

الجمهورية الجزائرية الديمقراطية الشعبية

République Algérienne Démocratique et Populaire

Ministère de l'enseignement supérieur et de la recherche scientifique

Université de 8 Mai 1945 – Guelma -

Faculté des Mathématiques, d'Informatique et des Sciences de la matière

Département d'Informatique



Mémoire de Fin d'études Master

Filière : Informatique

Option : Sciences et Technologies de l'Information et de la
Communication (STIC)

Thème :

**Réingénierie d'une plateforme d'apprentissage existante vers
une plateforme orientée services**

Encadré Par : Mr SERIDI Ali

Présenté par :
Hamdi Ahmed

Septembre 2020



Je tiens tout d'abord à remercier notre Dieu, qui m'a donné la force et la patience d'accomplir ce Modeste travail.

Je remercie Monsieur Seridi Ali, mon Encadreur pour sa disponibilité, sa patience, ses précieux conseils et ses remarques constructives. Ainsi que pour son aide précieuse en termes de documentation et de disponibilité.

Je désire aussi remercier les enseignants de l'université 08 MAI 1945, de Guelma (département d'informatique), qui m'ont fourni les outils nécessaires à la réussite de mes études universitaires.

Je tiens à remercier les membres du jury qui m'ont fait l'honneur d'évaluer mon travail.

Finalement, un grand merci chaleureux à mes parents, mes sœurs, mon frère, ma femme, mes petits garçons Mohamed Nazim et Iyad et tous les collègues de travail et en plus, Monsieur le Directeur Bourahdoun Salim pour sa compréhension et son assistance.

Merci à tous mes amis d'étude de ma promotion, je remercie également tous ceux qui m'ont aidé à la réalisation de ce projet et en particulier Mlle Grini Ahlem pour son aide.

Résumé

La réingénierie logicielle est une technique qui permet de réécrire un logiciel existant dans une nouvelle technologie d'implémentation afin d'assurer une meilleure maintenabilité et gagner en performance et en évolutivité sans perdre au niveau fonctionnel.

Parmi les nouvelles technologies d'implémentation qui est en plein expansion dans le domaine des applications web, on trouve l'architecture orientée service (SOA) et plus précisément les services web.

Nous voudrions à travers ce PFE, appliquer cette technique de réingénierie pour faire évoluer les plateformes d'apprentissage en ligne vers une architecture orientée service (SOA), ce qui permet une meilleure flexibilité, réutilisabilité, et maintenabilité pour ce type d'application. Plusieurs techniques existent pour atteindre ce but, Top-Down ; Bottom-Up ; Meet in the middle. Chacune peut être considérée plus adaptée à un domaine mieux qu'à un autre. Dans notre cas nous avons utilisé l'approche Top-Down qui permet d'avoir une vue globale du projet final et de donner une estimation rapide, bien qu'approximative, de sa complexité.

L'avantage de notre travail consiste à permettre d'identifier les parties des plateformes d'apprentissage qui peuvent être réutilisables à distance, ce qui permet de gagner en coût et en temps de développement.

Mots Clés : Réingénierie logiciel, plateforme d'apprentissage en ligne, SOA, Service web, BPMN 2.0

Table des Matières

Introduction Générale.....	1
----------------------------	---

Chapitre 1 : Réingénierie et migration vers SOA

1. Introduction	3
I. L'Architecture Orientée Service (SOA).....	3
1. Définition de la SOA.....	3
2. Les avantages.....	3
3. Limites de l'architecture SOA.....	3
4. Les caractéristiques d'une SOA	5
5. Les types des technologies SOA	6
6. Caractéristiques des services	6
7. Les services Web	8
8. Les standards du Web Service.....	8
A. Le Langage de Définition de Web Service (WSDL)	8
B. SOAP (Simple Object Access Control)	9
C. UDDI (Universal Description, Discovery and Integration)	9
II. Réingénierie	9
1. Définition.....	9
2. Les différents Types de la réingénierie.....	9
2.1. La réingénierie des systèmes d'information.....	10
2.2. La réingénierie des processus.....	10
2.3. La réingénierie des logiciels.....	10
3. Réingénierie et migration vers SOA.....	10
3.1. Bénéfices de la migration	10
3.2. Difficultés de la migration	11
3.3. Différentes approches de migration.....	11
3.3.1. Approche Top Down	12

3.3.2. Approche Bottom Up.....	12
3.3.3. Approche Outside In.....	13
3.3.4. Approche Middle Out.....	13
4. Conclusion.....	14

Chapitre 2 : les systèmes E-Learning à base de SOA

1. Introduction.....	15
2. Les systèmes E-Learning.....	15
1.1. Définition.....	15
1.2. Les avantages de E-Learning.....	15
1.3. Les inconvénients.....	16
1.4. Les composants d'une plateforme E-Learning.....	16
1.5. Les exemples sur les plateformes E-Learning.....	16
1.6. Les systèmes de gestion d'E-Learning.....	17
3. SOA et le E-Learning.....	18
3.1. Avantage d'intégration de la SOA dans le e-learning.....	19
3.2. Travaux dans le contexte.....	20
4. Bilan des travaux similaires.....	26
5. Conclusion.....	27

Chapitre 3 : modélisation des processus métiers

1. Introduction.....	28
2. Les processus.....	28
2.1. Définition.....	28
2.2. Les types de processus.....	29
2.2.1. Les processus de support.....	29
2.2.2. Les processus de management.....	29
2.2.3. Les processus de réalisation (métier).....	29
2.3. L'interaction entre les trois types de processus.....	30
2.4. Les différentes méthodes de modélisation.....	30
2.4.1. Business Process Management (BPM).....	30
2.4.2. Process Definition Language (XML).....	31

243. Business Process Execution Language (BPEL).....	31
3. Business Process Modeling Notation (BPMN).....	31
3.1. Objectifs de BPMN.....	32
3.2. Les principaux éléments graphiques de BPMN.....	32
3.3. Les outils de modélisation des processus.....	34
4. DE L'APPROCHE BPM AU STANDARD BPMN 2.0.....	34
4.1. Cycle de vie d'un processus.....	34
5. BPM et SOA : concurrence ou synergie ?.....	36
6. Conclusion.....	38

Chapitre 4 : Conception

I. Introduction.....	39
II. Objectif du projet.....	39
III. Architecture générale de notre système.....	40
<i>Phase 1 : Analyse des plateformes E-learning classique et extraction des fonctionnalités indépendantes.....</i>	40
1. Fonctionnalité gestion de bibliothèque.....	41
2. Fonctionnalité test niveau.....	42
3. Fonctionnalité gestion des cours.....	42
4. Fonctionnalité collaboration.....	42
5. Fonctionnalité aide au tutorat.....	42
6. Fonctionnalité de regroupement des apprenants.....	43
<i>Phase 2 : Modélisation des fonctionnalités de base.....</i>	43
<i>Phase 3 : exportation vers XPD (XML Process Definition Language).....</i>	47
<i>Phase 4 : Automatisation de l'identification des services web.....</i>	47
1. Le prétraitement des diagrammes BPMN.....	47
2. L'analyse des diagrammes BPMN.....	48
3. Règles d'identification des services.....	48
<i>Phase 5 : Validation des services candidats.....</i>	52

IV.	Conclusion	52
-----	------------------	----

Chapitre 5 : Implémentation

1.	Introduction	53
2.	Les outils de développement	53
	2.1. Java	53
	2.2. La plateforme Eclipse.....	53
	2.3. Bizagi Modeler.....	53
	2.4. XPDL (XML Processing Description Language).....	53
	2.5. JDOM (Java Document Object Model).....	54
3.	L'Architecture globale de notre plateforme	54
4.	Présentation des interfaces.....	55
	4.1. Page d'accueil.....	55
	4.2. La barre d'outils	55
	4.2.1. Charger un fichier XPDL.....	56
	4.2.1.1. La phase prétraitement.....	56
	4.2.2. Afficher le diagramme de BPMN 2.0 graphiquement.....	57
	4.2.3. Appliquer toutes les règles et identifier les services	57
	4.2.4. Appliquer une règle et identifier les services	59
	4.2.5. Ajouter une règle.....	60
	4.2.6. Gestionnaire des règles	61
	4.2.7. Le guide d'utilisation.....	62
	4.2.8. Exporter la Liste des services candidats	62
5.	Conclusion.....	63
	Conclusion générale et perspectives	64
	Références bibliographiques	65

Liste des Figures

Figure 1.1 : Les principales limites de SOA	05
Figure 1.2 : Les caractéristiques de service.....	07
Figure 1.3 : Architecture des web services.....	08
Figure 1.4 : Différentes approches de migration.....	12
Figure 2.1 : Architecture du système E-learning.....	20
Figure 2.2 : Architecture du système E-learning.....	21
Figure 2.3 : Architecture du système E-learning.....	22
Figure 2.4 : Architecture du système E-learning.....	23
Figure 2.5 : Architecture du système E-learning.....	24
Figure 2.6 : Architecture du système E-learning.....	25
Figure 2.7 : Architecture du système E-learning.....	26
Figure 3.1 : Représentation graphique d'un processus.....	29
Figure 3.2 : Les types de processus.....	29
Figure 3.3 : L'interaction entre les trois types de processus	30
Figure 3.4 – Éléments de base de la notation BPMN.....	32
Figure 3.5 : Éléments du langage BPMN.....	33
Figure 3.6 : Les outils de modélisation des processus	34
Figure 3.7 : Représentation commune d'un cycle de vie des procédures métier.....	35
Figure 3.8 : BPM – cycle de vie des processus métier.....	35
Figure 3.9 : Cycle de vie des procédures métier	35
Figure 3.10 : BPM - cycle de vie des processus métier	36

Figure 4.1 : les étapes principales pour la réingénierie d'une plateforme e-learning existante vers une plateforme orientée service	40
Figure 4.2 : Fonctionnalités de bases choisies après l'analyse.....	41
Figure 4.3 : Représentation BPMN Fonctionnalité bibliothèque à distance	44
Figure 4.4 : Représentation BPMN fonctionnalités aide au tutorat	44
Figure 4.5 : Représentation BPMN fonctionnalité de base Collaboration	45
Figure 4.6 : Représentation BPMN fonctionnalité de base Cours	45
Figure 4.7 : Représentation BPMN fonctionnalité Test niveau	46
Figure 4.8 : Représentation BPMN fonctionnalité regroupement des apprenants	46
Figure 4.9 : Processus d'Automatisation de d'identification des services	47
Figure 4.10 : Diagramme illustratif de la règle 1	49
Figure 4.11 : Exemple illustratif de la règle 01	49
Figure 4.12 : Exemple illustratif de la règle 02	50
Figure 4.13 : Diagramme illustratif de la règle 3	51
Figure 4.14 : Exemple illustratif de la règle 03	51
Figure 5.1 : l'architecture générale de notre Framework	54
Figure 5.2 : Page d'accueil de notre plateforme.....	55
Figure 5.3 : charger un fichier XPD L	56
Figure 5.4 : message d'affichage dans La phase prétraitement.....	56
Figure 5.5 : Affichage des diagrammes BPMN 2.0 sous forme graphique.....	57
Figure 5.6 : Boite de dialogue affiche le nombre des services candidats identifié	58
Figure 5.7 : affichage de l'option appliquer toutes les règles et identifier les services.....	58
Figure 5.8 : message d'alerte pour l'inexistence d'un fichier XPD L.....	58
Figure 5.9 : fenêtre de sélection de règle particulière	59
Figure 5.10 : Affichage du nombre des services identifiés avec l'application de la règle sélectionnée	59
Figure 5.11 : Affichage des services identifiés avec la règle 3	59

Figure 5.12 : Interface de l'ajout d'une règle.....	60
Figure 5.13 : exemple d'une règle ajoutée	60
Figure 5.14 : Affichage du gestionnaire des règles	61
Figure 5.15 : Guide d'utilisation	62
Figure 5.16 : Affichage des résultats d'identification dans un fichier texte	62

Liste des Tableaux

Tableau 1.1 : Les caractéristiques d'une SOA	05
Tableau 2.1 : Les différents composants des E-Learning	16
Tableau 3.1 : comparaison entre SOA et BPM	38

Liste des abréviations

SOA : **S**ervice **O**riented **A**rchitecture.

SI : **S**ystème d'**I**nformation.

LMS : **L**earning **M**anagement **S**ystem

LCMS : **L**earning **C**ontent **M**anagement **S**ystem

VCS : **V**irtual **C**lassroom **S**ystem

ITS : **I**ntelligent **T**utoring **S**ystems

IDE : **I**ntegrated **D**evelopment **E**nvironment.

JAR : **J**ava **A**rchive.

WS : **W**eb **S**ervice

BPM : **B**usiness **P**rocess **M**ethod

WSDL : **W**eb **S**ervice **D**escription **L**anguage

BPMN : **B**usiness **P**rocess **M**odeling **N**otion

XML : **E**xtensible **M**arkup **L**anguage

SOAP : **S**imple **O**bject **A**ccess **P**rotocol

UDDI : **U**niversal **D**escription, **D**iscovery and **I**ntegration

IBM : **I**nternational **B**usiness **M**achines

BPEL : **B**usiness **P**rocess **E**xecution **L**anguage

JEE : **J**ava **E**ntreprise **E**dition

OMG : **O**bject **M**anagement **G**roup

ISO : **I**nternational **O**rganization for **S**tandardization

XPDL : **X**ml **P**rocessig **D**escription **L**angage

JDOM : **J**ava **M**odèle d'**O**bjets de **D**ocument

BD : **B**ase de **D**onnées

Introduction générale :

Le monde moderne dans lequel nous vivons a connu un développement accéléré, reflétant positivement le développement de l'apprentissage en ligne, qui est devenu une alternative à l'apprentissage classique. Le développement de l'apprentissage en ligne est clairement lié au développement de nouvelles technologies et outils numériques, qui ont conduit à une large diffusion des plateformes e-learning. Avec l'accroissement énorme du nombre d'abonnés dans ces plateformes que ce soit des apprenants, des enseignants, des tuteurs, Il était nécessaire de moderniser ces plateformes et de les rendre plus flexibles afin de gagner du temps et réduire les coûts, ce qui a conduit au choix de la réingénierie de ces plateformes.

La réingénierie dans notre cas, consiste à l'amélioration et la transformation d'une plateforme d'apprentissage classique vers un autre plus moderne qui se base sur les nouvelles architectures. Parmi ces architectures on trouve l'architecture orienté service (SOA) Qui s'est fortement imposé dans l'arène en raison de ses caractéristiques et avantages dont les plus importants est la réutilisabilité, la maintenabilité ...etc.

L'architecture orientée service (SOA) est maintenant largement utilisée dans les plateformes E-learning, c'est une approche permettant de réutiliser et d'organiser des ressources existantes, dans une solution autorisant une interopérabilité entre plateformes et environnements, une évolutivité des modules applicatifs et une flexibilité autorisant l'utilisation dynamique d'applications. Cette solution permet donc d'intégrer divers systèmes : chaque ressource peut être accessible en tant que service possédant une interface. L'implémentation du fournisseur de service est donc libre de changer sans qu'il y ait un impact sur son utilisation. On peut voir ce service comme une boîte noire : on sait qu'elle va rendre le service voulu sans savoir comment est faite la boîte noire. On peut choisir de la remplacer par un autre service implémenté différemment mais répondant aussi à la même fonctionnalité. [w13]

L'implantation d'une architecture orienté service dans une plateforme E-learning se base essentiellement sur les services, l'identification de ces services est la pierre angulaire dans l'implémentation de cette architecture, la plupart des approche actuelles relais sur les descriptions des processus métiers pour identifier les services.

A travers notre PFE « réingénierie d'une plateforme d'apprentissage existante vers une plateforme orientée services », nous avons essayé de partir d'une description générale d'une plateforme e-learning classique puis de modéliser ses fonctionnalités autonomes en utilisant la norme BPMN 2.0 afin d'identifier des services candidats à partir des diagrammes BPMN à travers notre Framework implémenté avec JAVA. Le but final est de permettre de réutiliser ces fonctionnalités de façon distante et de les mettre à la disposition d'autres plateformes.

Notre mémoire est organisé en cinq (05) chapitres, ils sont comme suit :

Chapitre 01 : « Réingénierie et migration vers SOA »

Dans ce chapitre nous parcourons brièvement les concepts de base de la SOA, de la réingénierie et de la migration vers la SOA.

Chapitre 02 : « les systèmes E-Learning à base de SOA »

Dans ce chapitre nous avons montré la relation entre SOA et le E-learning à travers une synthèse générale de travaux antérieurs dans le domaine de l'intégration de SOA dans le E- learning.

Chapitre 03 : « modélisation des processus métiers »

Dans ce chapitre nous avons présenté l'approche processus métier, la norme BPMN et ces composantes graphiques ainsi que le cycle de vie d'un processus puis nous avons conclu avec la relation entre BPM et SOA.

Chapitre 04 : « Conception »

Dans ce chapitre nous avons expliqué notre approche pour répondre à notre objectif qui consiste à identifier de façon automatique les services dans une plateforme d'apprentissage existante en partant des diagrammes modélisés avec BPMN.

Chapitre 05 : « Implémentation »

Dans ce dernier chapitre nous avons présenté les outils de l'implémentation, les interfaces de notre Framework et les résultats obtenus.

Nous terminons par une conclusion générale et quelques perspectives.



Chapitre 01 :
Réingénierie et migration
vers SOA

1. Introduction :

Avec L'évolution rapide des nouvelles technologies, de nouveaux types d'architecture étaient nécessaires. L'une de ces architectures importantes c'est l'orienté service (SOA).

La SOA a changé la méthode de développement de système conventionnelle et a permis à l'intégration du système de devenir plus flexible. La réduction des coûts est un enjeu majeur dans la construction de systèmes d'apprentissage en ligne. Les composants logiciels de la SOA et des services Web se caractérisent par le fait qu'ils sont réutilisables et interchangeables, et sont donc capables de réduire le gaspillage de ressources pédagogiques, ainsi que les coûts de développement du système. [27]

La réingénierie logicielle est un processus qui a pour objectif la transformation et l'amélioration d'un logiciel à partir du modèle existant. L'objectif suivi est d'améliorer la maintenabilité du logiciel et d'assurer une migration des logiciels anciens vers de nouveaux matériels plus performants et de nouvelles technologies de programmation plus modulables et plus aisées à maintenir. [25]

Ce chapitre est structuré en deux parties, la première partie se focalise essentiellement sur l'architecture orientée service (SOA), les services web et la deuxième partie du chapitre est axée sur la réingénierie et la migration vers SOA.

I. L'Architecture Orientée Service (SOA) :

1. Définition de la SOA :

La SOA est un style architectural qui permet de diviser les applications en un ensemble de services, ces services sont des modules assurant des fonctionnalités de l'application avec des interfaces exposées et qui sont invoquées par messages. [23]

2. Les avantages : [43]

On peut identifier lors de l'application de l'architecture orienté service les avantages suivants :

- **Assurer une interopérabilité intrinsèque** : l'adoption du concept de service permet aux différentes solutions du système TI d'échanger des données et des fonctionnalités entre elles, même s'ils sont développés en différents langages.

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

3. Limites de l'architecture SOA : [43]

Un sondage réalisé en 2006 sur 85 entreprises par [45] (Figure 1.1), montre que les principales limites d'une architecture orientée services sont :

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

Comme toute technologie, SOA a ses avantages et ses limites, l'important est de bien profiter de ses apports et savoir pallier ses limites, pour cela il faut adopter une stratégie bien déterminée et savoir choisir la manière avec laquelle on va mettre en place cette architecture. [43]

Si vous estimez que l'outillage SOA est encore immature, quelles sont à vos yeux ses faiblesses principales ?



Figure 1.1 : Les principales limites de SOA. [45]

4. Les caractéristiques d'une SOA : [7]

Une SOA est une architecture qui se veut indépendante technologiquement. Mais les technologies qui implémentent cette architecture ne sont évidemment pas neutres.

Une SOA doit, pour être reconnue comme telle, respecter certaines caractéristiques.

Le Tableau 1.1 ci-dessus regroupe ces caractéristiques.

