélection Candidat

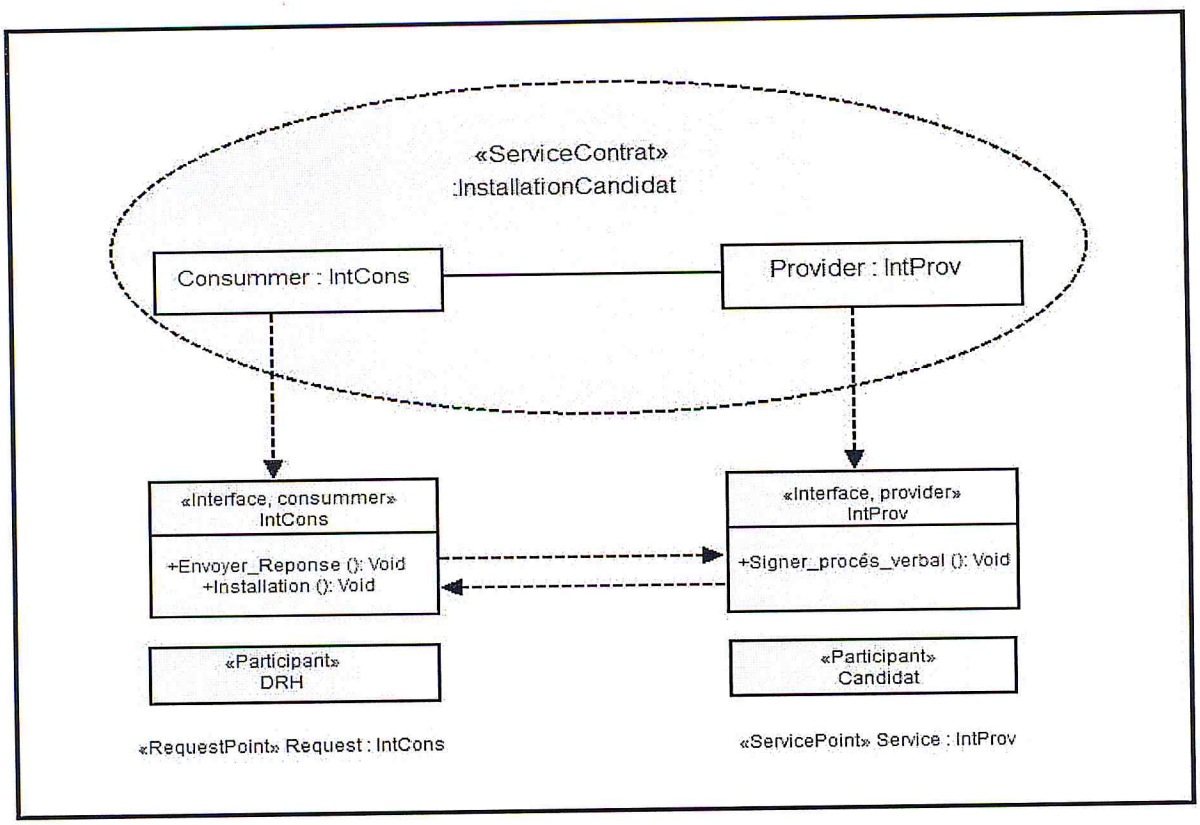


Figure V.12 Le contrat de service Installation Candidat

5. Phase 4 : réalisation

5.1. Les choix techniques

Dans cette section Nous présentons les différents choix techniques prisent dans le cadre de notre projet de fin d'étude, pour réaliser notre solution SOA.

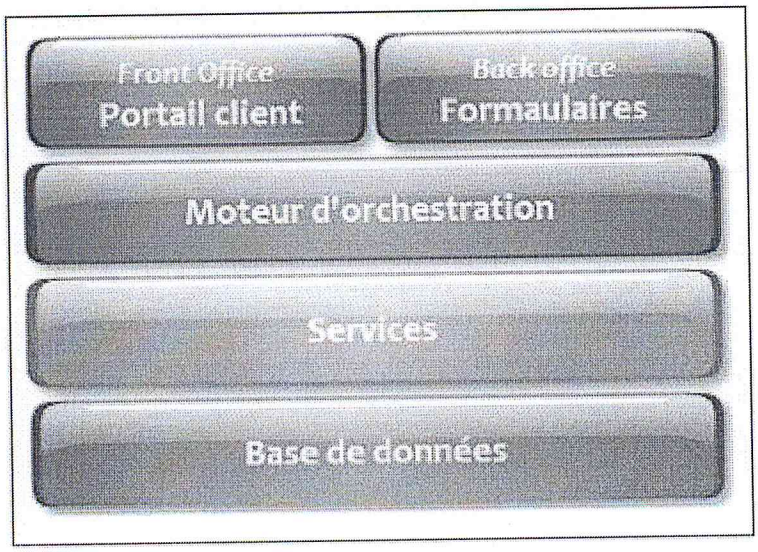


Figure V.13 Choix techniques

5.1.1. Système de gestion de base de données MySQL

est un système de gestion de base de données relationnelle basée sur le langage d'interrogation SQL (Structured Query laguage). C'est un des derniers logiciels open source de cette catégorie apparu sur le marché (si ce terme a du sens).développé a partir d'un autre SGBD portant le nom de mSQL, il possède de nombreuses qualité et notamment celle d'être portable, en ce sens qu'il s'exécute sur a peu prés tous les systèmes d'exploitation et tous les types de matériel (Etienne, 2005).

MySQL permet d'assurer :

- La définition et la manipulation des données ;
- La cohérence de données ;
- La sauvegarde et la restauration des données ;
- La gestion des accès concurrents.

5.1.2. Serveur http

Le projet Apache HTTP Server est un effort pour développer et maintenir un serveur HTTP open source pour les systèmes d'exploitation modernes, y compris UNIX et Windows

Le but de ce projet est de fournir un serveur sécurisé, efficace et extensible qui fournit des services HTTP en synchronisation avec les standards HTTP actuels.

Apache a été le serveur Web le plus populaire sur Internet depuis Avril 1996

Le November 2005 Netcraft Web Server Survey a trouvé que plus de 70% des sites sur l'Internet Web utilisent Apache, rendant ainsi plus largement utilisé que tous les autres serveurs Web combinées (Stephane, 2006).

5.1.3. Moteur d'orchestration

La solution **Intalio|BPMS** est architecturée autour des composants suivants :

- **Intalio|Designer**

est un outil de modélisation et de conception des processus métiers fondé sur la plate forme Eclipse. Il permet de créer des diagrammes de processus à l'aide de la notation BPMN et de compléter ces modèles en intégrant les Services Web et autres détails nécessaires à l'exécution des processus métiers. Cet outil traduit automatiquement les diagrammes BPMN en code source BPEL d'exécution des processus.