Caractéristiques	Description
Exposé	Les applications exposent leurs services pouvant être accédés par d'autres applications ou services.
Distribué	Les services sont distribués et peuvent être installés à l'intérieur ou à l'extérieur des pare-feu de l'organisation.
Faible couplage	Les services sont faiblement couplés.
Propriété	Un ensemble de services n'a pas nécessairement un seul propriétaire.
Standard ouvert	Les services utilisent des standards ouverts plutôt que propriétaires.
Neutralité	Les services sont indépendants de la plateforme, du langage et

	du vendeur.
Création	De nouvelles applications, de nouvelles transactions ou de nouveaux services peuvent être créés par l'assemblage, la composition et le chaînage de services de plus bas niveau.
Transparence	Le détail de l'implémentation d'un service (transport de bas niveau, communication, plateforme, langage) est dissimulé à l'application qui utilise le service.
Réutilisabilité	La réutilisation des services est encouragée autant que possible.

Tableau 1.1 : Les caractéristiques d'une SOA [46]

5. Les types des technologies SOA : [w5]

- a) Les services web de type *representational state transfer* (REST) exposent entièrement ces fonctionnalités comme un ensemble de ressources (URI) identifiables et accessibles par la syntaxe et la sémantique du protocole HTTP. Les Services Web de type REST sont donc basés sur l'architecture du web et ses standards de base : HTTP et URI ;
- b) Les *services web WS*, exposent ces mêmes fonctionnalités sous la forme de services exécutables à distance. Leurs spécifications reposent sur les standards SOAP et WSDL pour transformer les problématiques d'intégration héritées du monde middleware en objectif d'interopérabilité.

6. Caractéristiques des services :

Le service est caractérisé par :

- **Contrat standardisé** : L'ensemble des services d'un même système technique sont exposés au travers des contrats respectant les mêmes règles de standardisation (l'interface) [w5].
- **Couplage lâche** : Le couplage faible (loosely-coupled) est une propriété qui montre le degré de liaison des services. Les services sont connectés aux autres services ou aux

clients à travers des documents standards, ces documents sont réalisés en XML de la même manière que les web services et ils assurent le découplage ou autrement dit la réduction des dépendances [w4].

- **Abstraction** : Le contrat d'un service ne doit contenir que les informations essentielles à son invocation. Seules ces informations doivent être publiées [w5].
- **Réutilisabilité** : Les services doivent encapsuler une logique de traitement suffisamment générique pour être utilisés dans des contextes d'utilisation différents [w5].
- **Autonomie** : Un service doit être totalement autonome. On doit pouvoir le remplacer ou le déplacer sans que cela affecte d'autres services. Un service implémente ses propres composants et ses propres méthodes d'accès aux données, il ne doit dépendre d'aucun élément externe [w6].
- **Sans état (stateless)** : Son exécution ne dépend pas de l'invocation préalable d'autres services, l'ensemble du contexte qui est fourni lors de l'invocation par le consommateur suffit au service pour exécuter ses actions [w6].
- **Découvrabilité** : Un service est complété par un ensemble de métas données de communication au travers desquelles il peut être découvert et interprété de façon effective.
- **Composabilité** : Un service doit être conçu de façon à participer à des compositions de services [w6].
- **Synchrone ou asynchrone** : Attente de réponse après l'utilisation d'un service ou non. [w4]

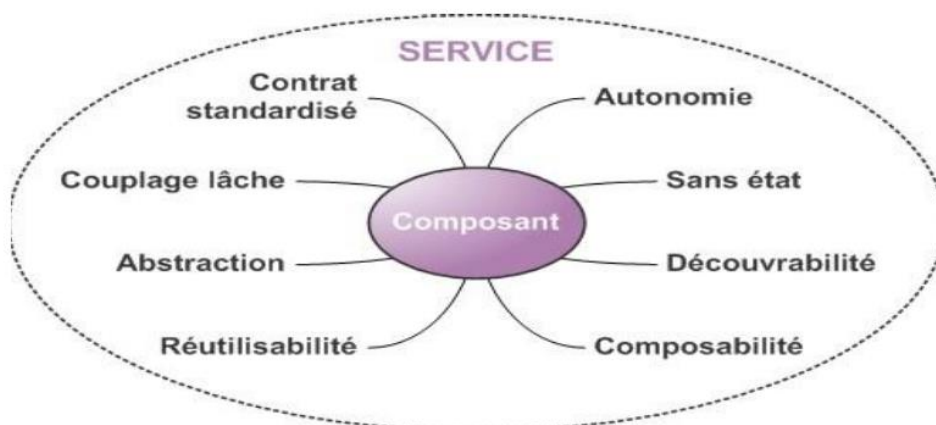


Figure 1.2 : Les caractéristiques de service [w5].

7. Les services Web : [7]

Les services Web sont une implémentation particulière et communément utilisée de SOA. Un service Web est un service logiciel auquel on accède par l'intermédiaire du Web (Daniel, 2003). Les services Web fournissent un standard d'interopérabilité entre différentes applications logicielles fonctionnant sur une variété de plateformes. Les services Web sont caractérisés par leur grande interopérabilité et leur grande extensibilité grâce à l'utilisation du standard XML.

Fondamentalement, un service Web exécute une tâche bien précise et permet à deux systèmes de communiquer entre eux.

Les services Web sont préférablement faiblement couplés et ont une forte cohésion interne. Cela implique qu'un service exécute une action bien précise et que pour réaliser cette action, les dépendances à d'autres services ont été ramenées au minimum.

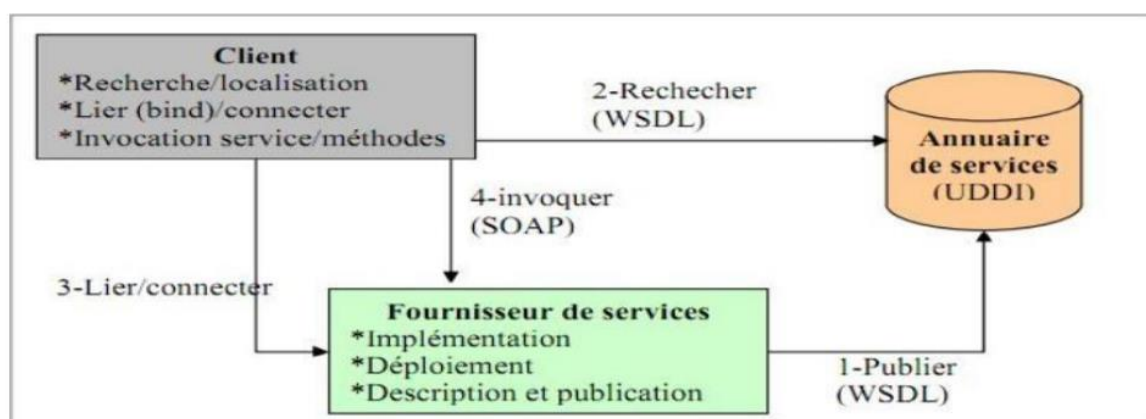


Figure 1.3 : Architecture des web services. [47]

8. Les standards du Web Service :

A. Le Langage de Définition de Web Service (WSDL):

WSDL est un format XML permettant de décrire les services réseau comme un ensemble de points de terminaison fonctionnant sur des messages contenant des informations orientées document ou procédure. Les opérations et les messages sont décrits de manière abstraite, puis liés à un protocole de réseau concret et à un format de message pour définir un point d'extrémité. Les points de terminaison concrets associés sont combinés en points de terminaison abstraits (services). WSDL est extensible pour permettre la description des points

de terminaison et de leurs messages quels que soient les formats de message ou les protocoles réseau utilisés pour communiquer, cependant, les seules liaisons décrites dans ce document décrivent comment utiliser WSDL en conjonction avec SOAP 1.1, HTTP GET / POST et MIME. [w2]

B. SOAP (Simple Object Access Control):

SOAP est une recommandation W3C qui le définit comme un protocole léger destiné à l'échange d'informations structurées dans un environnement distribué et décentralisé. SOAP est basé sur XML afin de mettre en place un mécanisme extensible d'échanges de messages à travers une variété de protocoles. Il a été conçu dans l'optique d'être indépendant du modèle de programmation de l'application ou d'une sémantique particulière. SOAP vise deux buts : la simplicité et l'extensibilité. SOAP peut être utilisé dans tous les styles de communication : synchrone ou asynchrone, point à point ou multipoints, intranet ou internet. [24]

C. UDDI (Universal Description, Discovery and Integration) :

UDDI ou Universal Description, Discovery and Integration a pour but d'établir un format d'annuaire des services Web. UDDI est le fruit d'un effort commun des grands fournisseurs de plates-formes et de logiciels au sein du Consortium de standards OASIS. UDDI a créé une plate-forme interopérable qui permet aux compagnies et aux applications de trouver et utiliser rapidement, facilement et dynamiquement les services Web à travers le Web. [24]

II. Réingénierie :

1. Définition :

La réingénierie consiste à reconcevoir, repenser ce qui a été conçu dans une démarche d'ingénierie. Il est difficile, voire impossible de faire de la réingénierie si ce qui a été conçu précédemment n'a suivi aucune règle, aucune norme.

2. Les différents Types de la réingénierie :

Dans le secteur de l'informatique il existe plusieurs types de réingénierie :

- Réingénierie des systèmes d'informations.
- Réingénierie des processus.
- Réingénierie logicielle [w7].

2.1. La réingénierie des systèmes d'information :

Il s'agit d'une remise à plat de tout ou partie d'un système d'information, afin d'atteindre de meilleurs niveaux de performances globales. Cela peut par exemple consister à transformer une architecture informatique initialement tournée vers la production en un dispositif orienté vers le client [w8].

2.2. La réingénierie des processus :

La réingénierie des processus traite la transformation des processus de l'entreprise, et ce, essentiellement au sein des grandes entreprises. Cette transformation est toujours le fruit d'un processus ou d'un plan de gestion du changement maîtrisé, dans le cadre duquel une organisation audite et compare les processus existants puis tente de les optimiser [w9].

2.3. La réingénierie des logiciels :

La réingénierie logicielle est un processus qui a pour objectif la transformation et l'amélioration d'un logiciel à partir du modèle existant. Le logiciel en question ne doit pas subir de régression au niveau fonctionnel mais doit gagner en performance et en évolutivité. L'obsolescence logicielle est rapide dans un secteur en évolution permanente et augmente le coût de la maintenance. Il y a pénurie des compétences pour ce qui concerne les plus vieux logiciels. Ces anciennes applications ne sont pas adaptées au changement de technologie et perdent de ce fait de leur performance. [w1]

3. Réingénierie et migration vers SOA :

La migration ou la modernisation implique des changements plus étendus qu'une simple maintenance, mais une partie significative de système est conservée. Ces changements incluent souvent la restructuration du système, l'amélioration des fonctionnalités ou la modification des attributs du logiciel. [8]

3.1. Bénéfices de la migration :

La migration offre plusieurs bénéfices aux entreprises et à leurs services, parmi ces bénéfices nous citons [9] :

- L'adaptation aux nouveaux besoins,

- L'amélioration des services clients,
- Une intégration plus étroite avec les partenaires et les fournisseurs,
- La réduction du coût d'usage des systèmes d'information,
- L'amélioration de la qualité des données,
- L'amélioration du contrôle et de la gestion de sécurité,
- L'augmentation de la flexibilité et la réactivité des systèmes d'information,
- L'élimination de la dépendance forte à l'ensemble des anciennes compétences.

3.2. Difficultés de la migration :

Les efforts de la modernisation de systèmes d'information d'entreprises ont échoué dans la plupart du temps à cause de plusieurs facteurs, nous citons les suivants [9] :

- La complexité des systèmes patrimoniaux : la complexité est considérée comme le plus grand limiteur du processus de migration, elle est produite à cause de la taille énorme de système, de l'incompréhensibilité des systèmes à migrer et des phases successives de maintenance.
- Les risques de migration : les risques de migration ne sont pas majeurs, IL est possible d'accepter un certain risque si nous devons accomplir une tâche de migration.

Malheureusement, beaucoup d'entreprises sont incapables ou ne veulent pas contrôler le risque correctement. Ceci également prévenir de l'insuffisance de compréhension de gestion des risques et des techniques de réduction du risque.

3.3. Différentes approches de migration

La migration ou la modernisation du système peut être distinguée par la stratégie de développement exigée pour accomplir la migration. Il existe deux approches proposées pour la migration, qui sont l'approche "*Top Down*", et l'approche "*Bottom Up*" [10]. Ces dernières sont considérées comme des approches de base. D'autres approches baptisées "*Outside In*" (aussi appelée *Meet In The Middle*) et "*Middle Out*" sont des approches hybrides des deux approches de base, et qui sont proposées pour pallier les désavantages des approches *Top Down* et *Bottom Up* [11]. Le choix entre ces approches dépend de certains critères décisifs, tel que :

- L'état des systèmes informatiques existants (nature, degré de complexité, capacités fonctionnelles et techniques).
- Les objectifs attendus de la migration.

- Les capacités des groupes de travail...etc.

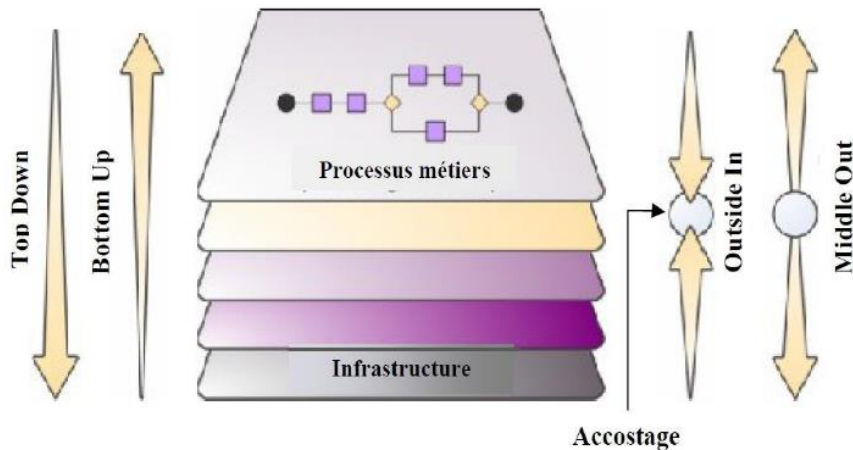


Figure 1.4 : Différentes approches de migration [11].

3.3.1. Approche Top Down :

Dans ce type de projet d'intégration le système d'information de l'entreprise est vu comme un super bloc applicatif. Cette approche est descendante et globalisante dans la mesure où elle touche à la globalité de l'entreprise. Elle vise à créer un socle commun d'intégration aligné sur le métier de l'entreprise.

Le point de départ de cette approche est la définition et la formalisation des processus métiers, qui représente les objectifs fonctionnels de l'entreprise, pour descendre ensuite au travers des différentes couches du système afin de définir les services nécessaires à la réalisation de ces processus.

Cette approche peut être appliquée dans le cadre d'un système urbanisé ou pour démarrer un nouveau projet. Cependant, elle est très coûteuse, très longue (plusieurs années) et trop risquée, puisqu'elle cause la reconstruction de tout ou d'une partie de système d'information [12] [13].

3.3.2. Approche Bottom Up :

A l'inverse de l'approche "Top Down", le système d'information de l'entreprise dans cette approche est vu comme un ensemble de blocs applicatifs autonomes et hétérogènes.

L'approche "Bottom Up" est une approche ascendante, conduisant à des projets départementaux où chaque projet démarre par une phase d'analyse de l'existant afin de

déterminer les fonctions transverses existantes dans le système. Ensuite, ces fonctions sont mises en mode service, pour qu'ils soient exploitables au sein d'un autre service et/ou un processus métier.

Cette approche est intéressante, puisqu'elle contraint à réaliser une cartographie du système, ce qui facilite la publication et la réutilisation de ses fonctionnalités intéressantes.

Cependant, cette approche présente un inconvénient majeur, est qu'elle ne prend pas en charge les nouveaux besoins de l'entreprise, dans la mesure où elle se limite à encapsuler les fonctions existantes au niveau du système dans des services, qui restent très fortement couplés à leur application d'origine [14] [13].

3.3.3. Approche Outside In :

Cette approche est faite d'itérations successives de l'approche "*Top Down*" et de l'approche "*Bottom Up*", afin de définir les services métiers qui satisfont au mieux les exigences et les contraintes métiers d'une part, et informatiques d'autre part.

L'approche "*Outside In*" propose de démarrer en parallèle, l'approche *Top-Down* "sans se préoccuper de l'existant", pour définir les besoins métiers en processus métiers et les services nécessaires à leur réalisation. Et l'approche *Bottom Up* "en ne considérant que l'existant", afin de cartographier l'existant applicatif dont dispose l'entreprise pour supporter les services métiers à forte valeur ajoutée métier.

Une fois ces deux chantiers en phase finale, commence l'étape de l'accostage. Son objectif est de réconcilier les résultats des deux approches afin de déterminer comment seront réalisés les processus métiers. Il faut, pour cela, comparer les besoins en services exprimés par l'approche *Top Down* avec ceux remontés de l'approche *Bottom Up*.

Cette approche réunit les bénéfices des approches *Top Down* et *Bottom Up*. Elle permet de piloter le projet d'intégration par les besoins métiers tout en facilitant la réutilisation de services et la capitalisation sur l'existant. Mais la réussite de cette approche dépend de la délicate étape de l'accostage qui nécessite de faire les compromis nécessaires pour réutiliser le maximum de code [11].

3.3.4. Approche Middle Out :

Par opposition à l'approche *Outside In*, cette méthode propose de commencer au milieu « en anglais : *In the middle* », c'est-à-dire là où le métier et les technologies d'information (ou,

IT pour *Information Technology*) parlent le même langage (en tout cas presque). Elle s'attaque donc d'emblée à ce qui reste un des facteurs limitateurs à l'adoption des SOA : La compréhension du métier de l'entreprise par les maîtrises d'œuvre et inversement, la compréhension des contraintes IT par les maîtrises d'ouvrage.

Une fois les différentes parties sont d'accord sur un premier socle de services "métiers" nécessaires :

- Les maîtrises d'ouvrage engagent un chantier "*Middle Up*" pour spécifier les processus métiers.
- Les maîtrises d'œuvre engagent un chantier "*Middle Down*", pour spécifier le socle de services de plus bas niveau permettant la réalisation des services métiers.

Cette approche limite par contre le pilotage de la SOA par les besoins métiers, puisque le point de départ est l'identification des services métiers nécessaires et non la définition des processus réalisant le métier de l'entreprise [15] [11].

4. Conclusion :

Nous avons présenté dans ce chapitre un état de l'art sur l'architecture orienté service (SOA), ces avantages et ces inconvénients, ainsi que les services web qui sont considérés comme la technologie d'implémentation de base les plus utilisés dans ce type d'architecture.

La deuxième partie du chapitre a présenté les différents types de migration vers SOA et les bénéfices de cette opération.

Le chapitre suivant présente les systèmes e-learning et les travaux dans le domaine de migration des plateformes E-Learning vers une architecture orienté service (SOA).



Chapitre 02 :
Les systèmes E-Learning à
base de SOA

1. Introduction :

La richesse des ressources sur Internet et la technologie multimédia ont donné naissance à une méthode d'apprentissage qui utilise Internet pour transférer et extraire des informations d'apprentissage et des contenus [1], qui est le concept de l'e-learning. Le 'e' dans l'e-learning représente l'électronique ou sur le Web, et c'est exactement ce qu'est l'e-learning, en utilisant la technologie de l'enseignement électronique pour participer à des activités d'apprentissage sur Internet.

SOA (architecture orientée services) a changé le développement du système conventionnel et a permis à l'intégration du système de devenir plus flexible. En raison de son couplage lâche, SOA permet de terminer la construction des systèmes en combinant les composants [3].

Dans ce chapitre nous présentons les systèmes E-Learning et ce qu'a apporté l'architecture orientée service (SOA) à la modernisation de ces systèmes d'apprentissage en ligne.

2. Les systèmes E-Learning :

2.1. Définition :

La formation « en ligne » dite « e-Learning » est "l'utilisation des nouvelles technologies multimédias de l'Internet pour améliorer la qualité de l'apprentissage en facilitant d'une part l'accès à des ressources et à des services, d'autre part les échanges et la collaboration à distance (définition par la Commission Européenne, 2001). [22]

Le E-Learning est adapté pour développer un espace d'éducation avec des méthodes spécifiques pour aider l'étudiant de progresser dans ses études à n'importe quel moment et n'importe quel endroit.

2.2. Les avantages de E-Learning :

- Le E-Learning permet à l'apprenant de suivre ses études suivant sa situation.
- D'avoir des diplômes en plus.
- Réduire les frais de vie.
- De gagner le temps par l'apprentissage de beaucoup de choses en temps court et à n'importe où.
- Formation au lieu de travail.

- Favoriser la créativité : accroître la compréhension et les connaissances de l'apprenant.
- Solution pratique et économique : sur place pas de déplacement, l'accès non limité et à n'importe quelle heure.

2.3. Les inconvénients :

- Pas de démonstrations sur les cours : il n'y a pas d'enseignants à la disposition des étudiants.
- Les différents problèmes techniques du fonctionnement du système de formation.
- Le type de communication est limité (virtuelle).
- Risque élevé d'abandon avant la fin des parcours d'apprentissage.
- L'isolement de l'apprenant dans le cas du manque d'interaction avec les autres acteurs du système d'apprentissage.

2.4. Les composants d'une plateforme E-Learning :

Composants	Description
Composantes technologiques	<ul style="list-style-type: none"> • Moyens techniques. • Infrastructure réseau. • Multimédia.
Composantes pédagogiques	<ul style="list-style-type: none"> • Relations tuteur-coach. • Accompagnement des apprenants. • Collaboration entre les différents acteurs.
Composantes didactiques	<ul style="list-style-type: none"> • Les modules avec des différentes versions (PDF, vidéo..). • L'actualisation du contenu.

Tableau 2.1 : Les différents composants des E-Learning

2.5. Les exemples sur les plateformes E-Learning : [5]

Il existe plus de 200 plates-formes pédagogiques disponibles. Plus 30 sont libres. Il n'est pas toujours aisé de faire un choix correspondant à ses besoins. De plus, un tel choix représente un engagement dans le temps, et un retour en arrière impossible dans certains cas. Dans un souci d'interopérabilité, des normes ont été mises en place dans ce domaine. Ils existent deux types de logiciel open source et logiciel payant :

- **Open source** : Les logiciels open source sont des logiciels gratuits, que chacun peut modifier à sa guise. Cela est particulièrement utile pour les grandes organisations qui gagnent en autonomie et en interopérabilité tout en conservant, si elles le souhaitent, des garanties sur le résultat moyennant un contrat (payant) de support.
- **Payant** : C'est un logiciel que l'on peut utiliser en libre essai durant une période ou un nombre d'utilisations donnés. Si son utilisateur souhaite utiliser définitivement le logiciel, il a l'obligation de rétribuer l'auteur du logiciel.

2.6. Les systèmes de gestion d'E-Learning : [6]

Les cours en ligne sont créés, gérés et utilisés grâce à des systèmes de gestion d'e-learning appartenant aux catégories suivantes :

- ***Learning Management System (LMS)*** :

LMS est une famille de systèmes, qui permet la gestion de toutes les activités de la formation. Les systèmes LMS catégorisent les utilisateurs, leur donnent certaines autorisations à des modules de formation et affectent les utilisateurs à des groupes spécifiques de formation. MOODLE, Dokeos et Sakai sont des exemples de LMS.

- ***Learning Content Management System (LCMS)***

LCMS est une famille de systèmes d'E-Learning les plus avancés technologiquement. Les systèmes LCMS intègrent des fonctionnalités des LMS et des CMS, assurant à la fois la création et la gestion des contenus éducatifs. Ils offrent la possibilité d'évaluer les connaissances assimilées par les utilisateurs. Parfois, la catégorie suivante est aussi distinguée : Exemples de LCMS : TopClass, SimplyLearn.

- ***Virtual Classroom System (VCS)***

Les VCS sont des systèmes qui incluent les fonctionnalités suivantes :

- Transmission de la voix et de la vidéo en temps réel entre tous les participants.
- Tableau blanc partagé (shared whiteboard).
- Espace intégré pour la projection de diapositives ou autres supports visuels.
- Capacité d'interaction textuelle, y compris les conversations ou « note-passing».
- Des moyens permettant aux apprenants d'indiquer qu'ils ont des questions.
- Outils pour évaluer l'humeur actuelle, les opinions et la compréhension ainsi que pour solliciter des questions ou des commentaires

- ***Intelligent Tutoring Systems (ITS)***

Les ITS sont similaires aux LMS. Ils peuvent donner des feedbacks intelligents à l'utilisateur. Les ITS emploient les techniques de l'intelligence artificielle, pour pouvoir comprendre, informer et diriger l'apprenant quand il termine ses exercices ou ses tests. Ils visent à répliquer le rôle du tuteur qui guide et forme les apprenants d'une manière efficace. Le tuteur humain est souvent remplacé par des entités intelligentes, appelées « agents » capables de suivre et guider l'apprenant durant son apprentissage.

- ***Le social Learning, Un nouvel air du e-Learning***

Bien que le terme social Learning fût utilisé bien avant l'apparition du e-Learning, le terme social Learning représente un nouveau phénomène du Web 2.04. Actuellement, l'expression de «social learning» s'est répandue sur le web comme un concept qui caractérise l'utilisation des médias électroniques synchrones ou asynchrones pour le développement des savoirs, par le biais de connexion avec des collègues, des mentors ou des experts dans une optique collaborative.

3. SOA et le E-Learning : [20]

L'architecture orientée service est un style architectural qui prend en charge l'orientation service [w14]. Il s'agit essentiellement d'un ensemble de services qui interagissent et communiquent. Cette communication peut être un simple retour de données ou une activité (coordination de plusieurs services).

Le concept de service est mieux défini dans un contexte commercial que dans un contexte technique. Cependant, dans un contexte d'entreprise, un service peut être décrit comme un moyen de spécifier une fonctionnalité métier encapsulée indépendante des implémentations concrètes. Dans ce contexte, un service est davantage un concept d'entreprise qu'un concept informatique [16].

3.1. Avantage d'intégration de la SOA dans le e-learning :

L'intégration de SOA au système d'apprentissage en ligne peut entraîner plusieurs avantages, notamment :

- Modularité : les services sont couplés dynamiquement ; cela facilite l'ajout ou le remplacement de services, même avec un système en cours d'exécution.
- Interopérabilité : les services communiquent à l'aide de protocoles standard et sont définis à l'aide de spécifications standard. Cela signifie que les spécifications tierces peuvent facilement être utilisées avec les systèmes basés sur les services existants.
- Extensibilité : en raison de leur nature modulaire et interopérable, les systèmes basés sur les services peuvent être facilement étendus, ce qui réduit le danger de «verrouillage» technologique [17].
- Agilité : Les services d'apprentissage en ligne peuvent être offerts par divers fournisseurs de logiciels. Ils doivent être découverts au moment de l'exécution, permettant ainsi une sélection et une utilisation flexibles des services appropriés sur les réseaux, car les besoins des utilisateurs changent constamment.
- Réduction des coûts. Au cours du processus de développement des services d'apprentissage en ligne, la réutilisation des services existants, plutôt que le développement de composants logiciels sur mesure, permet aux organisations d'utiliser des logiciels rentables [18].
- Distribution : L'avantage d'un service basé sur une plateforme d'apprentissage est que ses composants peuvent être développés dans différentes langues et distribués sur différents serveurs. Ainsi, chaque partie exposée en tant que service peut être utilisée par une plateforme différente.

3.2. Travaux dans le contexte :

De nombreux chercheurs ont réalisé l'importance d'utiliser la SOA dans les systèmes d'apprentissage en ligne. L'intégration, l'interopérabilité, l'évolutivité et la réutilisabilité sont les principaux axes sur lesquels les chercheurs en e-Learning basés sur la SOA tentent d'aborder, de résoudre et d'améliorer. [21]

Bien que certains chercheurs aient ignoré les caractéristiques pédagogiques des systèmes d'apprentissage en ligne, d'autres ont considéré ces caractéristiques comme le principal facteur de motivation de l'adoption de l'architecture SOA dans l'apprentissage en ligne. [21]

Dans [20] Ils ont identifié et fait émerger un ensemble de services qui pourraient être partagés et réutilisés par les systèmes e-learning. Afin de fournir des services de qualité, ils ont respecté les critères essentiels de la SOA, à savoir : autonomie, Independence, couplage lâche avec des parties externes et forte cohésion interne.

Voici ci-dessus l'architecture globale du système e-learning à base de SOA proposé dans [20] :

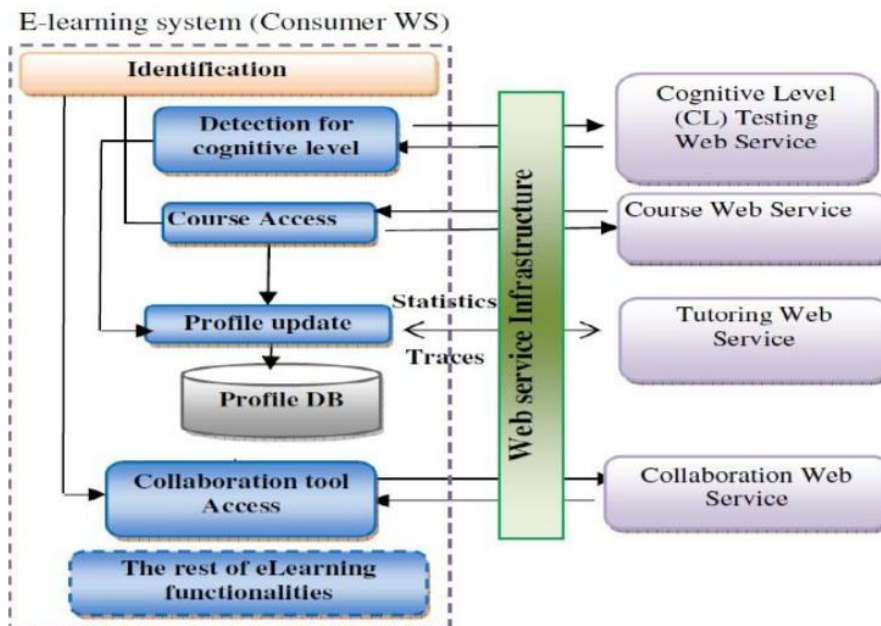


Figure 2.1 : Architecture du système E-learning proposée dans [20]

Dans [26] les auteurs ont développé un système d'apprentissage en ligne qui utilise les technologies de service Web. L'objectif du système proposé était d'assurer l'interopérabilité entre les composants écrits et fonctionnant sur différents matériels.

Ici, un ensemble de services Web e-Learning est construit en choisissant les services requis, notamment les services Web d'évaluation, de gestion des cours, de notation, de marquage, de métadonnées, d'enregistrement et de rapport.

Dans [27], les services Web développés avec la Technologie Microsoft .NET et XML pour construire une plateforme d'enseignement avec des spécifications standard, permettant aux développeurs de systèmes de construire rapidement un système d'apprentissage en ligne basé sur la SOA et des services Web.

Dans [28], un système « LearnServe » repose sur l'hypothèse qu'un système d'apprentissage typique est un ensemble d'activités ou de processus qui interagissent avec les apprenants et le

contenu choisi de manière appropriée, ces derniers ont été modélisés sous la forme d'objets d'apprentissage.

Ils ont subdivisé la fonctionnalité principale d'un système d'apprentissage en ligne en un certain nombre d'applications autonomes, qui peuvent ensuite être réalisées individuellement ou en groupe en tant que services Web.

La mise en place de ces services permet une réutilisation des fonctionnalités d'une plateforme e-learning.

Voici L'architecture du système proposé :

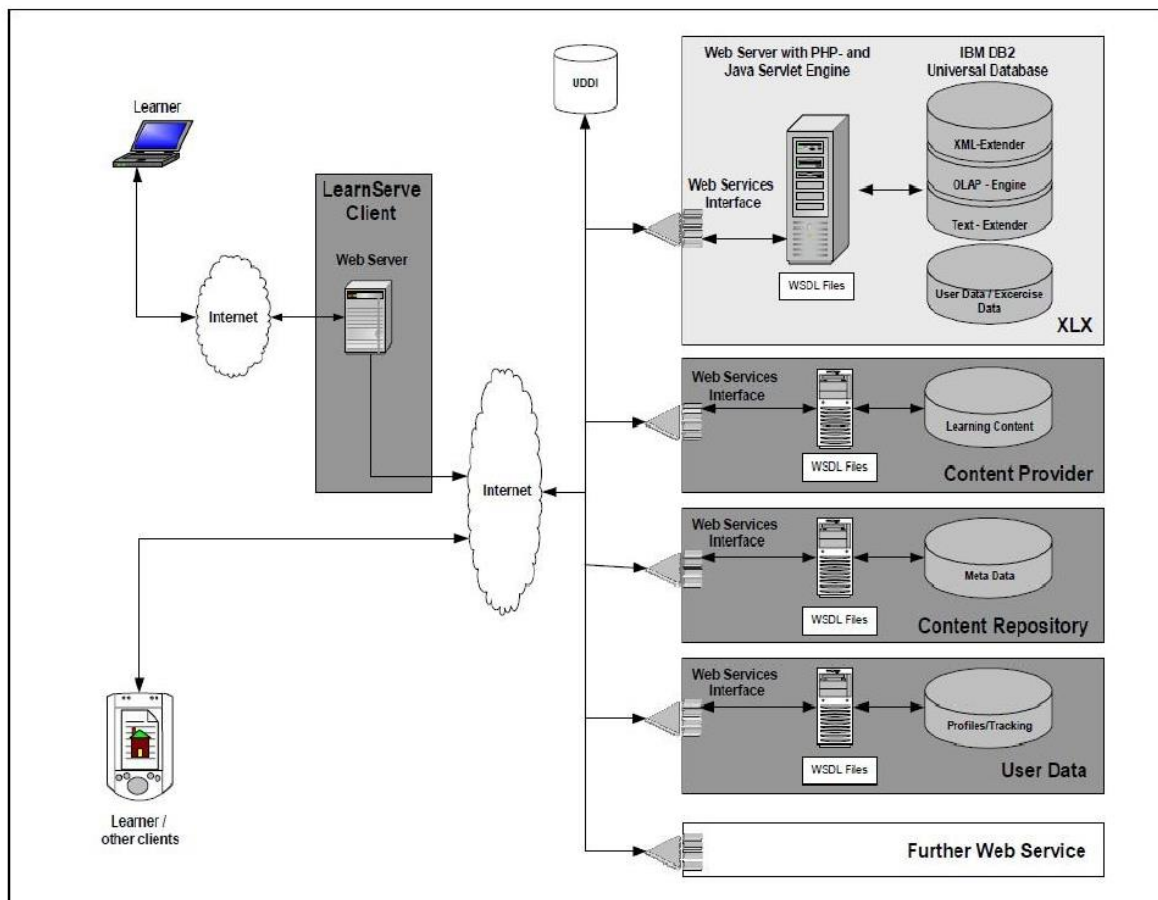


Figure 2.2 : Architecture du système E-learning proposée dans [28]

Dans [29], Ils ont proposés un système «SORAPES » dont ils ont réutilisés des services Web et des objets d'apprentissage. Il s'agit d'une architecture en couches et hautement évolutive pour un système d'apprentissage en ligne personnalisé.

L'objectif de ce travail est de concevoir une architecture de référence orientée services pour les systèmes d'apprentissage en ligne personnalisés (SORAPES) et de valider l'architecture.

Voici L'architecture du système proposé :

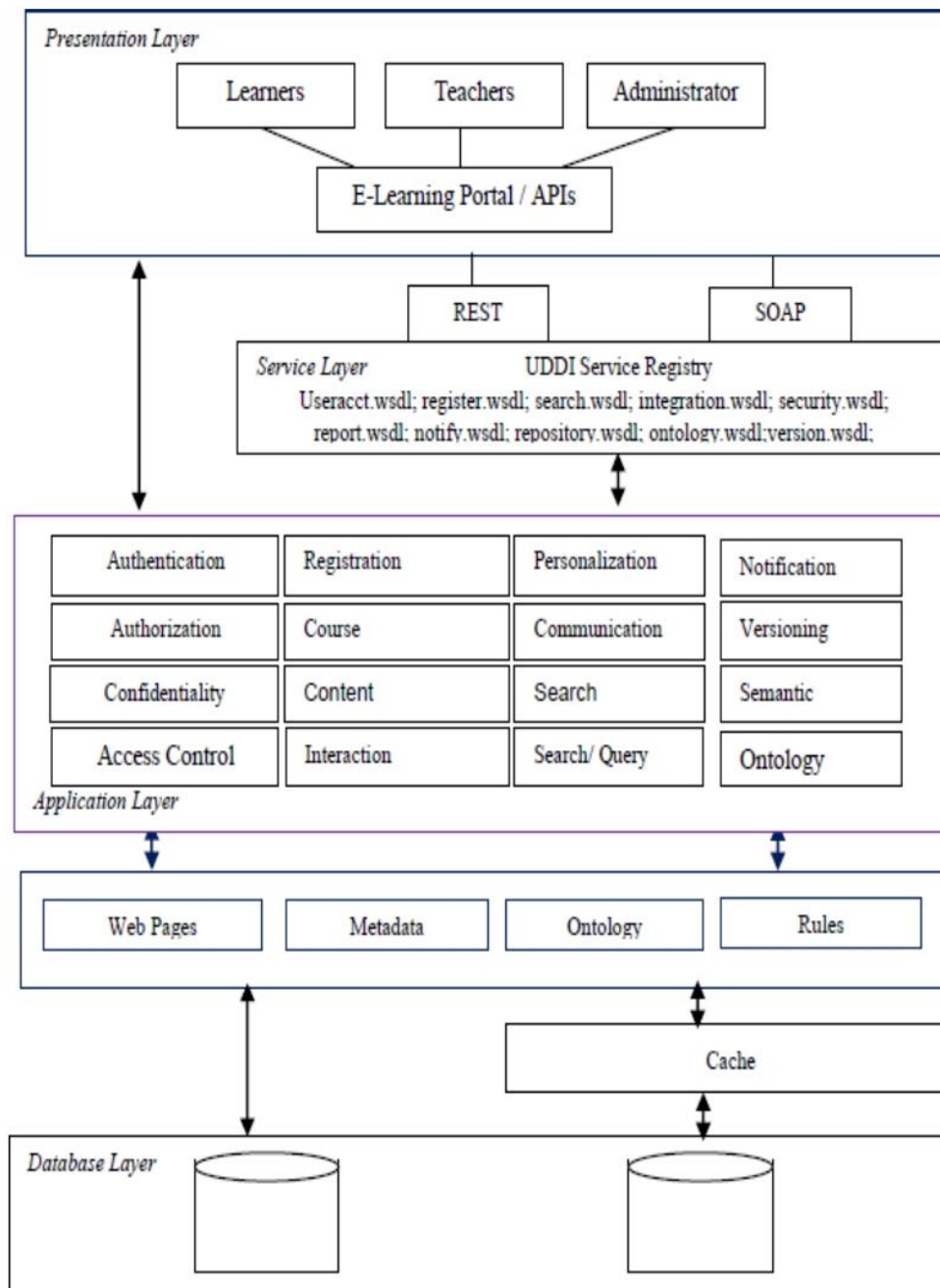


Figure 2.3 : Architecture du système E-learning proposée dans [29]

Pour [30], Ils ont proposé un modèle de recommandation qui combine les avantages de certains modèles de recherche et de classement existants. Il s'agit de créer un modèle de recommandation personnalisé avec un temps de traitement plus court sur la recherche et le classement. Pour archiver les objectifs, plusieurs modifications doivent être apportées sur la base du système e-learning SOA existant.