- **Intalio|Server**

comporte le moteur d'exécution des processus métiers en interprétant le langage BPEL. Il propose également l'ensemble des fonctionnalités d'administration.

- **Intalio|Workflow**

est basé sur la nouvelle extension BPEL4People il supporte l'exécution des activités humaines par le biais d'interfaces utilisateur AJAX générées automatiquement par le moteur XForms. Intalio|Workflow utilise Intalio|Server pour l'exécution des processus BPMN.

5.1.4. Formulaire

Qui dit processus d'entreprise dit aussi interface graphique orientée utilisateur final. On parle donc ici de formulaires web.

Outre l'édition des processus, Intalio offre aussi une interface graphique permettant de dessiner des XForms et d'appliquer un certain nombre de contraintes (champs obligatoires, types de champ...).

Dans le cadre de notre projet, nous préférons d'utiliser les formulaires AJAX munies d'un ensemble de composant UI 2 plus riche que les XForms, ainsi l'environnement de réalisation de ces formulaires est aussi intégré dans Intalio|Designer.

5.1.5. Portail client (candidat)

Un portail est une application web qui permet aux utilisateurs d'avoir un point d'accès aux applications composites de l'entreprise.

Pour la réalisation du portail client de notre application nous avons choisi le langage php.

- **PHP**

Le langage **PHP** (Personale Home Page) est un langage de programmation web côté serveur, ce qui veut dire que c'est le serveur qui va interpréter le code PHP (langage de scripts) et générer du code HTML qui pourra être interprété par le navigateur. Il permet d'ajouter des fonctionnalités de plus en plus complexe, d'avoir des sites dynamiques, de pouvoir gérer une administration de boutique en ligne, de modifier un blog, de créer des réseaux sociaux...

PHP connaît aujourd'hui un succès croissant en raison de sa facilité d'utilisation. Par le biais d'un ensemble de fichiers installés sur le compte Web, le PHP permet un interfaçage simple avec de nombreux Systèmes de Gestion ou de Bases de Données (Etienne, 2005).

5.1.6. Environnement de développement (WampServer)

WampServer est une plate-forme de développement Web sous Windows, permettant de faire fonctionner localement (sans connecter à un serveur externe) des scripts PHP.

WampServer est un environnement comprenant deux serveurs (Apache et MySQL), un interpréteur de script (PHP), ainsi que phpMyAdmin pour l'administration Web des bases MySQL.

La Figure V.14 résume les divers choix effectuées, et donne une vue globale de la plateforme de destination.

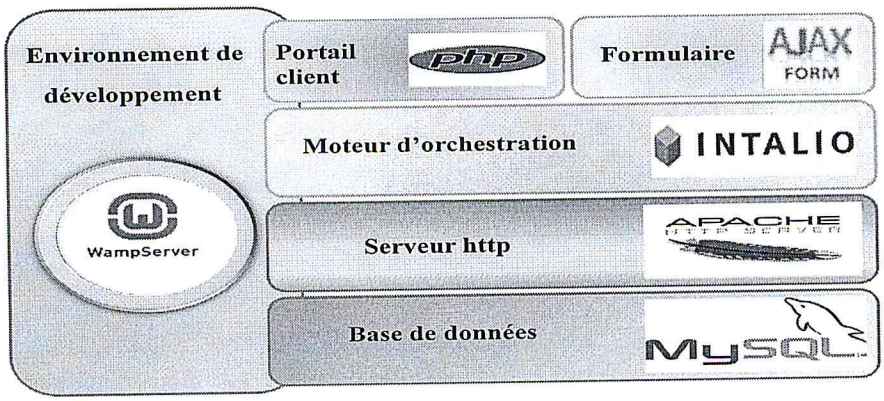


Figure V.14 une vue globale de la plateforme de destination

6. Conclusion

Dans ce chapitre nous avons présenté la démarche de développement de notre application, en appliquant la méthode SOA+d, suivant les étapes d'utilisation de ses aspects.

Dans le prochain chapitre nous allons présenter le système orientée service développé pour aider la commission à trier et sélectionner les bons candidats.

Chapitre VI

Réalisation et démonstration

1. Introduction

Une des étapes de la vie d'un projet, aussi importante que la conception, est l'implémentation. Cette étape constitue la phase d'achèvement et d'aboutissement du projet. Dans ce chapitre nous présentons un aperçu général sur la réalisation de notre solution SOA. Ensuite, nous illustrons les étapes de la réalisation et les différents vus de notre application implémentée.

2. Réalisation

Nous présentons dans cette section les étapes de réalisation de notre application strate par strate, de bas en haut.

2.1. La couche métier

Pour chaque machine logique (MLM) de la couche métier correspond un service CRUD, qui permet de créer, rechercher, mettre à jour et supprimer l'information dans les référentiels du système d'information, ces services sont réalisés avec PHP.

```
<h4> Afficher la liste des postes</h4>
<table style="margin-top:20px;"align="left;">
<tr>
<th align="center"> Poste </th> <th align="center"> Nombre Poste </th><th align="center"> Diplome </th><th align="center"> Specialite </th
<th align="center"> Description </th>
</tr>
<h4> Modifier un postes</h4>
<td>  <a href="poste_update.php?id=<?php echo $tab_id[$i] ?>" onClick="return confirm('Voulez vous Modifier ce centre?');">
            </td>
<h4> Supprimer un postes</h4>
            <td >
            <a href="poste_delete.php?id=<?php echo $tab_id[$i] ?>" onClick="return confirm('Voulez vous Supprimer cet actualité ?');">
      </td ></tr></table>
<h4> Ajouter un poste :</h4></br>
<form method="post" action="poste_ajout.php">
<p>
poste :              <input  id="nom_poste" name="nom_poste"  value ="<?php if
nombre poste : <input  id="nbr_poste" name="nbr_poste"  value ="<?php if(isset($_COOKIE['nbr_poste'])) echo $_COOKIE['nbr_poste']; ?>" typ
diplome :            <select name="diplome" id="diplome" value ="<?php if(isset($_COOKIE['dip
                                                    <option name="diplome" id="diplome" value ="<?php if(isset($_COOKIE['dipl
                                                    </select>
specialité :        <select name="specialite" id="specialite" value ="<?php if(isset($_COOKIE['specialit
                                                  <option name="specialite" id="specialite" value ="<?php if(isset($_COOKIE['specialit
                                                  </select>
Description :      <input  id="description" name="description"  value ="<?php if(isset($_COOKIE['description'])) echo 
      <input style="margin-left:40px;" type="submit" value="Ajouter " />
       </p>
    </form>
```

Figure VI.1 Service CRUD.