Voici l'architecture SOA révisée du système d'apprentissage en ligne (avec le modèle de recommandation) qui sont proposés :

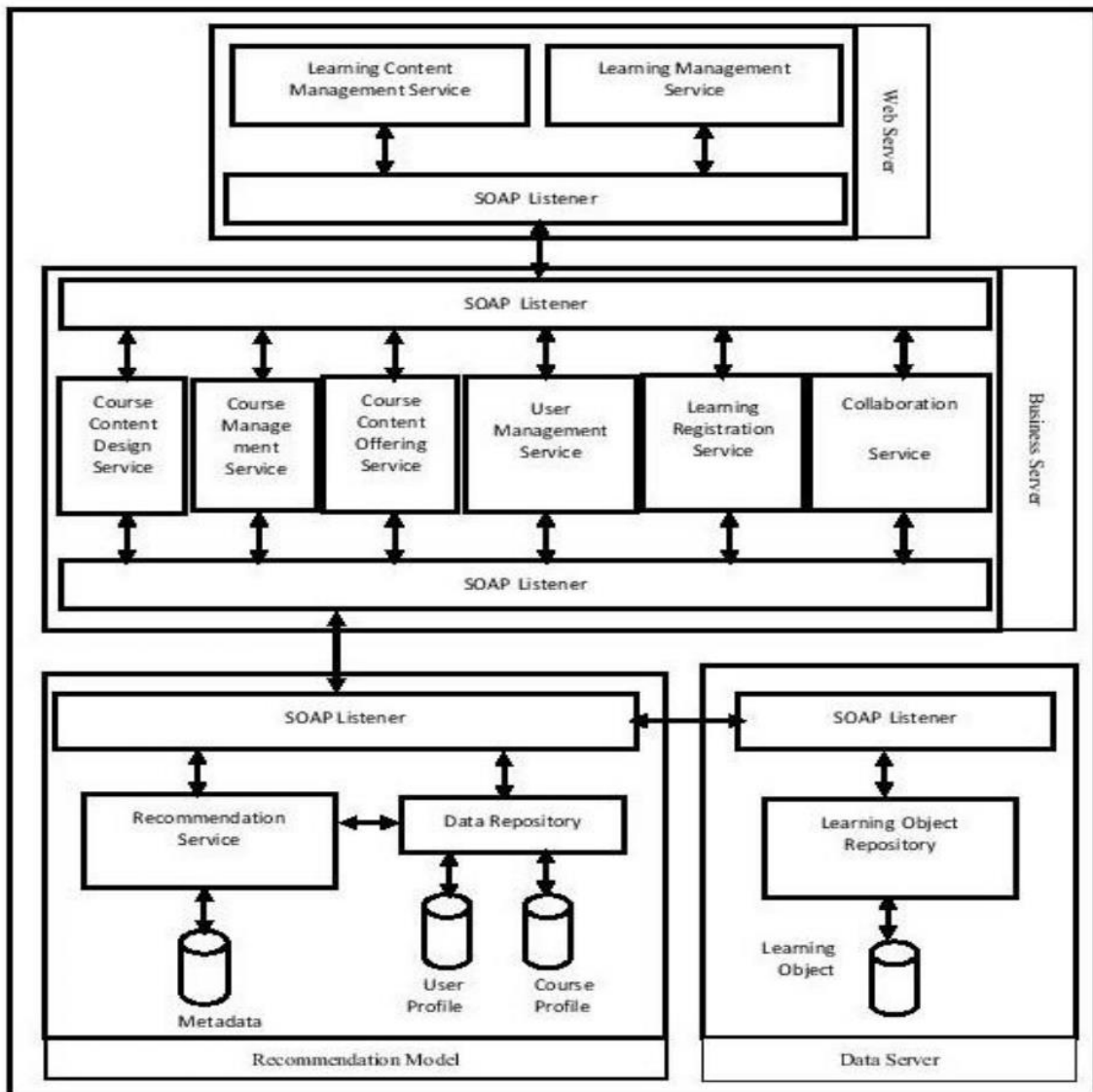


Figure 2.4 : Architecture du système E-learning proposée dans [30].

La recherche de [31] utilise des composants logiciels de SOA et de service Web développés avec les technologies .NET et XML pour créer la plate-forme d'enseignement avec des conditions standard et permet aux développeurs de créer rapidement un système d'apprentissage en ligne basé sur SOA et un service Web. Le système proposé est comme suit :



Figure 2.5 : Architecture du système E-learning proposée dans [31]

Pour [32], Elle a proposé une solution d'adoption SOA pour les universités dans le but de créer un environnement où les apprenants accèdent aux informations numériques afin de soutenir leurs objectifs de formation et de carrière tout au long de la vie.

Voici les services du système E-learning qui a été proposé :

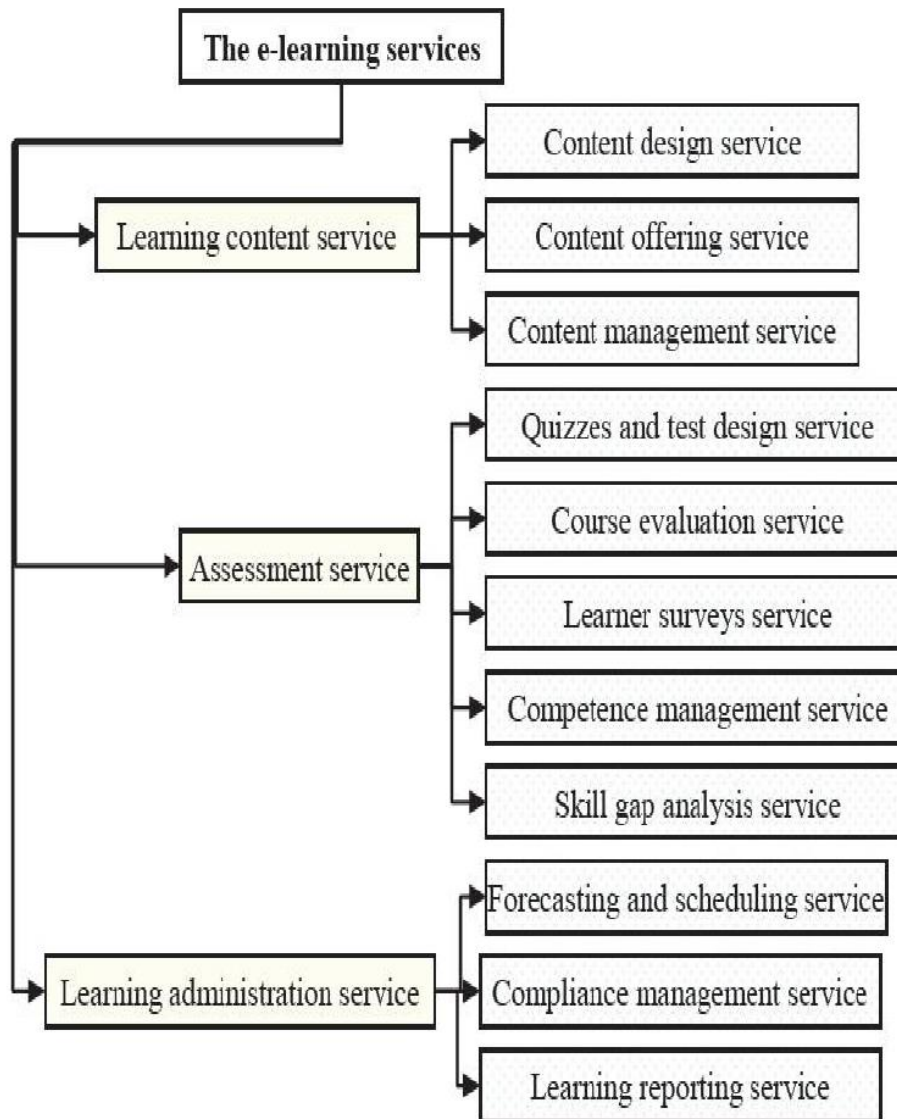


Figure 2.6 : Architecture du système E-learning proposée dans [32]

L'étude de [33] vise à construire un ensemble de services web e-Learning. Avec ces services Web, de nouveaux systèmes d'apprentissage en ligne peuvent être construits en choisissant les services requis. Les services Web développés comprennent l'évaluation, la gestion des cours, la notation, les métadonnées, l'inscription et les services Web de rapport.

Une architecture multicouche a été proposée, divisée en couches de présentation, couche de service E-Learning, couche de service commune et couche de ressources.

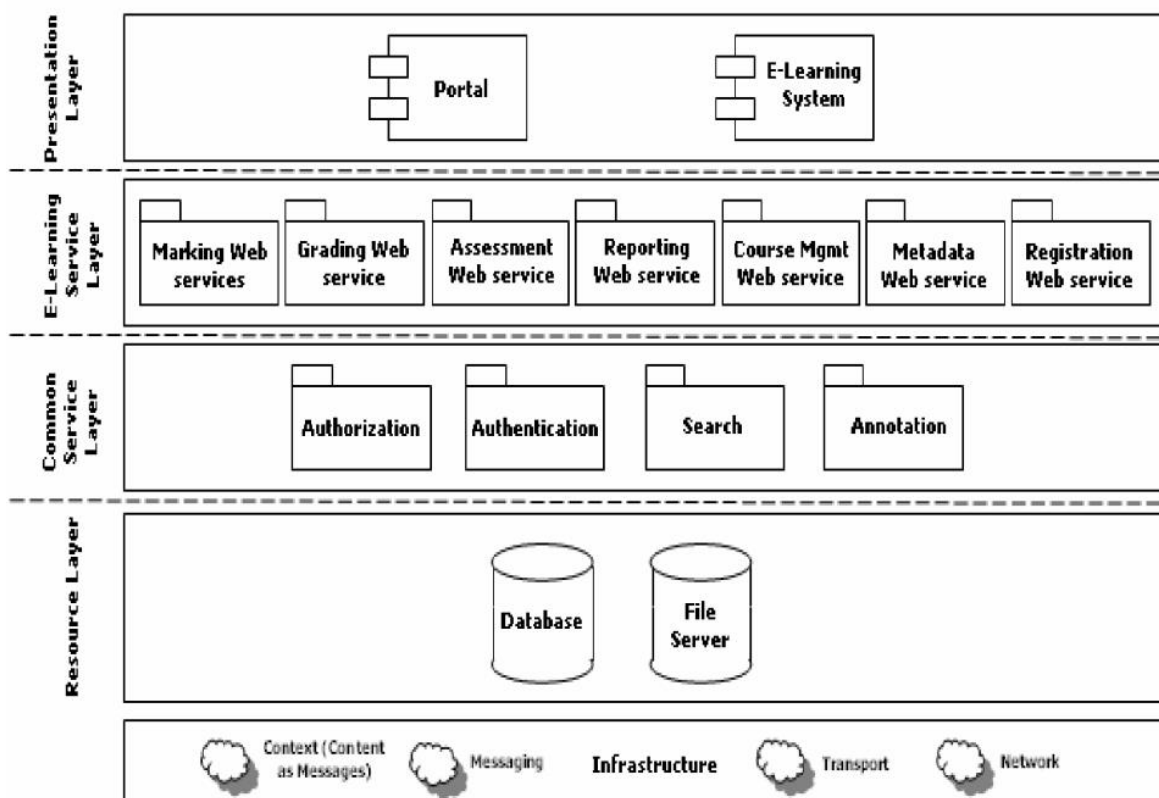


Figure 2.7 : Architecture du système E-learning proposée dans [33]

4. Bilan des travaux similaires :

En faisant un résumé des travaux effectués dans le domaine de l'intégration de la SOA avec le e-learning, on peut dire que la plupart de ces travaux n'ont pas respecté à la lettre les propriétés et les caractéristiques d'une architecture orientée services (SOA), Surtout en ce qui concerne la propriété de l'autonomie des services, en outre le manque de clarté totale concernant les services et la communication entre eux.

Notre travail à deux (02) avantages principaux, ce qui le rend différent du reste des travaux susmentionnés, à savoir :

- Le respect total et strict des propriétés et des caractéristiques des services suivant l'architecture orientée services (SOA) surtout en ce qui concerne la réutilisabilité et l'autonomie des services, où il apparaît dans les diagrammes BPMN (communication avec message).

- Nous nous sommes appuyés sur l'approche Top-Down pour identifier automatiquement les services candidats malgré sa lourdeur mais elle donne une vision globale du projet.

Cela sera expliqué explicitement dans le chapitre de conception et d'implémentation.

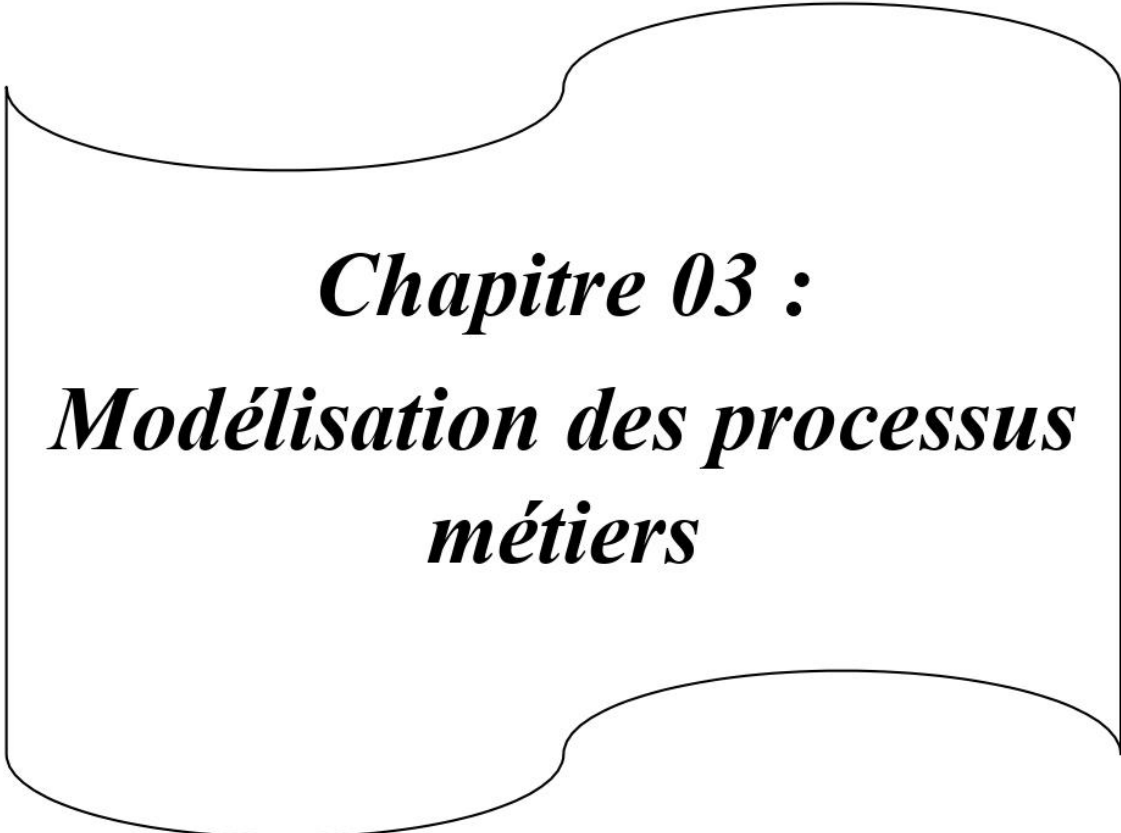
5. Conclusion :

Une plateforme e-learning selon le point de vue de l'architecture orientée services (SOA) est considérée comme un ensemble de services qui coopèrent entre eux pour fournir un ensemble de fonctionnalités aux acteurs de la plateforme. [20]

Les services web ont facilité l'échange et le partage des fonctionnalités réutilisables. La proposition de création d'une plateforme E-Learning basé sur un ensemble de services permet de réutiliser ses derniers dans d'autres applications.

Dans le chapitre qui suit nous allons présenter l'approche processus et la norme BPMN utilisée pour modéliser une plateforme e-learning classique pour la moderniser et la réingénier vers une plateforme basée sur l'architecture orientée service.

Pour moderniser une plateforme d'apprentissage classique et passer vers une nouvelle architecture SOA on a besoin soit de redévelopper la plateforme du début, soit de suivre un processus de réingénierie. Dans notre cas nous avons choisi de procéder à la réingénierie dont les techniques et les différentes approches seront le sujet du chapitre suivant.



Chapitre 03 :
Modélisation des processus
métiers

1. Introduction :

L'approche processus repose sur la possibilité de construire une représentation. Celle-ci sert de base pour porter une appréciation sur des processus métiers existants, comme pour concevoir des améliorations ou de nouveaux processus, intégrant ou non la dimension système d'information. Différentes opinions peuvent être exprimées et des options éventuellement négociées sur ces vues. C'est pourquoi la modélisation est un élément clé de la maîtrise des processus et de leur appropriation. [34]

Un modèle de processus métiers (MPM) vous aide à identifier, décrire et décomposer des processus métiers. Vous pouvez analyser votre système à différents niveaux, en mettant l'accent alternativement sur le flux de contrôle (la séquence d'exécution) ou sur le flux de données (l'échange des données). Vous pouvez modéliser, procéder au reverse engineering et générer pour BPEL, BPMN et d'autres langages de processus. [35]

Dans ce chapitre, nous allons d'abord définir un processus d'une manière générale, évoquer les différents types de processus qui existent. Ensuite, nous allons présenter les différentes méthodes pour modéliser un processus métier et enfin la notion BPMN comme un langage de modélisation.

2. Les processus :

2.1. Définition :

- **ISO 9000:2000** : Un **processus** est un ensemble d'activités corrélées ou interactives qui transforment des éléments d'entrée en éléments de sortie. Ces moyens peuvent inclure le personnel, les finances, les installations, les équipements, les techniques et méthodes. [39]

- Un **processus** : est un ensemble d'activités ayant un déclencheur commun, reliées entre elles par des flux d'information ou de matière significatifs et qui se combinent pour fournir un produit matériel ou immatériel, important et bien défini que l'on peut rattacher à un client externe ou interne. [38]

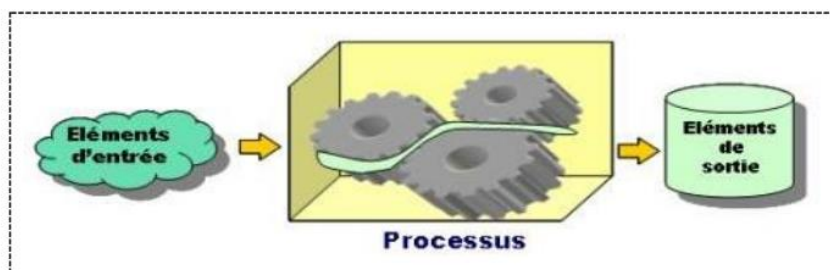


Figure 3.1 : Représentation graphique d'un processus. [38]

2.2. Les types de processus :

Certainement nous pouvons définir 3 catégories de processus comme montre la figure suivante :

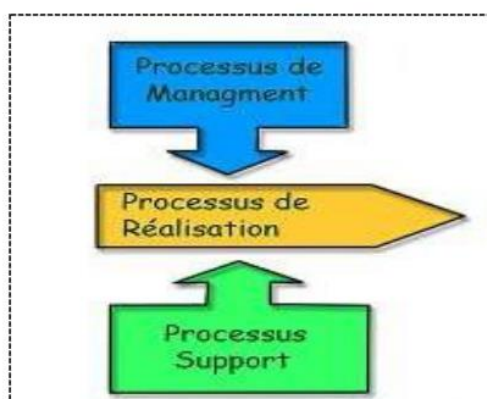


Figure 3.2 : Les types de processus. [W4]

2.2.1. Les processus de support :

Les processus de support (soutien) fournissent les ressources nécessaires au bon fonctionnement de tous les autres processus. Ils ne sont pas liés directement à une contribution de la valeur ajoutée du produit mais sont toujours indispensables. [W4]

2.2.2. Les processus de management :

Aussi appelés de direction, de pilotage, de décision. Ils participent à l'organisation globale, à l'élaboration de la politique, au déploiement des objectifs et à toutes les vérifications indispensables. Ils sont les fils conducteurs de tous les processus de réalisation et de support. [W4]

2.2.3. Les processus de réalisation (métier) :

Les processus de réalisation sont liés au produit, augmentent la valeur ajoutée et contribuent directement à la satisfaction du client.

Ils sont composés d'un enchaînement d'activités ou d'ensembles d'activités, alimentés par des entrées et consomment des ressources, qui créent des sorties en y apportant une valeur ajoutée. [W4]

2.3. L'interaction entre les trois types de processus :

La figure 3.3 ci-dessous symbolise l'interaction entre les trois types de processus vue précédemment :

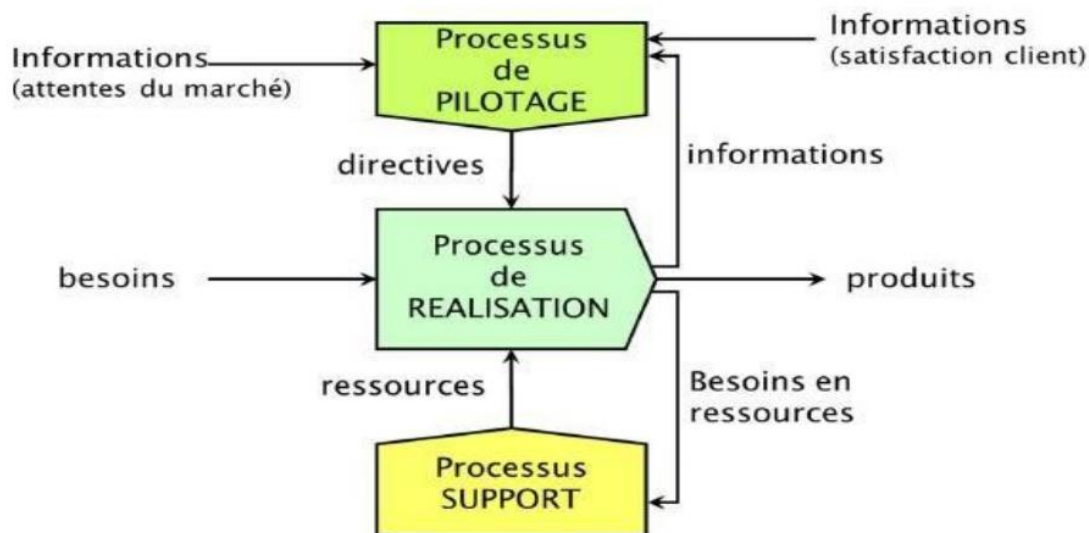


Figure 3.3 : L'interaction entre les trois types de processus. [48]

Comme montre la figure précédente, le processus métier est le responsable à la fourniture des produits, en consommant certain besoin, en utilisant les ressources fournies par le processus du support et sous la direction du processus de pilotage. [40]

2.4. Les différentes méthodes de modélisation :

Dans le monde des entreprises, il existe de nombreuses méthodes qui permettent la modélisation des processus métier. Devant ce large choix nous avons choisi d'en présenter certain :

2.4.1. Business Process Management (BPM) : [37]

Le BPM est défini par une application informatique qui permet l'intégration des données, des gens et des applications à travers un processus métier commun. C'est une approche globale de la gestion des processus d'entreprise. Elle couvre la modélisation, l'informatisation, l'exécution, l'administration et l'optimisation des processus d'entreprises.

Les objectifs des BPM sont : [37]

- Automatiser les processus métier de l'entreprise ;
- Accélérer grandement les étapes de décisions en apportant en temps réel les informations pertinentes aux bonnes personnes ;
- Capitaliser sur les applications du système d'information.

24.2. Process Definition Language (XML):

La définition d'un processus comporte les principaux éléments suivants, sous forme de balises: [37]

- ✓ les marques de début et de fin du ou des processus
- ✓ les activités
- ✓ leurs interrelations (les transitions)
- ✓ les attributs qualifiant certains comportements de l'activité
- ✓ les participants / rôles / groupes
- ✓ les interactions / relations entre les acteurs et les activités

24.3. Business Process Execution Language (BPEL):

BPEL est un langage exécutable sur des moteurs BPEL comme ODE (Orchestration Director Engine) d'Apache. Le langage BPEL – Business Process Execution Language – a été lancé à l'initiative de Microsoft et IBM en réponse à l'initiative de BPML. Depuis, ce langage a reçu le support de la plupart des acteurs du marché. BPEL est devenu le standard. Il vient en complément de la spécification sur les services Web. Depuis 2003, c'est l'organisme de standardisation OASIS qui a en charge l'évolution du langage BPEL.

Il est possible de modéliser un "Business Process" en utilisant BPMN et de le convertir en BPEL, puis de déployer le BPEL sur un moteur BPEL. [37]

3. Business Process Modeling Notation (BPMN):

BPMN (Business Process Model and Notation) est une notation pour la représentation graphique des processus métier dans un workflow. BPMN est un langage récent conçu pour être au cœur d'une approche de modélisation et d'implémentation utilisant les cadres

conceptuels de l'ingénierie dirigée par les modèles (IDM), de l'architecture orientée services (SOA) et de la gestion des processus métier (BPM). [41]

L'ensemble des éléments de modélisation constituant la notation BPMN sont nombreux et constituent des variantes des éléments de base que nous allons présenter. Ces éléments peuvent être catégorisés en quatre groupes, comme présentés dans la Figure 3.4. [42]

BPMN est une notation de modélisation largement utilisée par les analystes métier, les développeurs informatiques et les professionnels qui gèrent les processus métier. [w11]

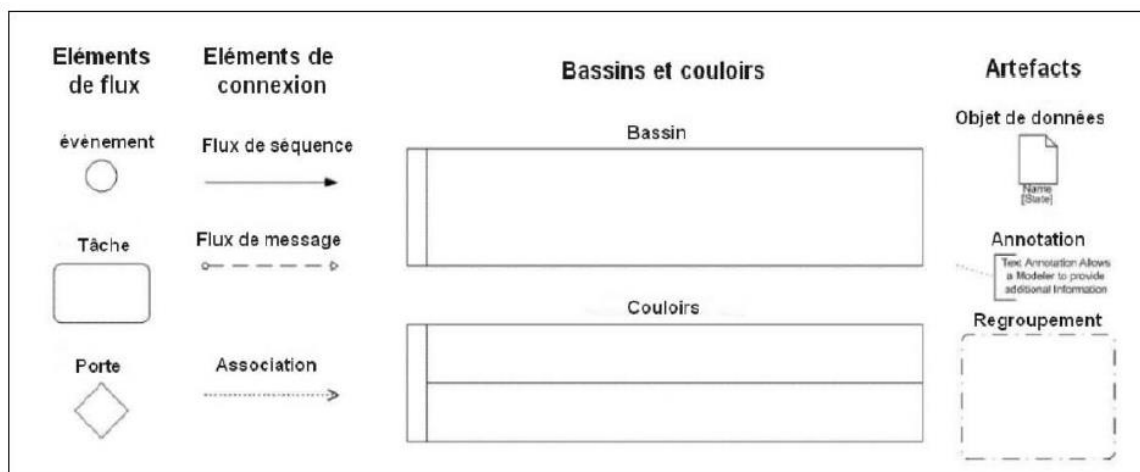


Figure 3.4 – Éléments de base de la notation BPMN [41]

3.1. Objectifs de BPMN :

L'objectif principal de BPMN est de proposer une notation standard facilement compréhensible par les partenaires professionnels. Ces partenaires incluent les analystes métiers qui créent et raffinent les processus, les développeurs techniques responsables de l'implémentation des processus, et les directeurs commerciaux qui suivent et gèrent les processus. Par conséquent, BPMN est prévu pour servir comme langage visant à combler un déficit de communication qui survient souvent entre le design des processus métier et l'implémentation. [37]

3.2. Les principaux éléments graphiques de BPMN :

Les éléments sont organisés en six catégories. L'objectif est de réaliser un regroupement ne trahissant pas la nature des éléments graphiques (leur définition et leur emploi selon la norme OMG) mais faisant sens pour la conception d'un scénario pédagogique. [41]

3.3. Les outils de modélisation des processus :

Les outils de modélisation des processus sont de plus en plus nombreux sur le marché en raison de l'augmentation des besoins d'automatisation des processus métiers. A l'aide de ces outils, les entreprises d'aujourd'hui peuvent plus aisément créer des modèles de leurs processus et suivre leur exécution. [40]

La figure suivante montre les outils les plus utilisés dans le domaine de modélisation des processus :



Figure 3.6 : Les outils de modélisation des processus.

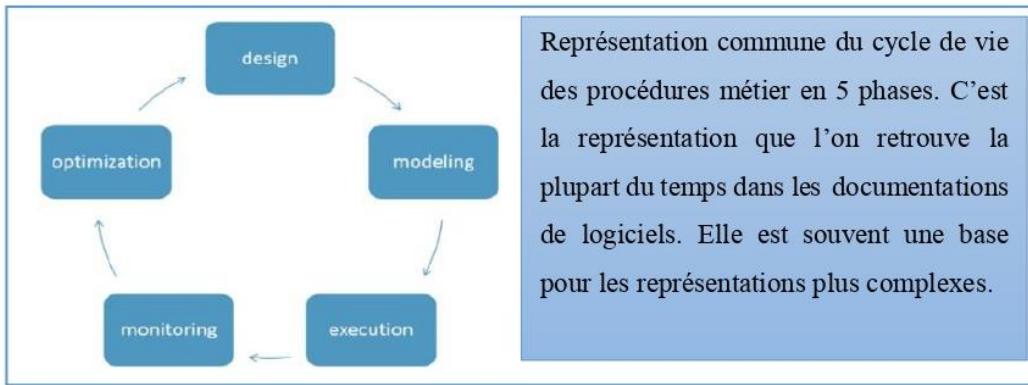
4. DE L'APPROCHE BPM AU STANDARD BPMN 2.0 :

L'approche BPM trouve son origine dans les travaux sur le « Workflow », apparu avec l'industrialisation du secteur tertiaire, puis considérablement développé dans les années 90 avec leur gestion par systèmes informatiques (Ulmer, 2011). Un Workflow peut être défini selon la Workflow Management Coalition (1999) comme « l'automatisation des procédures métiers²¹, en partie ou totalité, durant laquelle les documents, informations et tâches sont transférés d'un participant à un autre selon un système de règles établies » [41]

4.1. Cycle de vie d'un processus :

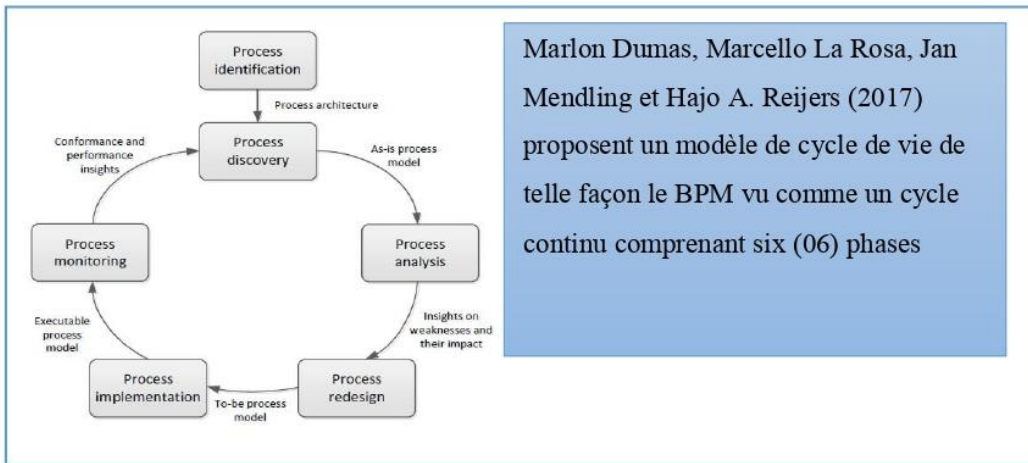
L'approche BPM définit toujours le cycle de vie d'un processus métier comme une boucle continue. L'essence d'une approche BPM est le principe de réingénierie. La modélisation d'un processus incluant l'activité humaine est souvent imparfaite mais doit pouvoir être évaluée et améliorée continuellement pour parvenir à la gestion optimale des activités. [41]

Nous présentons ce qui suit quelques représentations de ce cycle de vie :



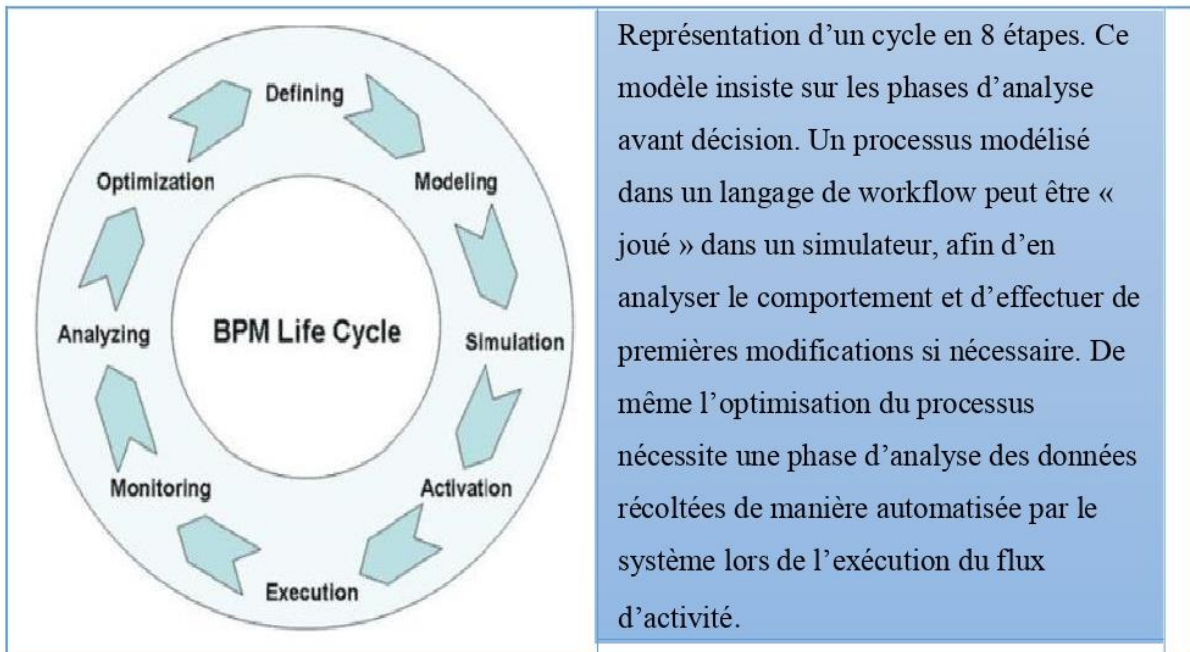
Représentation commune du cycle de vie des procédures métier en 5 phases. C'est la représentation que l'on retrouve la plupart du temps dans les documentations de logiciels. Elle est souvent une base pour les représentations plus complexes.

Figure 3.7 : Représentation commune d'un cycle de vie des procédures métier. [41]



Marlon Dumas, Marcello La Rosa, Jan Mendling et Hajo A. Reijers (2017) proposent un modèle de cycle de vie de telle façon le BPM vu comme un cycle continu comprenant six (06) phases

Figure 3.8 : BPM – cycle de vie des processus métier. [51]



Représentation d'un cycle en 8 étapes. Ce modèle insiste sur les phases d'analyse avant décision. Un processus modélisé dans un langage de workflow peut être « joué » dans un simulateur, afin d'en analyser le comportement et d'effectuer de premières modifications si nécessaire. De même l'optimisation du processus nécessite une phase d'analyse des données récoltées de manière automatisée par le système lors de l'exécution du flux d'activité.

Figure 3.9 : Cycle de vie des procédures métier. [49]

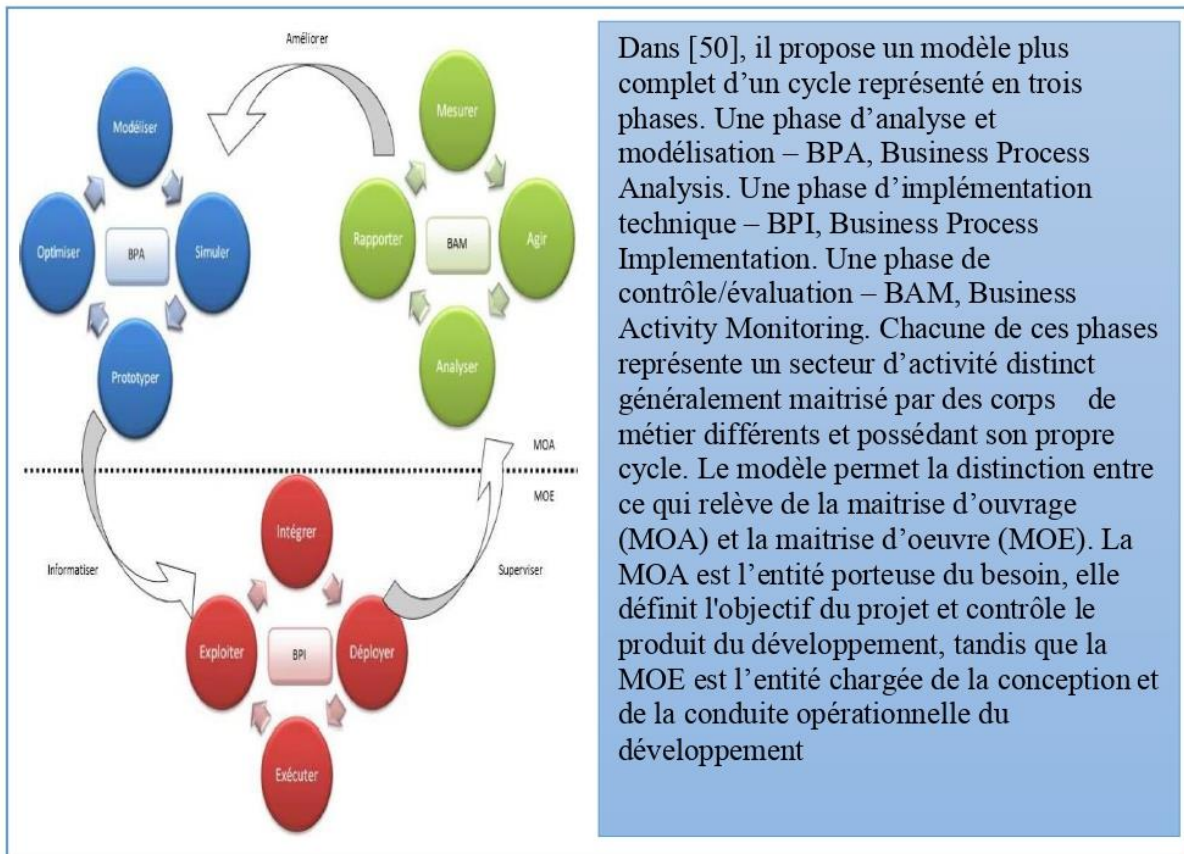


Figure 3.10 : BPM - cycle de vie des processus métier. [50]

5. BPM et SOA : concurrence ou synergie ?

Plusieurs travaux ont montré des synergies entre BPM et SOA. L'utilisation de SOA permet de découvrir des services réutilisables pouvant être orchestrés pour mettre en place des processus métiers dynamiques. Cette combinaison assure la conception itérative des processus basés sur des services qui pourront être changés rapidement [36].

La combinaison de BPM et SOA crée des opportunités en rendant visibles les processus de l'entreprise et en l'aidant à développer une infrastructure IT flexible. Cependant, l'alignement de ces deux concepts exige une profonde transformation des organisations [36].

D'une part, la démarche BPM ne fournit pas d'unité élémentaire de granularité suffisamment fine pour construire un système, alors que la notion de service offre un cadre d'architecture pour la conception de processus métiers en assurant la répartition des capacités de l'entreprise.

D'autre part, SOA peut produire des services réutilisables, mais sans garantie d'agilité métier [36].

Le but de SOA est de découvrir des services stables et réutilisables alors que les processus découverts avec BPM sont centrés sur les besoins du client (amenés à changer pour garantir l'agilité de l'entreprise). SOA promeut la consolidation des opérations redondantes et améliore la capacité d'adaptation aux changements métiers. [36]

Pour comprendre la synergie SOA-BPM, il faut répondre à une question essentielle à savoir : *que représente un processus métier dans SOA ?* Deux visions s'opposent. La première, souvent utilisée dans des approches techniques de SOA, positionne les processus métiers au-dessus des services métiers. La deuxième explique qu'un processus métier qui invoque un service fait lui-même partie d'un autre service métier. [36]

BPM et SOA offrent une combinaison parfaite pour l'informatique d'entreprise. BPM fournit l'abstraction de plus haut niveau pour définir les processus métier, ainsi que d'autres capacités importantes de surveillance et de gestion de ces processus. Les services fournissent les fonctions qui prennent en charge ces processus. SOA fournit les capacités pour les services à combiner et pour soutenir et créer une entreprise agile et flexible. Le BPM sans SOA est utile pour créer des applications, mais difficile à étendre à l'entreprise. La SOA sans BPM est utile pour créer des services réutilisables et cohérents, mais n'a pas la capacité de transformer ces services en une entreprise agile et compétitive. [44]

Nous savons tous que les processus s'exécutent sur des services, de sorte que l'architecture orientée services et la gestion des processus métier appartiennent clairement les uns aux autres.