2.2. La couche réalisation

Les machines logiques (MLO) de la couche organisation sont réalisés par des services web, ces derniers seront par la suite orchestrés par le processus BPM, ou appelés directement par le portail client.

2.2.1. Les services web

Pour la création des services web nous pouvons utilisés deux approches qui sont :

- **Bottom- up**

Création du logique métier dans un langage donné, puis génération du WSDL et des classes nécessaires. Cette approche masque la partie définition WSDL, elle permet surtout de réutiliser du code existant.

- **Top-Bottom**

Définition d'un fichier WSDL puis génération de code vers un langage donné. Cette approche est indépendante de la plateforme de destination, mais nécessite une bonne maîtrise de la notion de langage de définition d'interface.

Les Services Web de notre application sont créés suivant l'approche Bottom-Up à l'aide de apache server et déployés sur WampServer.

- **Exemple d'un fichier WSDL**

Un fichier WSDL contient une description de tout ce qui est nécessaire à l'appel d'un Service Web :

- L'URL et l'espace de noms du service.
- Le type de Service Web.
- La liste des fonctions disponibles.
- Les arguments de chaque fonction.
- Le type de données de chaque argument.
- La valeur de retour de chaque fonction et le type de données de chaque valeur de retour.

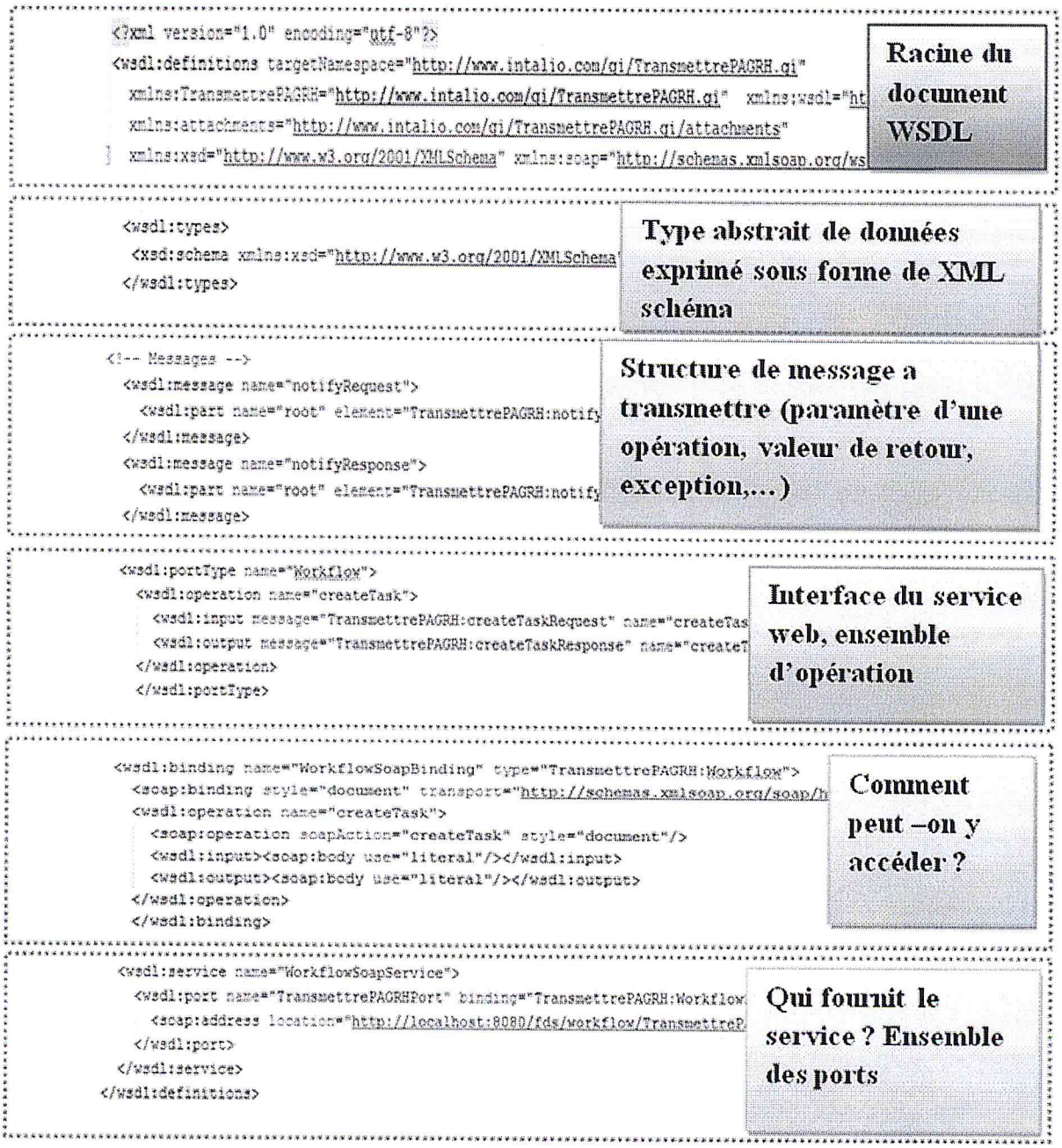


Figure VI.2 Exemple d'un fichier WSDL.

2.2.2. Les processus métier

Les modèles produits durant l'étape de conception adressent une réponse organisationnelle orientée « métier » et de gestion des individus, alors il faut les transposés en modèles de processus exécutables représentés avec la notation BPMN en utilisant l'outil choisi « Intalio|Designer ».

Ces modèles sont automatiquement traduits en langage d'exécution des processus métiers BPEL et exécutés directement sur un système de gestion des processus métiers « Intalio|BPMS ».

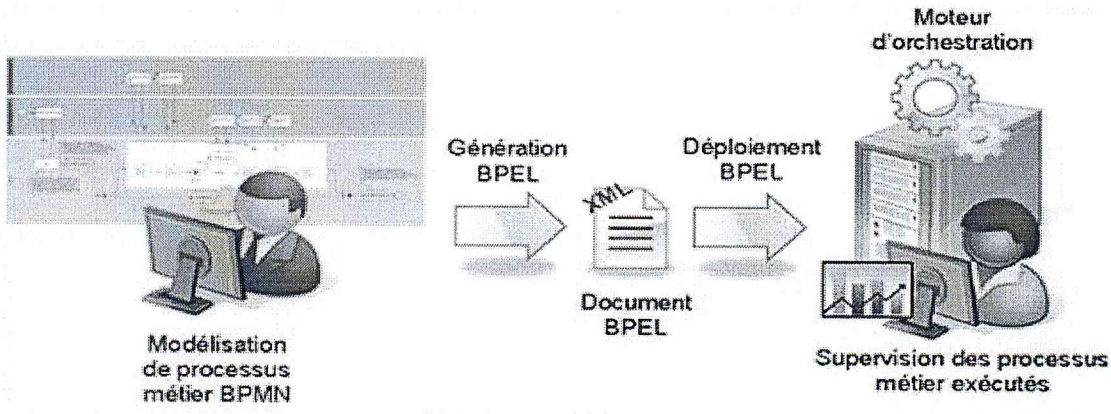


Figure VI.3 Les étapes de transposition des processus métiers.

La mise en œuvre des processus métier avec Intalio|BPMS est réalisée en plusieurs étapes :

- La Figure suivante illustre la représentation des intervenants des processus dans sa définition graphique.

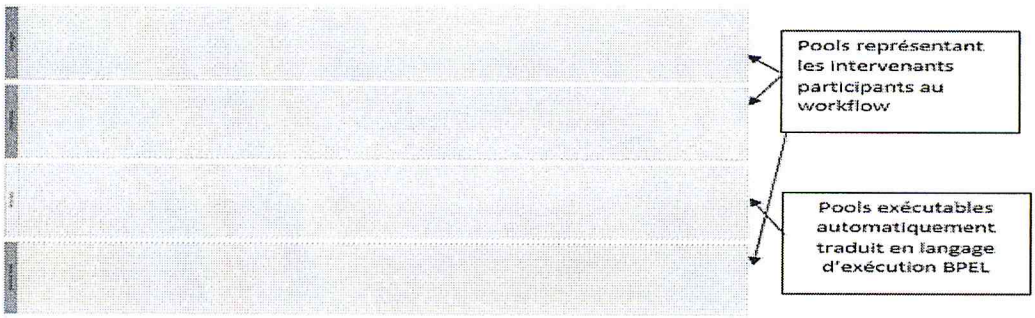


Figure VI.4 Définitions des rôles

- La définition du flux d'activités du processus principal.

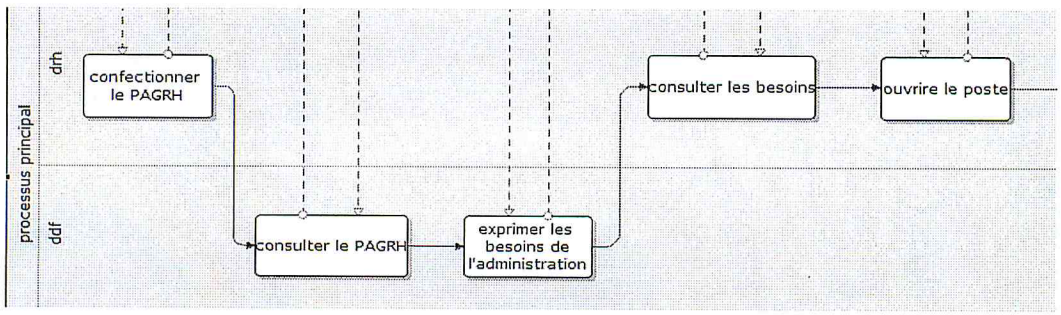


Figure VI.5 Partie du processus principale

- Intégration des formulaires utilisés dans les étapes du Workflow, ces formulaires représentent les tâches à réaliser ou les notifications provenant de l'exécution du processus.

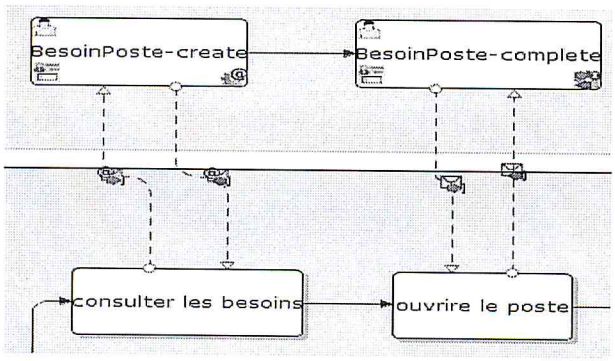


Figure VI.6 Intégration des formulaires

- L'intégration des Services Web impliqués dans l'exécution du processus métier, l'interface de description de Service Web WSDL est utilisée comme moyen de référencement d'intégration.

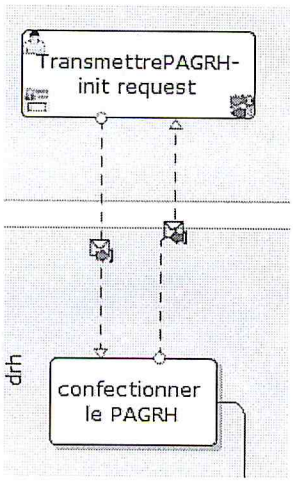


Figure VI.7 Intégration des Services Web

- Le déploiement des diagrammes BPD sur Intalio|Server, un diagramme BPD déployé sur le serveur devient lui-même un service Web. Il est éventuellement appelé depuis Intalio|Workflow via un navigateur web.

2.3. Couche présentation

2.3.1. Les formulaires

Ces formulaires représentent les tâches à réaliser ou les notifications provenant dans l'exécution du processus.

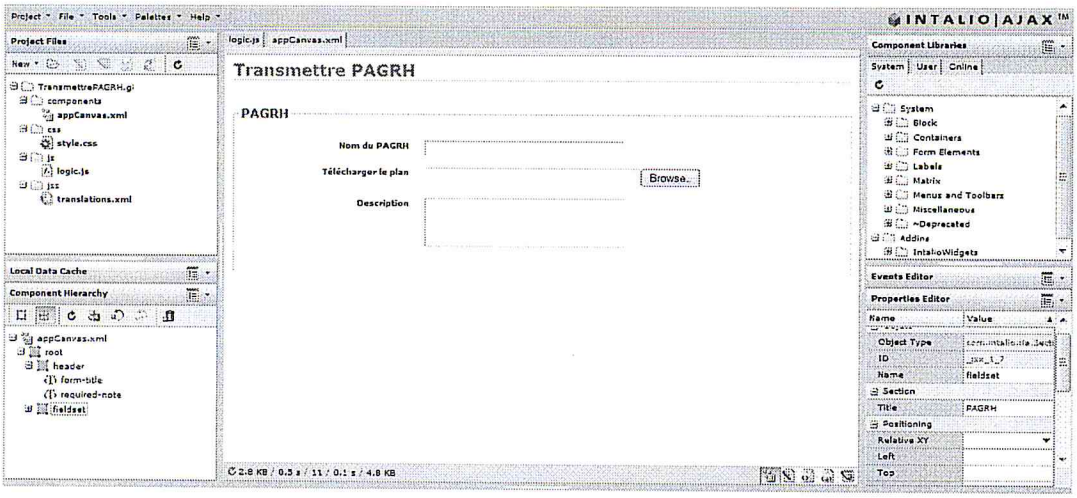


Figure VI.8 Formulaire AJAX « Transmettre PAGRH ».