Nous essayons à travers le tableau 3.1 de dresser une comparaison entre BPM et SOA :

N°	BPM	SOA
1	Il fournit des solutions opérationnelles qui sont essentielles pour les analystes métier et processus.	C'est un style architectural important pour les architectes et les ingénieurs.
2	Il utilise les services Web réutilisables pour fournir les capacités de l'entreprise.	Il est également identique au BPM qui fournit les fonctionnalités d'entreprise en fonction des services Web réutilisables.
3	Il utilise des services pour faire ressortir leurs objectifs.	Il contient des services qui sont mis en œuvre avec des processus métier pouvant être modifiés rapidement en fonction des changements commerciaux.
4	Il spécifie comme organisation des capacités de l'entreprise telles que les personnes, les processus, la technologie et les données.	Il décrit comme organisation des capacités techniques telles que des activités, des événements, des documents et des données.
5	C'est une règle commerciale qui pointe vers l'amélioration des opérations.	C'est un style architectural qui pointe vers les systèmes de l'entreprise.

Tableau 3.1 : comparaison entre SOA et BPM

6. Conclusion :

Dans ce chapitre, nous avons vu que le BPMN permet de modéliser les services métiers sous forme de processus, ce qui rend la transition vers des services de la SOA plus simple, Dans notre cas nous allons utiliser cette notation de l'approche processus comme point de départ pour identifier les services potentiels qui peuvent être programmés et faciliter le passage vers une architecture orientée service en se basant sur le respect des caractéristiques de cette dernière (SOA).

Dans le chapitre suivant nous allons détailler la problématique et les objectifs de notre travail ainsi que les solutions proposées.



Chapitre 04 :
Conception

I. Introduction :

La SOA (architecture orientée services) a changé la méthode de développement de système conventionnelle et a permis à l'intégration des systèmes de devenir plus flexible. La réduction des coûts est aussi un enjeu majeur dans la construction des systèmes que la SOA essaye de prendre en charge. Les plateformes d'apprentissages peuvent aussi bénéficier des avantages de la SOA puisque les composants logiciels de la SOA et des services Web se caractérisent par le fait qu'ils sont réutilisables et interchangeable, et sont donc capables de réduire le gaspillage de ressources pédagogiques, ainsi que les coûts de développement du système. [27]

Dans notre travail, nous nous sommes appuyés sur l'approche TOP-DOWN pour la réingénierie d'une plateforme d'apprentissage en ligne classique vers une plateforme orientée service. Nous avons démarré d'une description générale des fonctionnalités existantes dans une plateforme e-learning jusqu'à l'arrivée à une plateforme moderne qui respecte les propriétés d'une architecture orientée service (SOA).

II. Objectif du projet :

Notre objectif principal est d'automatiser la phase d'identification des services d'une plateforme e-learning classique, afin de faciliter sa migration vers une nouvelle architecture orientée service. Cette étape d'identification de services est incontournable dans n'importe quelle approche de mise en œuvre d'une plateforme orientée service. Plusieurs démarches existent, nous avons choisi de suivre une approche top-down, qui consiste à démarrer l'identification des services à partir d'une analyse et description générale des fonctionnalités indépendantes existantes dans une plateforme e-learning classique, puis nous avons modélisé ces fonctionnalités sous forme de processus métiers (une forme qui facilite le passage vers une architecture orientée service). Nous avons choisi les diagrammes BPMN pour représenter les processus métiers sous la norme BPMN 2.0.

Pour atteindre notre but nous avons suivi les étapes suivantes :

- Nous avons fait une analyse des plateformes d'apprentissage en ligne existantes afin d'extraire toutes les fonctionnalités indépendantes candidates à la réutilisation.
- L'étape suivante consiste à modéliser les fonctionnalités indépendantes sous forme de processus métiers. Nous avons opté pour la norme BPMN comme langage formel pour l'opération de modélisation.

- Afin d'identifier des services qui soient conformes aux exigences de la SOA, nous avons proposé un ensemble de règles qui nous permettent de choisir les tâches qui respectent les caractéristiques de la SOA tels que l'autonomie, la réutilisabilité, le couplage lâche, la cohésion forte ...etc.
- Nous avons développé un Framework (ensemble de composants logiciels cohérents) qui permet d'automatiser l'identification des services en appliquant les règles prédéfinis par défaut, mais qui permet aussi d'introduire de nouvelles règles qui peuvent être enrichies au fur et à mesure par les experts.
- Une évaluation des services candidats identifiés a été proposée à travers une évaluation des règles d'identification proposée sous forme d'un taux de confiance introduit par les experts.

III. Architecture générale de notre système :

Nous détaillons dans ce qui suit les étapes que nous avons suivi afin d'identifier les services d'une plateforme e-learning. Dans notre cas une tâche ou bien un ensemble de tâches automatiques, peuvent former un service à condition de respecter les propriétés d'un service.



Figure 4.1 : les étapes principales pour la réingénierie d'une plateforme e-learning existante vers une plateforme orientée service

❖ Phase 1 : Analyse des plateformes E-learning classique et extraction des fonctionnalités indépendantes :

Dans cette étape nous avons fait une analyse des plateformes e-learning existantes afin de dégager les fonctionnalités de base communes, puis nous avons gardé seulement les fonctionnalités indépendantes qui respectent les propriétés et caractéristiques

suivantes : autonomie, réutilisabilité, couplage faible avec l'extérieur, cohérence, interopérabilité ...etc.

L'analyse des plateformes e-learning nous a permis d'identifier et de mettre en évidence les modules autonomes (qui sont prédisposés à être partagés par les systèmes e-learning). Nous avons gardé six (06) fonctionnalités de base qui respectent les exigences citées ci-dessus et qui sont :

- ✓ Fonctionnalité d'aide au tutorat
- ✓ Fonctionnalité de gestion des cours
- ✓ Fonctionnalité de Test de Niveau
- ✓ Fonctionnalité de gestion de Bibliothèque
- ✓ Fonctionnalité de collaboration
- ✓ Fonctionnalité de regroupement des apprenants

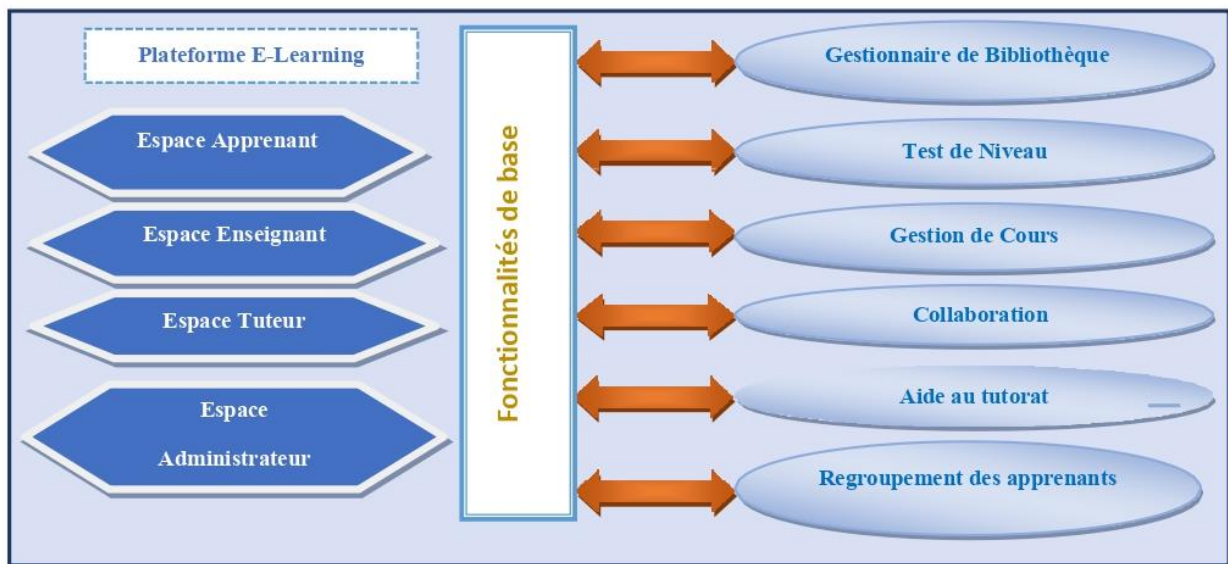


Figure 4.2 : Fonctionnalités de bases choisies après l'analyse.

1. Fonctionnalité gestion de bibliothèque :

La Bibliothèque à distance est une collection des documents numériques accessibles à distance dans toutes les disciplines enseignées à l'université, elle est adaptée aux besoins de la communauté universitaire grâce à sa propre base de données.

2. Fonctionnalité test niveau :

Cette fonctionnalité est développée par des pédagogues et des chercheurs dans le domaine de la psychopédagogie et des linguistiques. Elle comporte quatre types de test : test cognitif, test de personnalité, test d'intelligence et test psychotechnique. Après avoir récupéré les questions du test et passer le test par l'apprenant, le module ne communique les résultats finaux à l'étudiant et aussi au système pour faire la mise à jour sur la BDD profil. Ces résultats sont utilisés par la suite dans le regroupement des apprenants et pour le tuteur.

3. Fonctionnalité gestion des cours :

La fonctionnalité gestion des cours permet d'une part l'accès au cours locaux ainsi que le partage des contenus pédagogiques distants de façon à ce qu'ils soient accessibles par une multitude de plateformes de formation. Elle prend en charge aussi le suivi de la progression de l'apprenant dans le cours à travers des tests d'évaluation à la fin de chaque chapitre puis la mise à jour de la BDD profil de l'apprenant. Pour récupérer un chapitre, on a besoin de l'identifiant de l'apprenant de son niveau cognitif et de son état de progression dans le cours. De la même façon, si on veut récupérer les exercices, alors on a besoin du code du chapitre concerné.

4. Fonctionnalité collaboration :

Cette fonctionnalité est une plateforme collaborative qui permet à des apprenants du même groupe de collaborer ensemble pour discuter et implémenter la solution d'un travail donné en ligne et permet aussi de capturer des statistiques et des traces sur les activités collaboratives de chaque apprenant. Ces derniers seront transmis pour mettre à jour la BDD profil pour être exploités par le tuteur ultérieurement.

5. Fonctionnalité aide au tutorat :

La fonctionnalité aide au tutorat est un module qui fournit un ensemble de fonctionnalités qui peuvent être exploitées par différentes plateformes d'apprentissage. Elle permet de fournir des statistiques sur le comportement de l'apprenant et sur toutes ces activités dans la plateforme d'apprentissage, allant des avancements dans les cours, la collaboration avec ces paires, la communication avec les différents acteurs, la participation aux réunions etc... Toutes ces informations aident le tuteur à mieux cerner l'avancement de l'apprenant afin d'intervenir et assister ce dernier.

6. Fonctionnalité de regroupement des apprenants :

C'est un module intelligent qui inclut des stratégies de regroupement des apprenants, en exploitant les traces de toutes les interactions où l'apprenant intervient depuis l'inscription et le test de niveau.

Il existe plusieurs méthodes de regroupement, dans notre cas on a mentionné trois types de regroupement à savoir : regroupement par profil, regroupement par choix et regroupement par algorithme génétique. Les méthodes de regroupement peuvent être étendues en cas de besoin.

❖ Phase 2 : Modélisation des fonctionnalités de base.

C'est la phase de modélisation des fonctionnalités de base sous forme de processus métier. Nous avons choisi le standard BPMN (Business Process Model and Notation) qui est la dernière version du standard BPMN développé par l'Object Management Group (OMG). Il simplifie les tâches de l'utilisateur en proposant un diagramme logiciel clair.

Dans cette phase nous avons proposé des diagrammes pour les six (06) fonctionnalités de base choisies dans la phase 1.

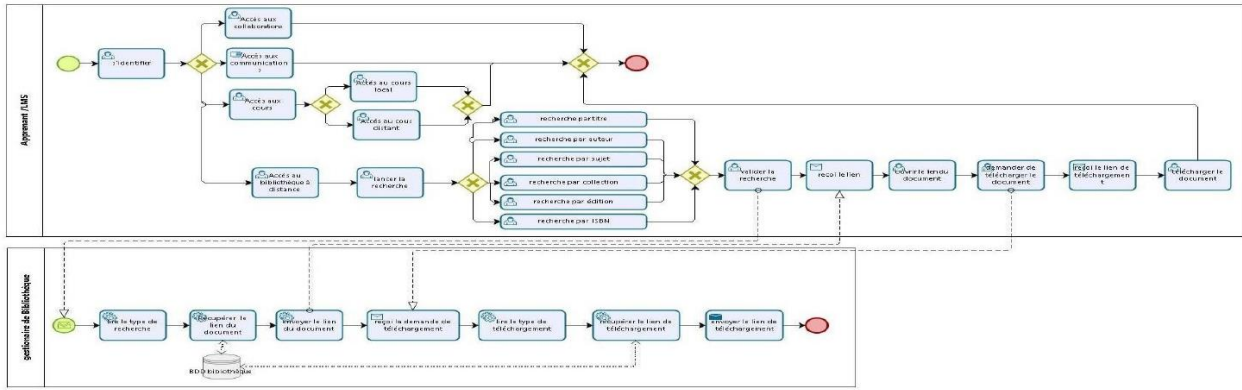


Figure 4.3 : Représentation BPMN Fonctionnalité bibliothèque à distance.

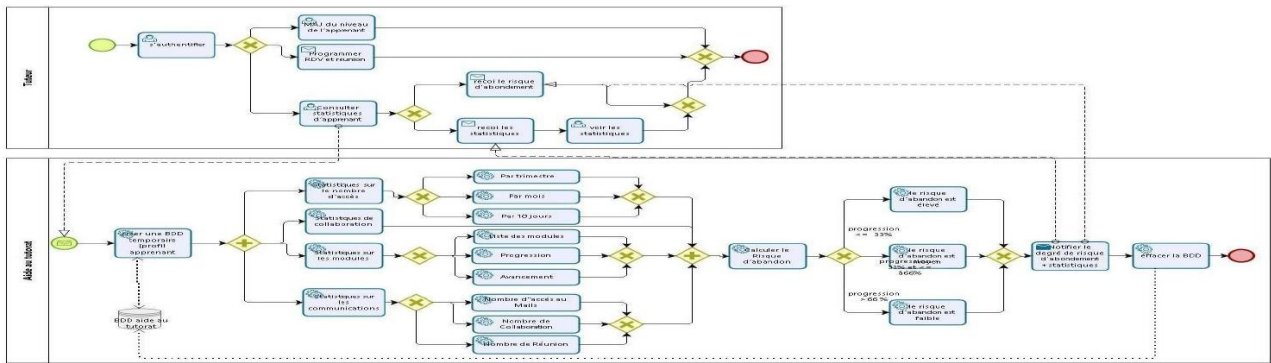


Figure 4.4 : Représentation BPMN fonctionnalités aide au tutorat.

44 Réingénierie d'une plateforme d'apprentissage existante vers une plateforme orientée services

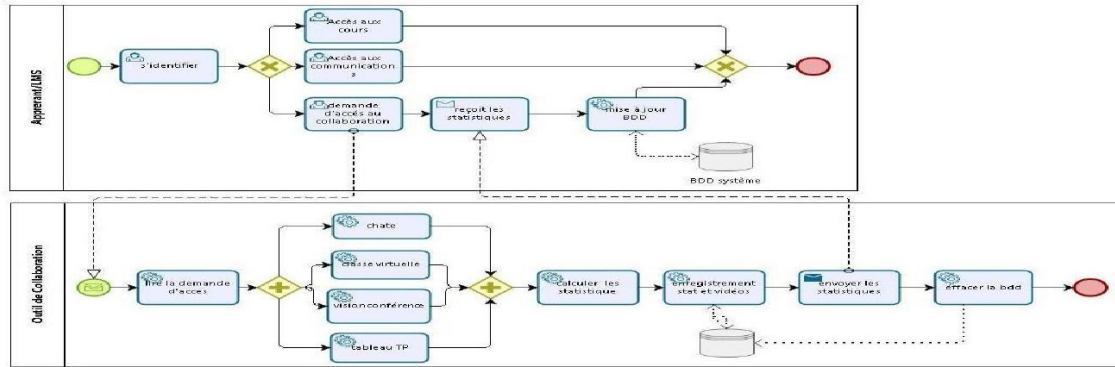


Figure 4.5 : Représentation BPMN fonctionnalité de base Collaboration

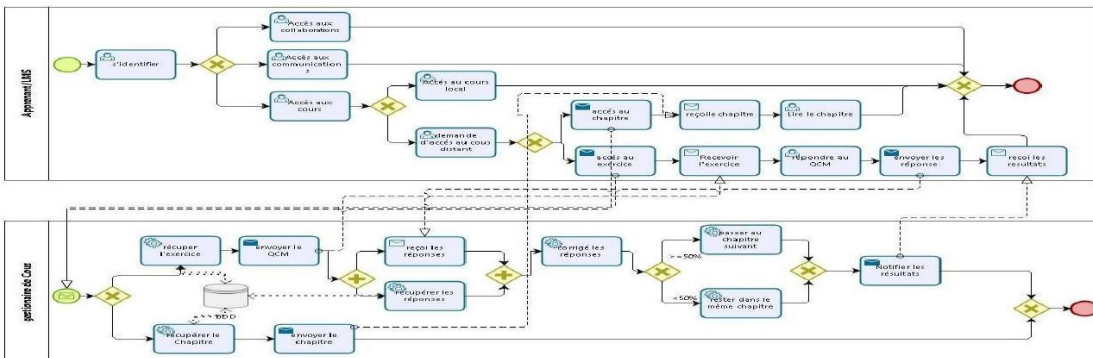


Figure 4.6 : Représentation BPMN fonctionnalité de base Cours

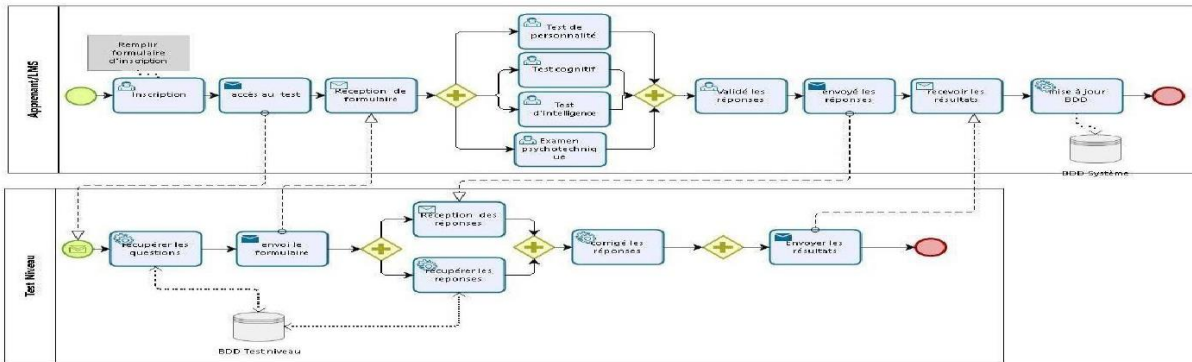


Figure 4.7 : Représentation BPMN fonctionnalité Test niveau.

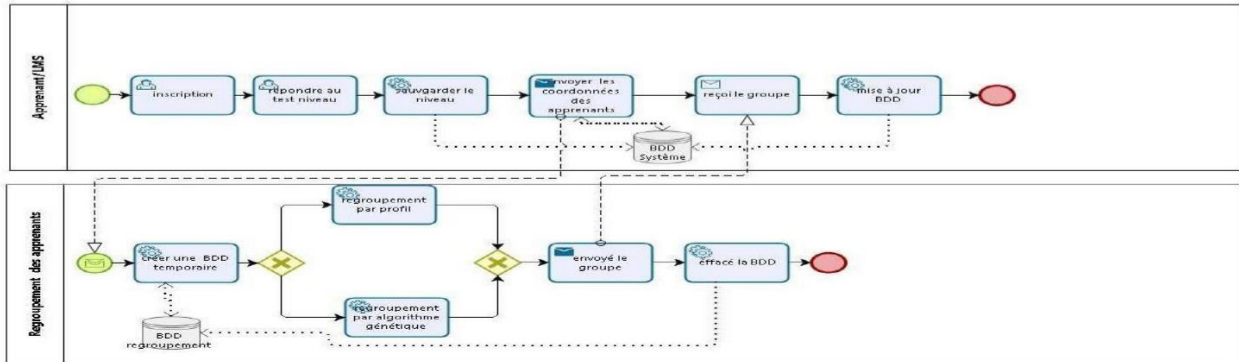


Figure 4.8 : Représentation BPMN fonctionnalité regroupement des apprenants.