3. Les interfaces

Dans ce qui suit nous allons présenter les différentes vues d'exécution de notre application.

➢ **Accueil :**

Le lancement de l'application conduit l'utilisateur à l'interface suivante :

Figure VI.9 La fenêtre Accueil

➢ Formulaire d'inscription du candidat

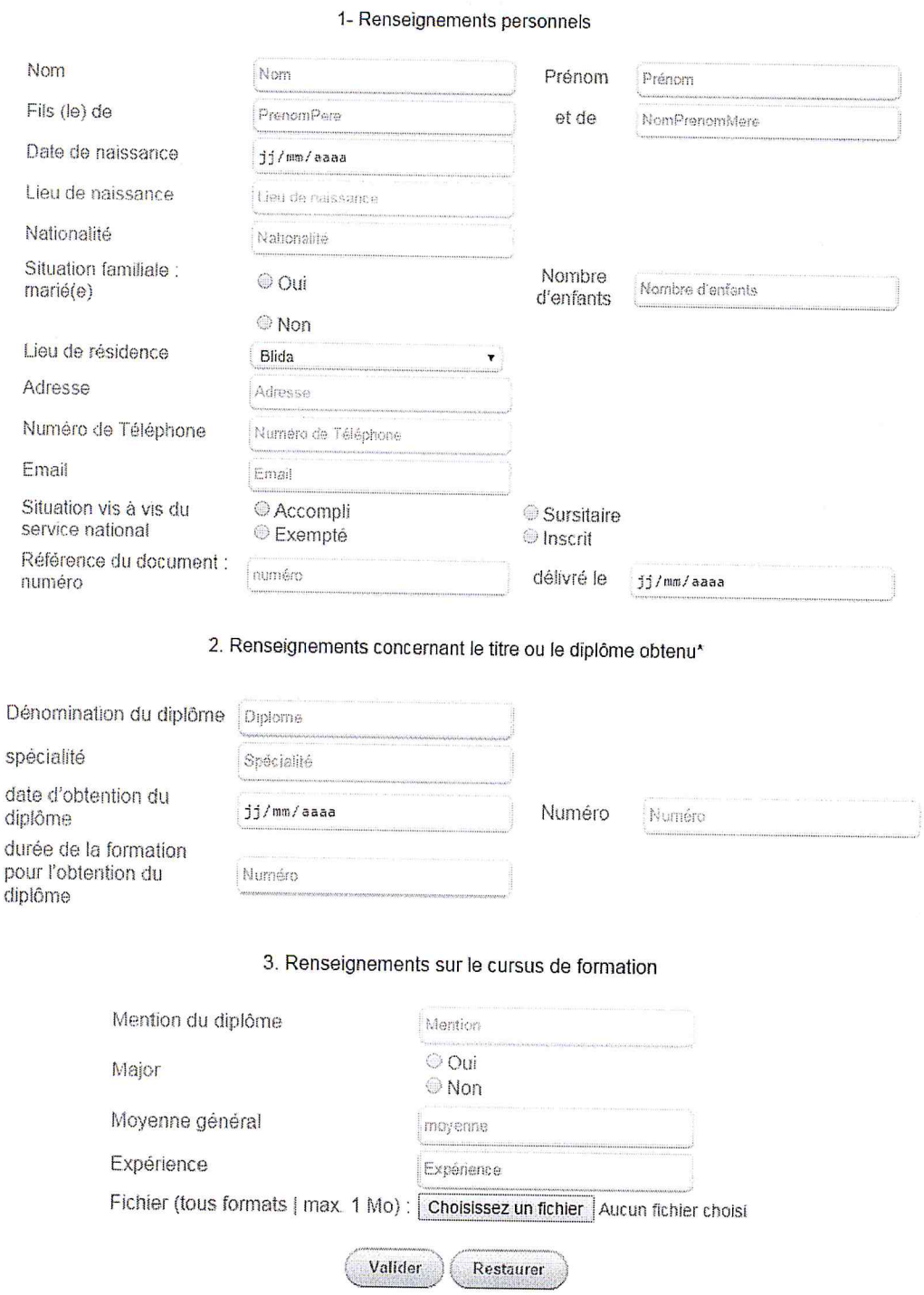


Figure VI.10 Formulaire d'inscription du candidat

- **Service CRUD Service CRUD**

L'acronyme informatique anglais CRUD (pour Create, Read, Update, Delete) désigne les quatre opérations de base pour la persistance des données, en particulier le stockage d'informations en base de données, dans notre application les services CRUD sont appliqué sur tous les table persistante de notre base de donnée.

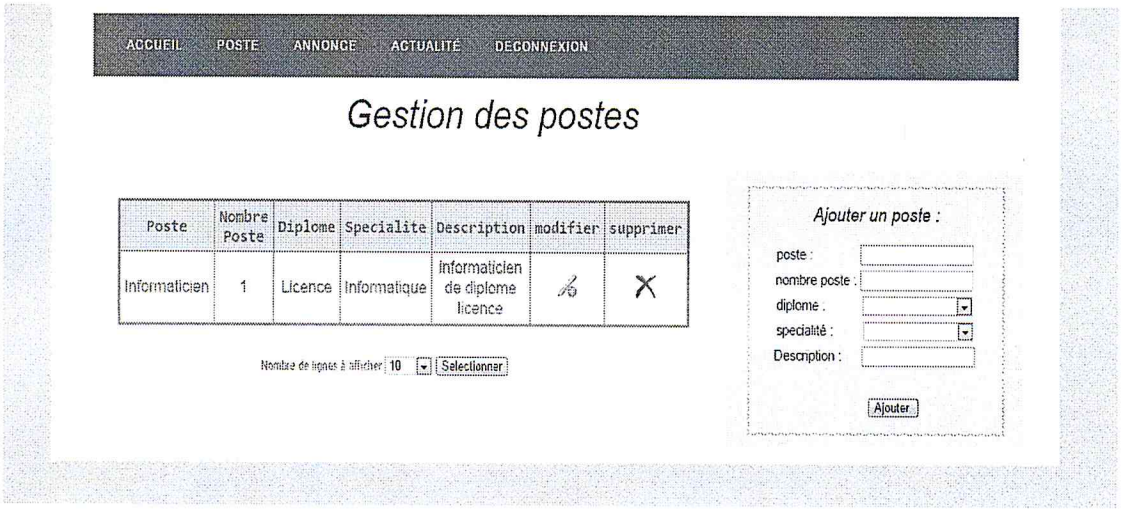


Figure VI.11 services CRUD

- **La liste des candidats triée**

La figure suivante montre la liste des candidats triée d'après l'utilisation de la méthode de classement AHP.

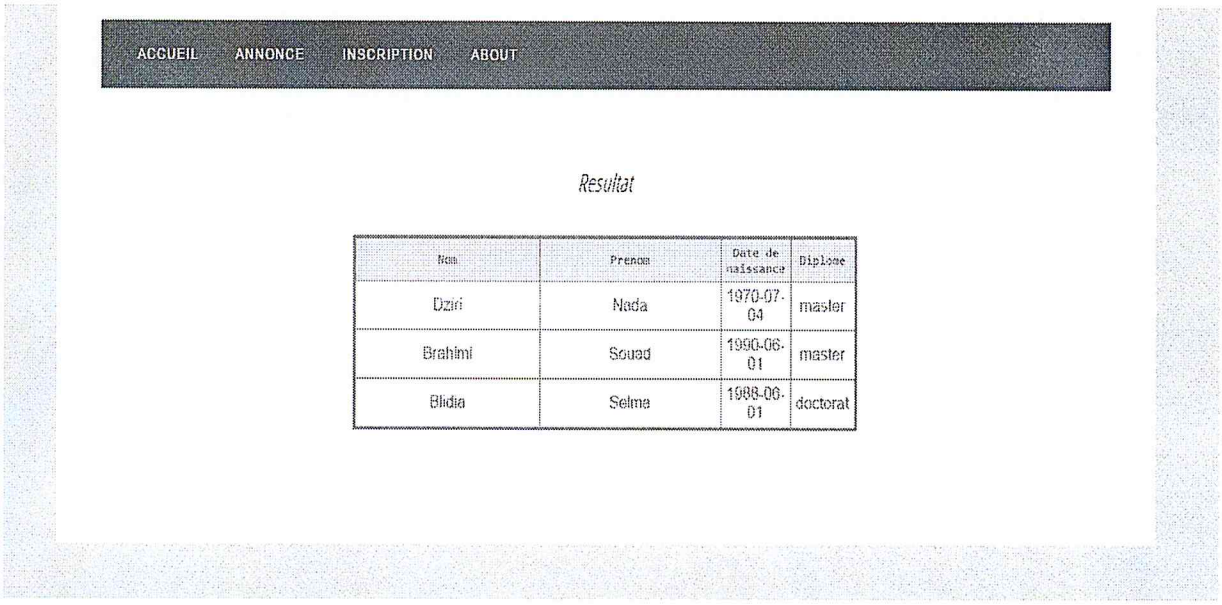


Figure VI.12 Liste des candidats triée.

Les formulaires

L'utilisateur peut procéder recrutement des candidats en accédant à son compte via l'interface Intalio|Workflow :

Figure VI.13 Intalio|Workflow «Authentification».

Après l'authentification, la fenêtre suivante apparaît au DRH :

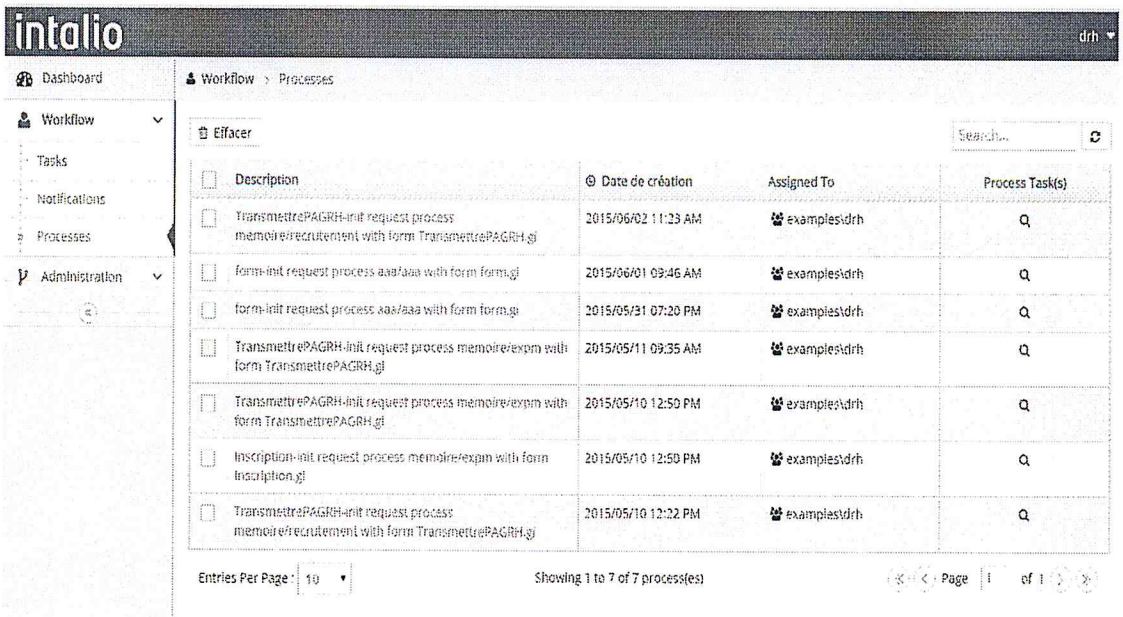


Figure VI.14 Intalio|Workflow «Liste des tâches».