❖ Phase 3 : exportation vers XPDL (XML Process Definition Language) :

Le but de cette étape est de convertir la représentation graphique (BPMN) des processus métiers vers une représentation textuelle dans un langage formel. Cela permet la lecture automatique et l'analyse des processus métier afin d'identifier les services candidats. Le format destinataire est le XPDL (XML Process Definition Language), c'est une norme de format ouvert et extensible du consortium Workflow Management Coalition qui permet de définir des processus d'affaires à l'aide du langage XML et de les mettre en œuvre avec un moteur de workflow. XPDL est compatible avec la norme BPMN qui permet de représenter graphiquement les workflows, et dont XPDL est un format d'échange. [W6]

❖ Phase 4 : Automatisation de l'identification des services web

Dans cette phase on charge les fichiers XPDL dans notre Framework, le processus d'identification automatique des services web se déroule à ce stade et cela sur plusieurs axes (voir figure), à savoir

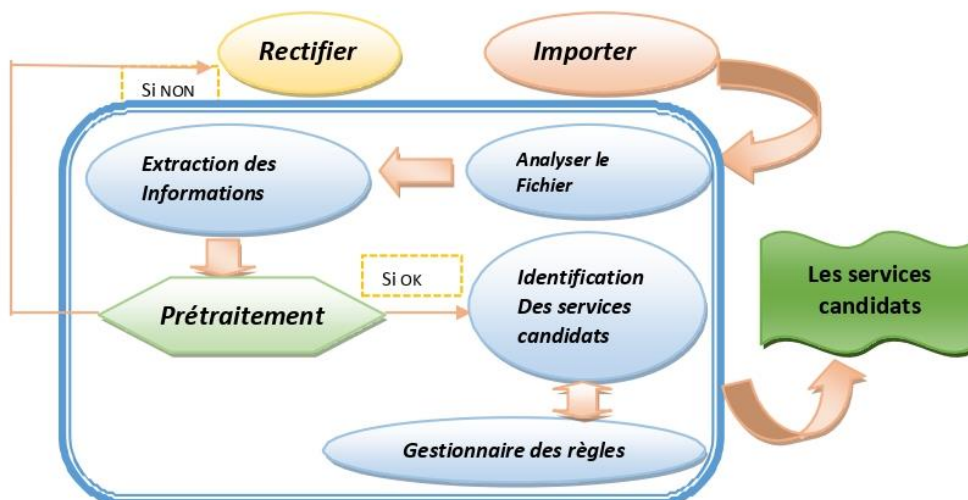


Figure 4.9 : Processus d'Automatisation de l'identification des services.

1. Le prétraitement des diagrammes BPMN :

Cette étape consiste à vérifier que le diagramme BPMN soit bien formulé et que toutes les informations nécessaires pour l'identification des services soient présentes, en particulier toutes les tâches doivent être renseignées par l'information type de tâche. Si les informations manquent, le traitement sera arrêté et l'utilisateur sera invité à compléter le diagramme.

2. L'analyse des diagrammes BPMN :

L'étape d'analyse permet d'extraire les informations d'une représentation textuelle du diagramme BPMN qui est le format XPDL. Ces informations concernent les activités qui composent un processus. Cela nécessite un parcours de la totalité du processus pour atteindre ces derniers. A partir de la représentation du processus, nous parcourons tous les éléments et les informations utiles dans les instructions simples (de type de tâche) et composées (les branchements, les sous-processus) et d'autres éléments significatifs comme l'échange des messages à travers les flux de messages et aussi l'envoi et la réception des messages, les services identifiés peuvent être des services composites qui orchestrent des services atomiques pour offrir des fonctionnalités plus avancées ou des services atomiques qui peuvent fournir des fonctionnalités par eux-mêmes.

En effet les propriétés des services identifiés sont liées à la réutilisabilité, l'autonomie, la composition. Ces mêmes propriétés sont exigées pour la SOA.

3. Règles d'identification des services :

Afin de détecter et d'identifier les services candidats, nous avons proposé un certain nombre des règles qui nous permettent de regrouper une ou plusieurs activités de la représentation BPMN sous forme de services.

Pour fixer ces règles, nous nous sommes inspiré du travail [19] puis elles ont été adaptées suivant les diagrammes BPMN des fonctionnalités de base indépendantes qu'on a choisies et à partir aussi des caractéristiques que doit respecter un service dans une architecture SOA, à savoir garantir une autonomie, réutilisabilité, compossibilité, couplage faible avec son environnement extérieur et une cohésion forte avec les composants internes du service.

Règle 1 : Si l'ensemble des tâches dans un processus sont toutes des tâches automatiques alors on peut considérer tout le processus comme un service.

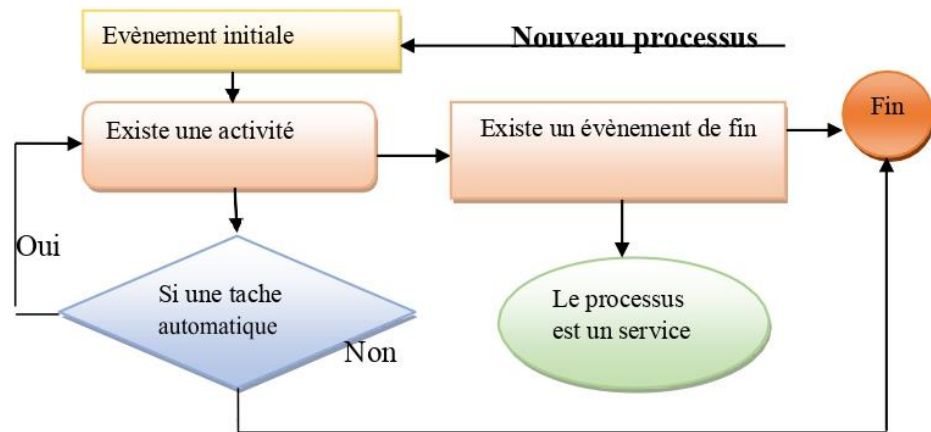


Figure 4.10 : Diagramme illustratif de la règle 1

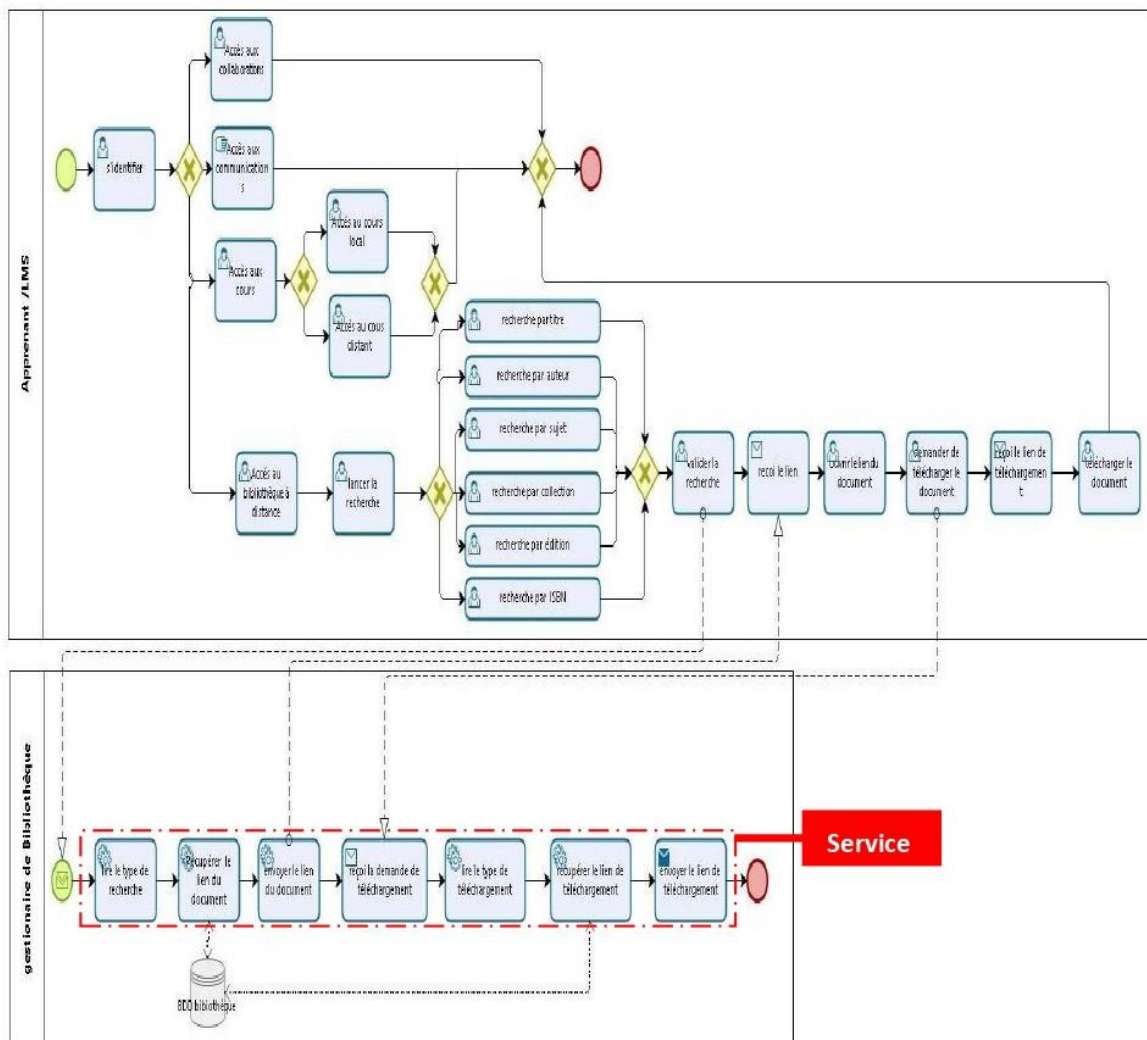


Figure 4.11 : Exemple illustratif de la règle 01.

Règle 2 : Un service est une tâche ou un ensemble de tâches successives entre la réception et l’envoi de message vers le même processus demandeur de traitement.

Si A et B deux processus qui échangent des messages tout en respectant les conditions suivantes :

- A demande un traitement à B.
- B envoie un résultat à A.
- La ou les tâches qui sont automatiques ou interactifs de B et qui se trouvent entre la réception et la réponse peuvent être considérées comme service.

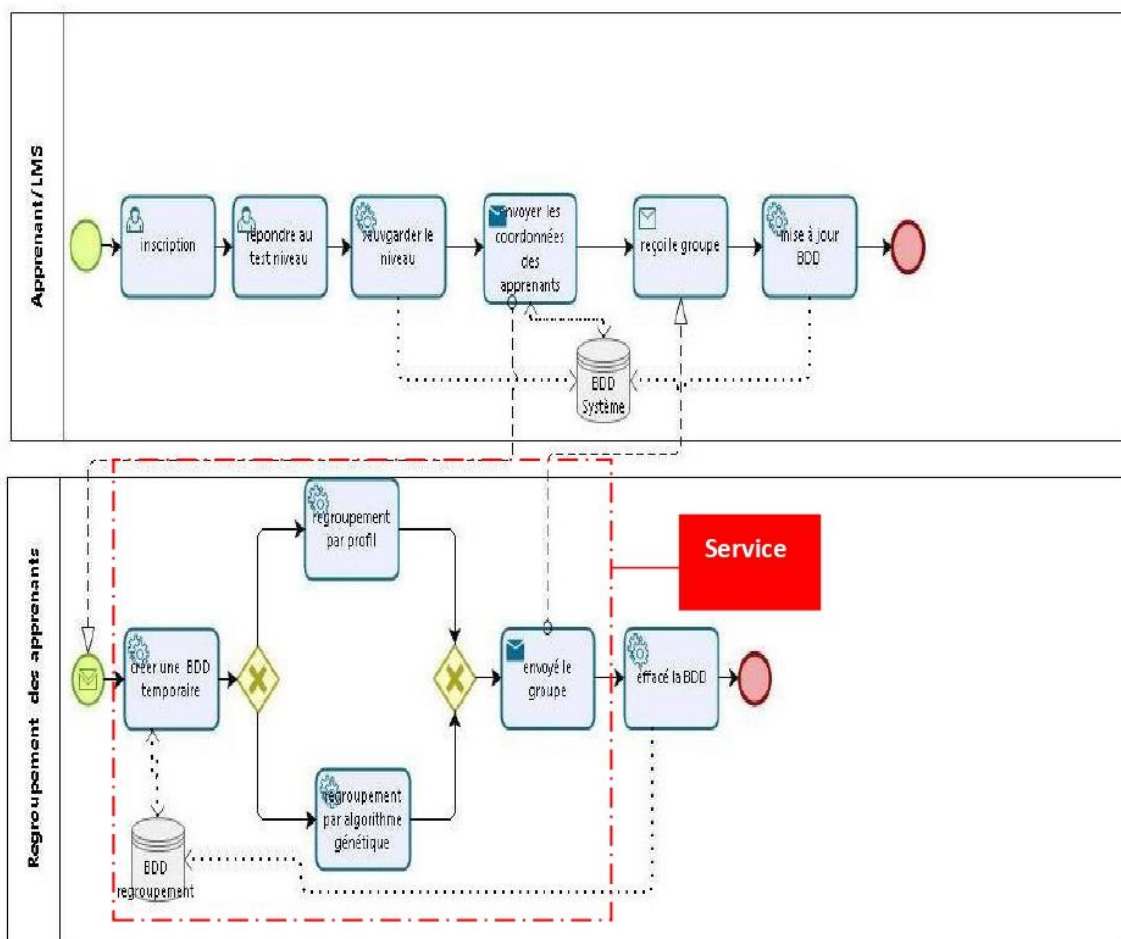


Figure 4.12 : Exemple illustratif de la règle 02

Règle 3 : un service candidat peut être identifié à partir d'une séquence de tâches automatiques qui a comme bornes les événements début, fin, branchement ou une tâche non automatique.

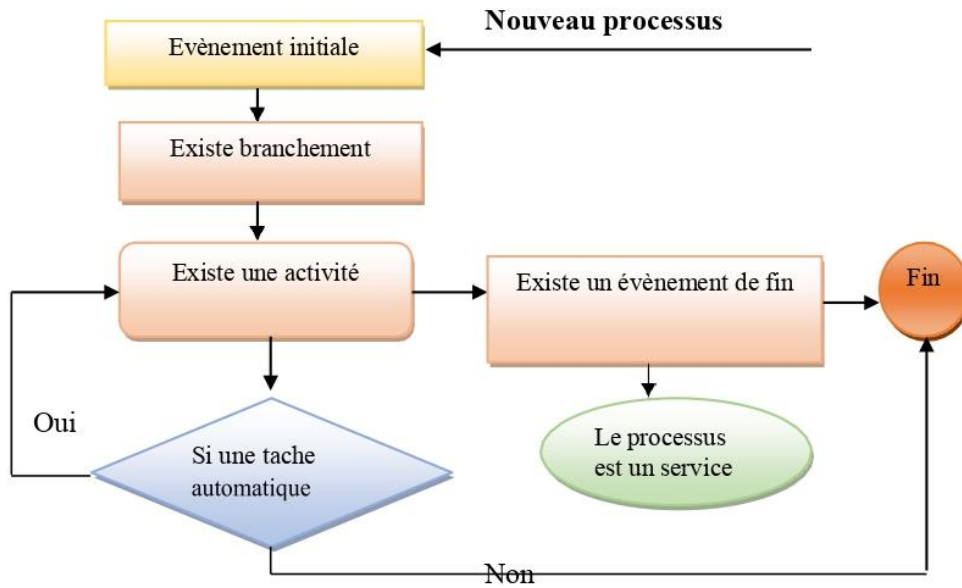


Figure 4.13 : Diagramme illustratif de la règle 3

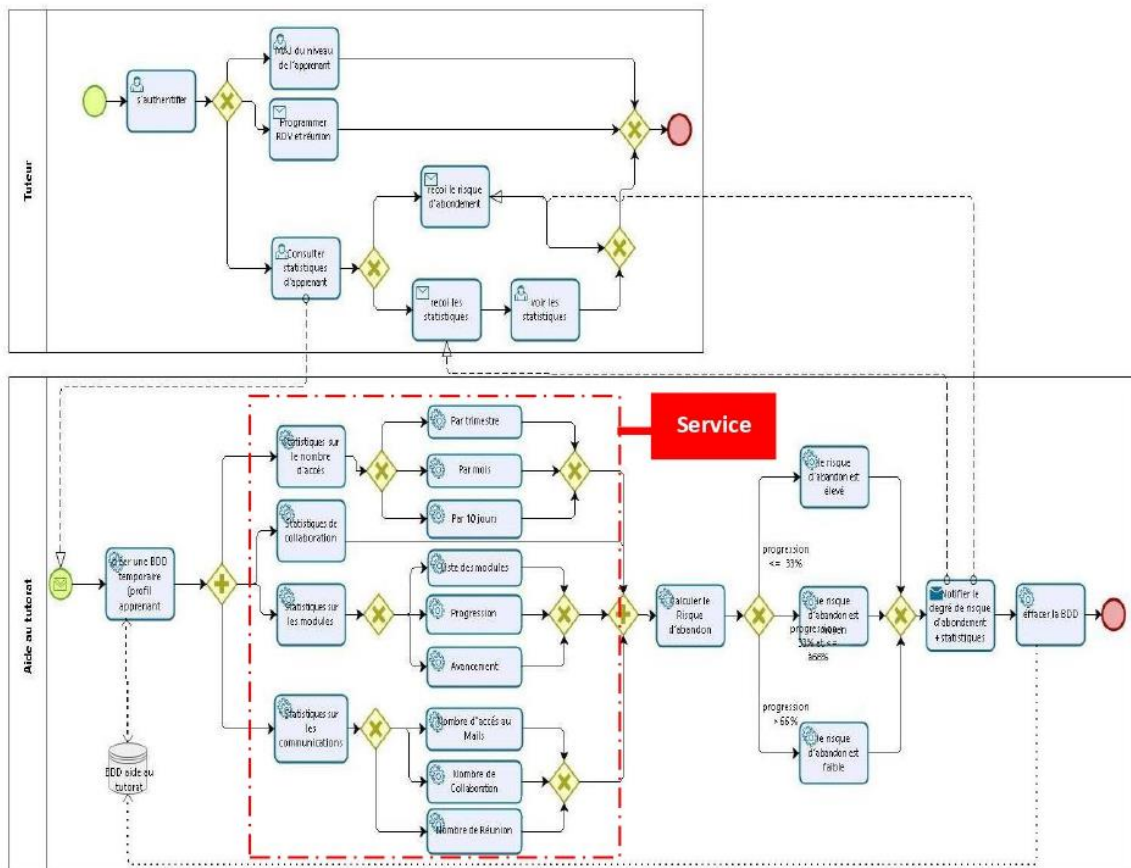


Figure 4.14 : Exemple illustratif de la règle 03

❖ Phase 5 : Validation des services candidats :

La validation des services candidats est une étape très important dans le processus d'identification des services, dans notre PFE, on comptait sur une méthode qui se base sur l'évaluation par les experts.

Cette évaluation est complètement empirique, nous donnons la main aux experts du domaine d'affecter à chaque règle un taux de confiance, qui exprime le niveau de confiance donné à une règle pour identifier les services.

Ce taux de confiance est affecté automatiquement aux services identifié à partir de la règle, et sa valeur est comprise entre 0 et 1.

IV. Conclusion :

Dans ce chapitre nous avons une solution basée sur l'approche TOP-DOWN pour notre problématique qui consiste à identifier automatiquement des services web dans une plateforme d'apprentissage en ligne classique pour la réingénierie la vers une plateforme basée sur le service, en fonction du langage de modélisation d'un processus métier BPMN 2.0

Les détails techniques de l'implémentation feront l'objet du chapitre suivant.



Chapitre 05 :
Implémentation

1. Introduction :

Dans ce chapitre, nous allons présenter les détails d'implémentation de notre projet de fin d'étude. Où nous allons présenter la phase de réalisation de notre Framework qui consiste à automatiser l'identification des services dans une plateforme E-learning classique à partir d'une représentation BPMN2 d'un processus métiers de ces fonctionnalités pour la réingénierie vers une plateforme moderne en fonction de l'architecture orientée service (SOA).

Ainsi seront présentés les outils de développement et les différents composants de notre Framework.