Cette fenêtre présente les tâches associées à cet utilisateur, les messages qu'il a reçus ainsi que les processus qu'il peut déclencher. Un formulaire apparaît à l'utilisateur dès qu'il a choisi la tâche à réaliser. Il peut ainsi:

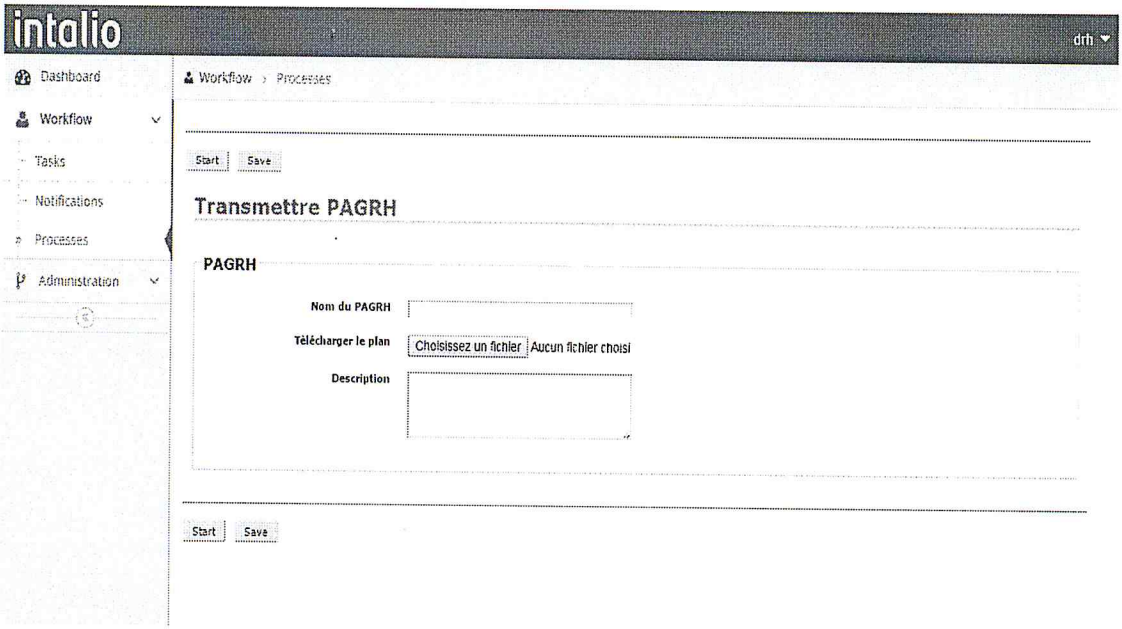


Figure VI.15 Formulaire de transmettre un PAGRH

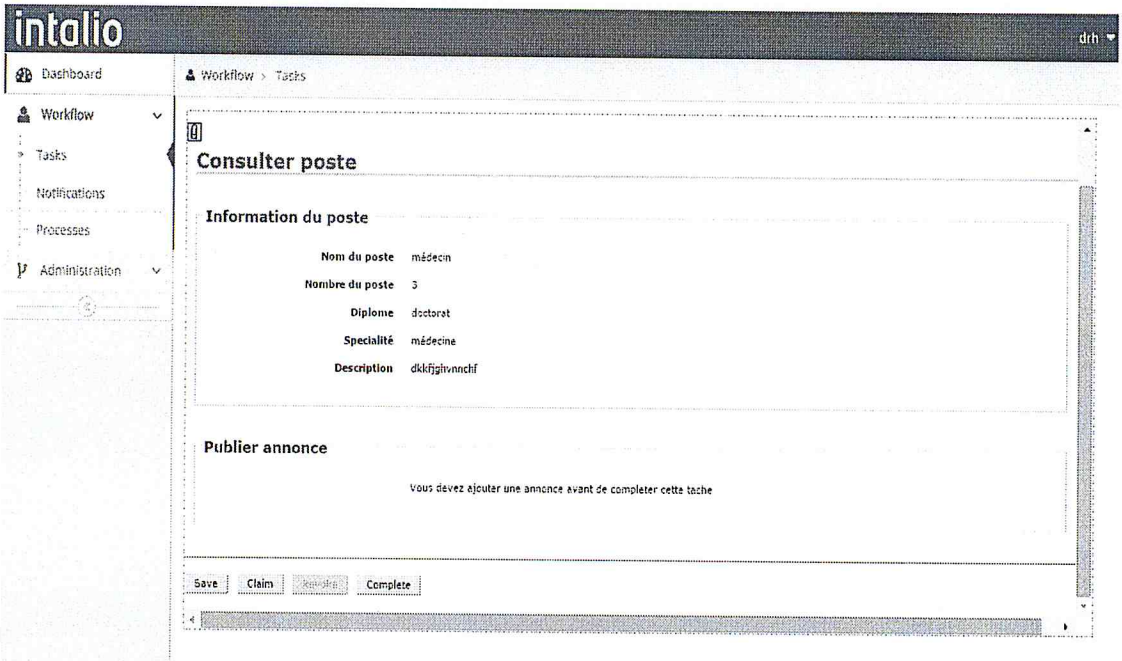


Figure VI.16 Consulter poste

Enregistrer le formulaire en choisissant le bouton « Save », la tâche reste alors en attente. Accomplir la tâche en choisissant le bouton «Complete ».

4. La console d'administration des processus :

La figure suivante illustre la console d'administration des instances des processus exécutées sur le serveur, qui est un espace réservé pour l'administrateur.

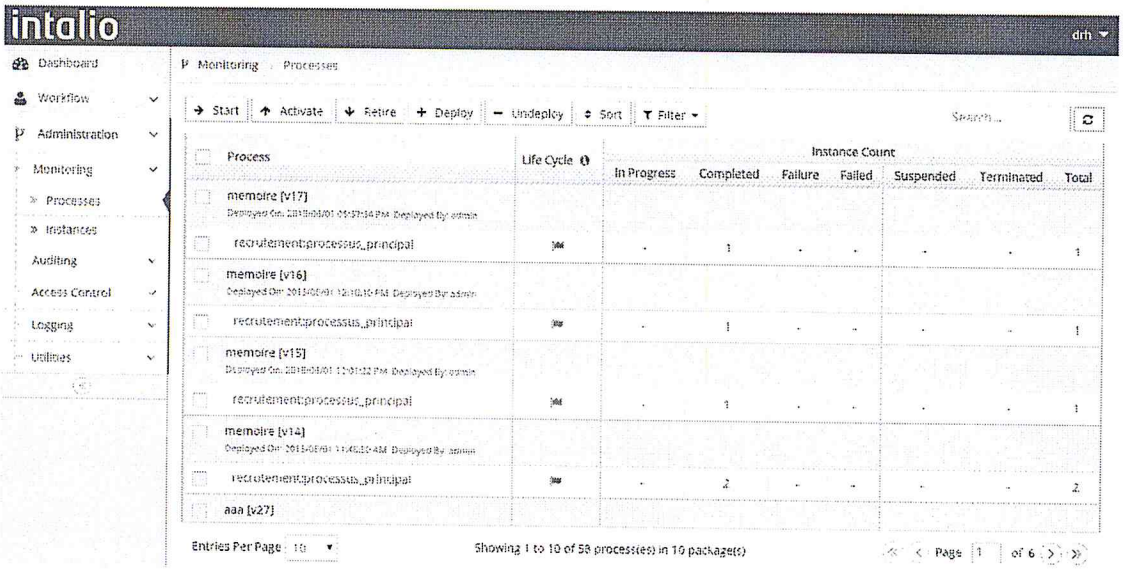


Figure VI.17 Intalio console.

5. Conclusion

L'exécution du processus informatisés reflète le déroulement du processus métier modélisé lors de la conception. Dans ce chapitre nous avons montré l'application de la méthode SOA +d, pour aider le décideur à prendre des décisions pertinentes.

Conclusion Générale

Conclusion

Le projet qui nous a été remis consistait à réfléchir à une architecture orientée service pour assurer l'aspect décisionnel dans le « processus de recrutement du CHU de Blida unité M'hamed Yazid ».

A cet effet, nous avons effectué un travail indispensable de recherche et d'analyse sur l'architecture orientée services et l'automatisation de processus métier.

Après l'étude du domaine de processus de recrutement, nous avons retenu une méthode de classement (la méthode AHP).

La présence d'un processus de développement formalisé, bien défini et bien géré est un facteur de réussite d'un projet. Pour cette raison nous avons suivi les phases de la méthode Soa+d, cette méthode vise à intégrer l'aspect décisionnel de l'architecture de type SOA au sein d'une entreprise.

Une fois que nous avons acquises des connaissances sur l'architecture SOA basée sur les services réutilisables, nous avons conçue, pour les besoins de l'étude, un système de processus de recrutement qui intègrent ces méthodes. Par la suite, nous l'avons implémenté sous une architecture SOA sous forme de web services.

Le début a été très difficile, ce qui est principalement dû à la multiplicité des concepts SOA, BPM, service web, orchestration…etc. des notations BPMN et BPEL et la nouvelle notation DMN. Cependant, ces difficultés nous ont permis d'acquérir un esprit de synthèse, ce qui est loin d'être négligeable.

Le projet nous a de plus permis d'approfondir nos connaissances sur la SOA, sur le BPM , ainsi que l'acquisition de connaissance sur des techniques récentes de modélisation de processus métiers (BPMN), et nous avons l'honneur d'être le 1^{ier} binôme en Algérie qui utilise le nouveau standard DMN .

En ce qui concerne la partie réalisation nous avons utilisé le SGBD mysql, l'environnement de développement Wamp Server et le serveur http apache server, pour orchestrer les services nous avons utilisé le moteur d'orchestration Intalio|BPMS.

Mais, comme tout projet le nôtre n'est pas exhaustif, il sera intéressant d'enrichir ce travail par :

- Implémenter d'autres méthodes d'aide a la décision Electre, Prométhée…etc. et comparer les résultats obtenus.
- La mise en œuvre d'un ESB (donner accès aux services web pour différents utilisateurs),

Bibliographie

Bibliographie

ABDELLATIF, Iheb. 2011. *VERS UNE DEMARCHE D'AIDE À LA DÉCISION POUR L'IDENTIFICATION DES SERVICES D'UNE ARCHITECTURE ORIENTÉE SERVICES.* [interpr.] M. Marc Paquet, M. Amin Chaâbane, M. Amar Ramudhin. UNIVERSITÉ DU QUÉBEC : s.n., 4 Février 2011.

Allen, p. 2007. *The service Oriented Process.* 2007.

An analysis of distributions of priority values from alternative comparison scales within AHP. **Beynon, Malcom James. 2002.** 2002, Operational Research .

Arradi, A, Annand, S et Kulkarni, n. 2006. *SOAF Architecturel Fromwork for service and Realisation. Service computing.* s.l. : SCC'06. IEEE International conference, 2006.

Arsanjani, A, et al. 2008. *SOMA: A methode for developping Service Oriented solution.* 2008.

BEA Systems,. 2005. *Méthodologie SOA en six domaines.* 2005.

Boumahdi, F et Rahal, E. 2015. *Intégration des aspects décisionnels dans les architectures SOA.* s.l. : ESI, 2015.

Brian Elvesæter, Dima Panfilenko , Sven Jacobi , Christian Hahn. 2010. *Aligning Business and IT Models in Service-Oriented Architectures using BPMN and SoaML.* 2010.

Caillet, Renaud. 2003. *Etude et comparaison des méthodes existantes en vue d'une application en analyse de cycle de vie.* Aout 2003.

CBDI-SAE. 2007. Meta Model. *Soa Versien 2.* [En ligne] 2007. [Citation : 20 Fevrier 2015.] http://www.cbdiforum.com/public/metamodelV2.

COUTURIER, Jean-François. 2011. *ÉLABORATION ET EXPÉRIMENTATION D'UNE MÉTHODOLOGIE AGILE PERMETTANT LA MIGRATION VERS UNE ARCHITECTURE ORIENTÉE SERVICES EN PME À L'AIDE D'OPENUP.* [interpr.] Roger Champagne, Ghizlane El Boussaidi, M. François Coallier. UNIVERSITÉ DU QUÉBEC : s.n., 7 Octobre 2011.

De Castro, V., Marcos, E., Vara, J. M. 2011. *Applying cim-to-pim model transformations for the service-oriented development of information systems.* s.l. : Information and Software Technology, 2011.

Determining What to Benchmark: An Analytic Hierarchy Process Approach. **Partovi, Fariborz Y. 1994.** 1994, International Journal of Operations & Production Management.

DMN. 2014. Decision modeling notationTechnical report . *OMG.* [En ligne] 2014. http ://www.omg.org/spec/DMN/1.0/Beta1/PDF.

Erl, Thomas et Upper, Saddle River. 2005. *Service Oriented Architecture : Concepts Technology and designe.* ISBN : Prentice Hall PTR, 2005.

Erl, Thomas. 2008. *Service-Oriented Architecture: Conceptss, Technology, and Design.* . s.l. : Prentice Hall, 2008.

—. 2006. *Service-Oriented Archetecture, Concept Technology and designe* . Indiana : printice Hall, 2006.

Ernest Forman, Mary Ann Selly , Expert Choice Inc. 2001. *Decision By Objectives (How to convince others that you are right)* . 2001.

Etienne, vandeput. 2005. *Developper une application PHP et Mysql.* 2005.

JENN, OLIVIER. 2008. *MÉTHODES ET MODELES POUR LA SÉLECTION DES FOURNISSEURS.* UNIVERSITÉ DU QUÉBEC A MONTREAL, Service des bibliothèques : s.n., Decembre 2008.

Jones, S et Morris, M. 2005. *A methodology for service architectures.* s.l. : Capgemini UK plc, 2005.

LEMRABET, Youness. 2012. *Proposition d'une méthode de spécification d'une architecture orientée services dirigée par le métier dans le cadre d'une collaboration inter-organisationnelle.* ECOLE CENTRALE DE LILLE , Lille, France : s.n., 7 Juin 2012.