2. Les outils de développement :

2.1. Java :

C'est un langage de programmation orienté objet, développé par Sun Microsystems en 1995. C'est un langage portable qui ne dépend pas d'une plateforme donnée. Il peut être utilisé sous n'importe quel système d'exploitation à savoir Windows, macintosh, linux, etc. Java est donc un langage multiplateforme qui permet aux développeurs d'écrire un code qu'ils peuvent exécuter sur diverses plateformes. Java est rapide, sécurisé et fiable.

2.2. La plateforme Eclipse :

Eclipse est un environnement de développement intégré (Integrated Development Environment) dont le but est de fournir une plate-forme modulaire pour permettre de réaliser des développements informatiques.

2.3. Bizagi Modeler :

C'est une application bien appréciée et avec licence Gratuite pour Windows, c'est l'outil de modélisation avec le meilleur environnement permettant de représenter graphiquement les activités et les processus, Il permet de créer des diagrammes BPMN [w12].

2.4. XPD (XML Processing Description Language) :

L'entrée de notre application est un fichier XPD, XPD est une Norme du WfMC (Workflow Management Coalition) destinée à décrire les processus métier à l'aide du langage de balisage XML (Extensible Markup Language). Avec XPD, un produit peut écrire une définition de processus avec une pleine fidélité, et un autre produit peut lire et reproduire le même schéma qui a été envoyé.

2.5. JDOM (Java Document Object Model) :

C'est une bibliothèque open source pour manipulation des fichiers XML en Java. Nous avons utilisé JDOM comme un package dans notre application pour analyser le fichier XML. JDOM n'est pas un analyseur XML, C'est un modèle objet de document qui utilise des analyseurs XML pour créer des documents, de naviguer dans leur structure, d'ajouter, de modifier, ou de supprimer leur contenu.

3. L'Architecture globale de notre plateforme :

L'architecture de notre Framework est composée de trois (03) modules et ils sont :

- **Module d'analyse** : il est responsable de vérifier la bonne structuration du fichier (XPDL) à analyser et ensuite la récupération de toutes ces informations nécessaires pour l'identification.
- **Module de gestion des règles** : contient des règles d'identification des services à partir des diagrammes BPMN, il est flexible car il intègre un générateur de règles qui permet d'introduire les nouvelles règles.
- **Module d'identification** : qui travaille en coordination avec les deux (02) modules cités précédemment afin d'identifier les services candidats en appliquant les différentes règles sur les différentes listes issues de l'étape d'analyse.

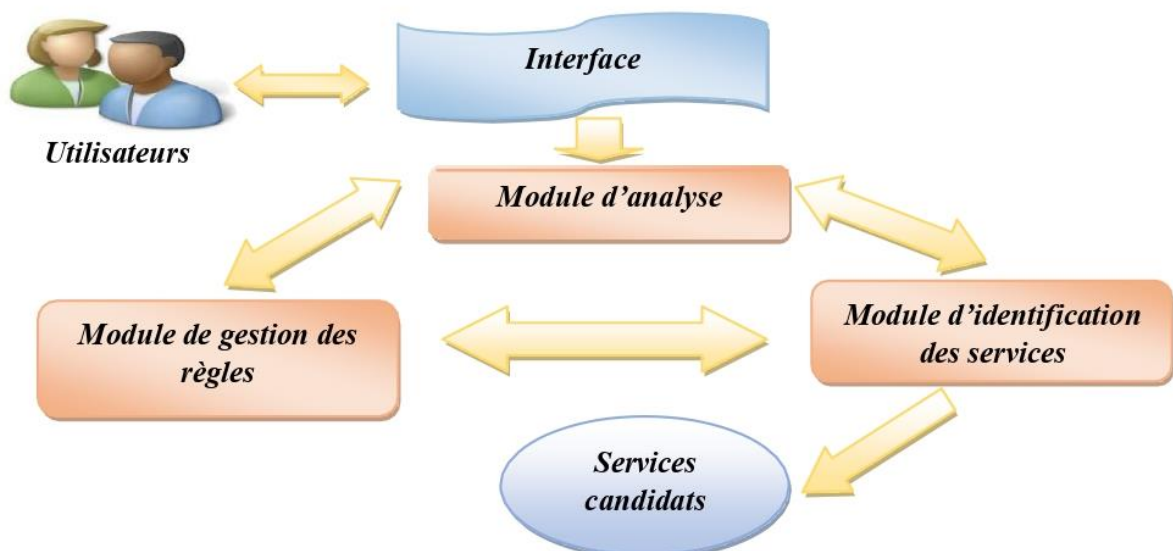


Figure 5.1 : l'architecture générale de notre Framework

4. Présentation des interfaces :

4.1. Page d'accueil :



Figure 5.2 : Page d'accueil de notre plateforme

4.2. La barre d'outils :

Charger un fichier XPDL :



Afficher le diagramme de BPMN 2.0 graphiquement :



Appliquer toutes les règles et identifier les services :



Appliquer une règle et identifier les services :



Ajouter une règle :






Gestionnaire des règles :



Exporter la liste des services candidats :



- Le guide d'utilisation : 
- Réinitialiser les fenêtres précédentes : 
- Quitter l'application : 

4.2.1. Charger un fichier XPDL :



C'est la première étape dans notre application, cette étape vient après l'étape de modélisation avec BPMN 2.0 d'une plateforme e-learning classique qui existe déjà à l'aide de l'outil Bizagi, elle permet d'exporter les diagrammes graphiques vers une représentation textuelle de type XPDL, ce qui permet une lecture et analyse facile par notre plateforme.

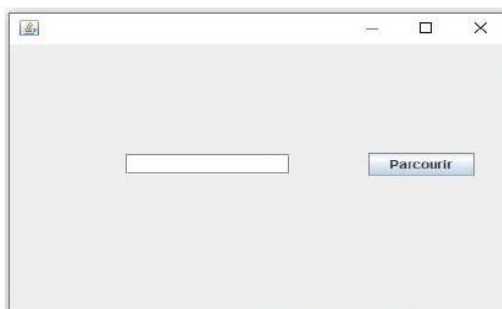


Figure 5.3 : charger un fichier XPDL

4.2.1.1. La phase prétraitement :

Dans cette phase le système vérifie si le type des toutes les tâches sont défini, si ce n'est pas le cas alors une boîte de dialogue affiche un message d'information qui signale les tâches qui ne sont pas définies.



Figure 5.4 : message d'affichage dans La phase prétraitement.

4.2.2. Afficher le diagramme de BPMN 2.0 graphiquement :



C'est l'étape qui suit immédiatement la phase de prétraitement, on peut afficher graphiquement les diagrammes BPMN importés à partir un fichier XPDL :

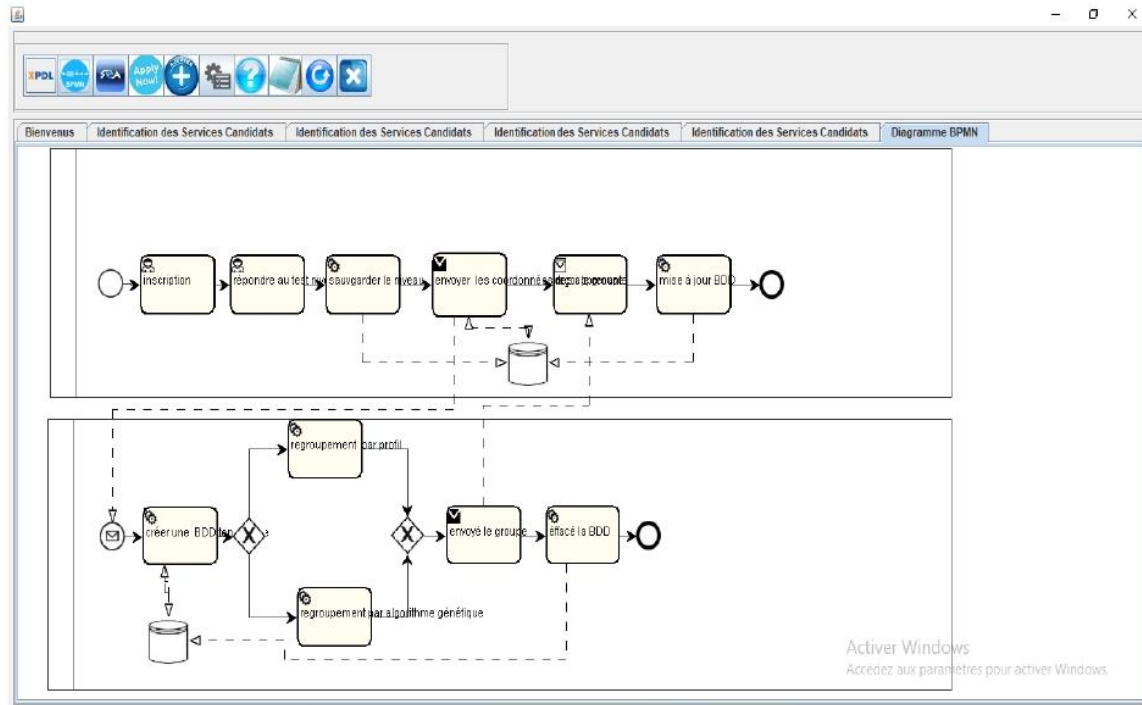


Figure 5.5 : Affichage des diagrammes BPMN 2.0 sous forme graphique.

4.2.3. Appliquer toutes les règles et identifier les services :



Dans cette partie, on peut appliquer toutes les règles prédéfinies suivantes :

- ❖ **Règle 01** : Regrouper les tâches de type service (Une Tâche de type service c'est une tâche automatisée c.à.d. sans intervention humain).
- ❖ **Règle 02** : Un processus (pool) est un service constitué de l'ensemble de ses tâches si toutes ces tâches ne sont pas de type manuel.
- ❖ **Règle 03** : Un service est une tâche où un ensemble de tâches successives entre la réception et l'envoi de message vers le même processus demandeur de traitement.

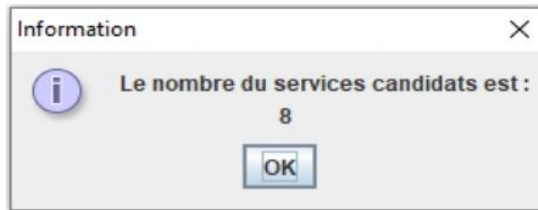


Figure 5.6 : Boite de dialogue affiche le nombre des services candidats identifié

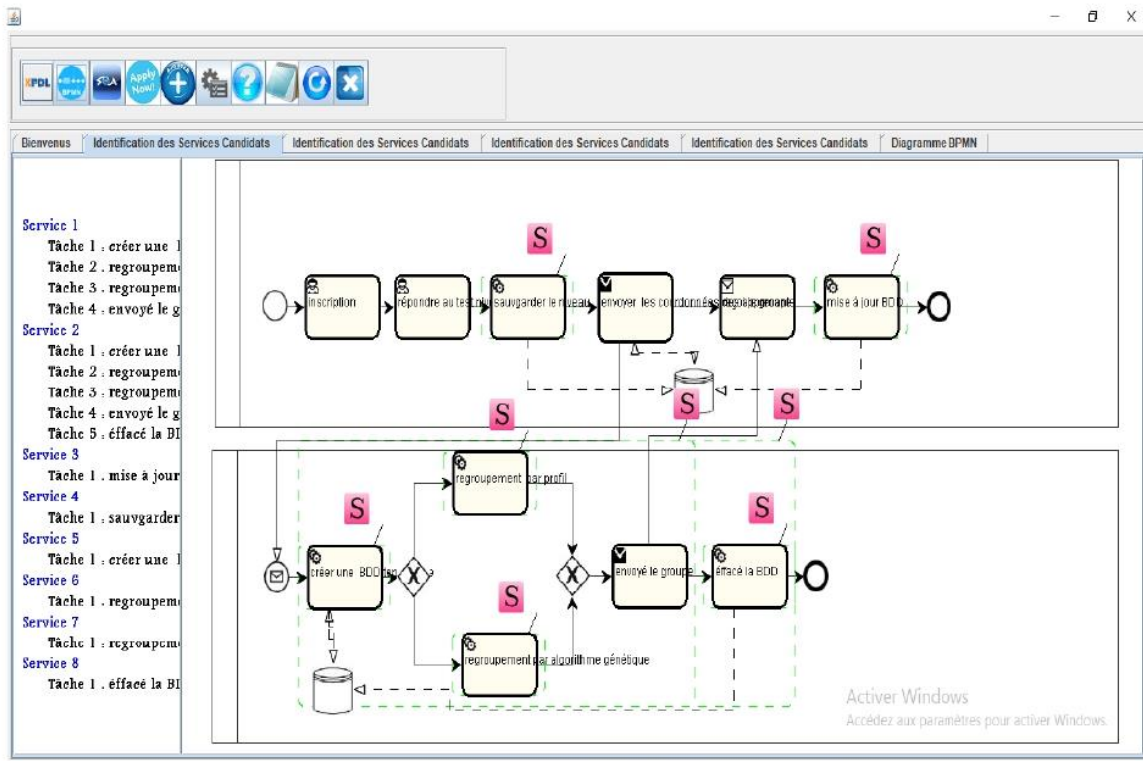


Figure 5.7 : affichage de l'option appliquer toutes les règles et identifier les services.

Si vous avez cliqué sur le bouton  « appliquer toutes les règles et identifier les services » sans importer un fichier XPDL, un message vous avertit qu'il n'y a pas de fichier XPDL.

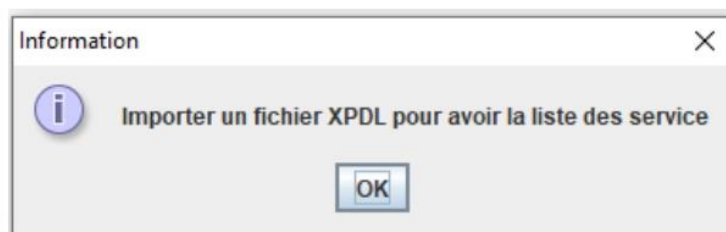


Figure 5.8 : message d'alerte pour l'inexistence d'un fichier XPDL.

4.2.4. Appliquer une règle et identifier les services :



Dans cette partie, l'utilisateur doit sélectionner la règle à appliquer parmi les règles prédéfinis dans notre plateforme :

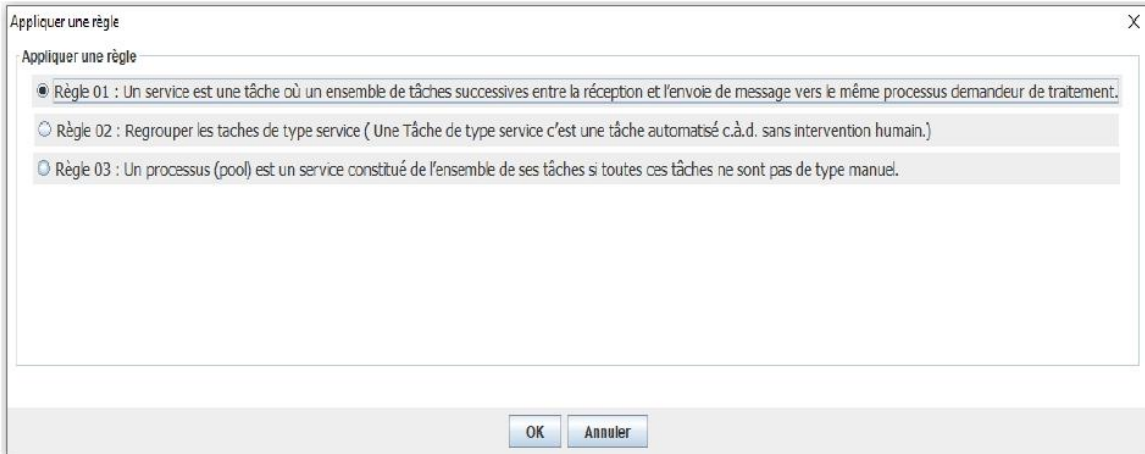


Figure 5.9 : fenêtre de sélection de règle particulière.

Enfin, on va afficher le nombre des services candidats identifié avec la règle 3.

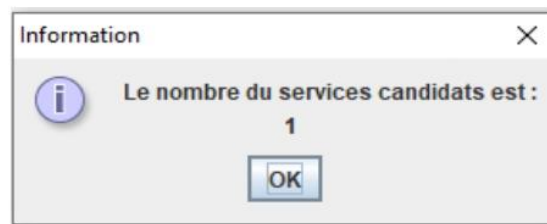


Figure 5.10 : Affichage du nombre des services identifiés avec l'application de la règle sélectionnée

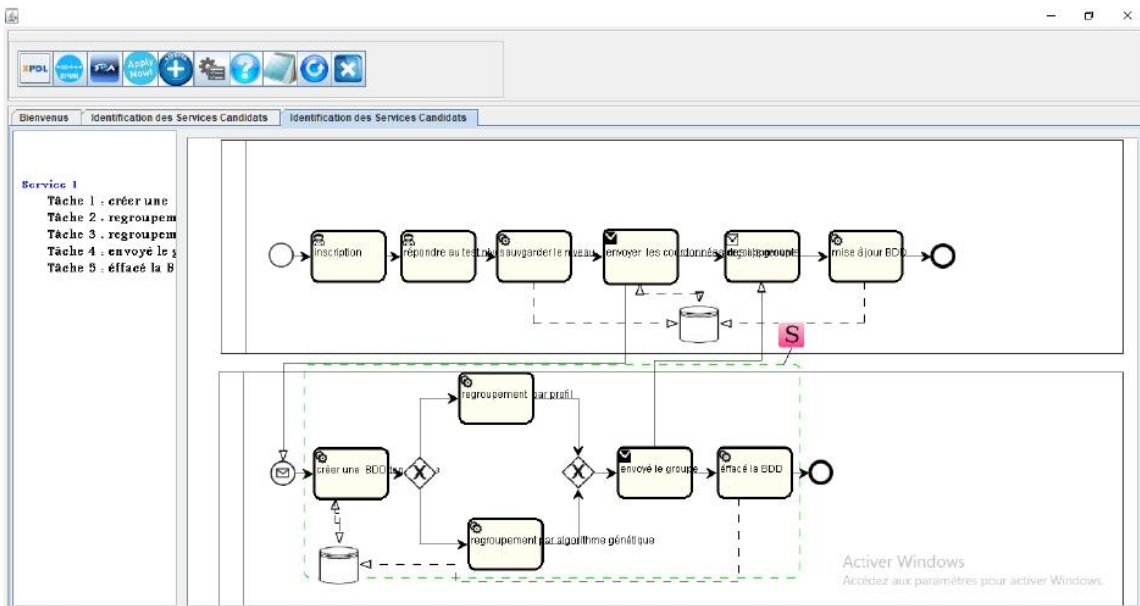


Figure 5.11 : Affichage des services identifiés avec la règle 3

4.2.5. Ajouter une règle :



Cette partie qu'on appelle aussi générateur des règles, permet de donner la main à l'utilisateur d'ajouter une règle qui n'existe pas dans notre catalogue des règles prédéfinies, et cela à travers une interface qui comporte les éléments de modélisation BPMN

Figure 5.12 : Interface de l'ajout d'une règle.

- **Bouton « Ajouter »** : permet de valider un élément.
- **Bouton « Terminer »** : permet de regrouper les éléments validés et de les enregistrer sous forme de règle dans un fichier, puis de quitter la page.
- **Bouton « Annuler »** : permet de quitter la page sans enregistrer la règle.

Voici un exemple d'une règle ajoutée :

Figure 5.13 : exemple d'une règle ajoutée.

4.2.6. Gestionnaire des règles :

Dans cette partie, lorsque l'utilisateur ajoute une règle cette dernière s'affiche dans la partie des règles ajoutées comme le montre la figure 5.14 en bas :



Figure 5.14 : Affichage du gestionnaire des règles.

Ci-dessous, nous expliquerons le rôle de chaque bouton comme suit :

- **Bouton « Appliquer »** : Ce bouton permet d'appliquer la règle que nous avons choisie et d'afficher les services candidats.
- **Bouton « Appliquer toutes les règles ajoutées »** : permet d'appliquer toutes les règles ajoutées.
- **Bouton « Appliquer toutes les règles »** : permet d'appliquer toutes les règles additives et les règles prédéfinies.
- **Bouton « Ajouter une règle »** : permet d'ajouter une nouvelle règle.
- **Bouton « Supprimer »** : pour supprimer la règle sélectionnée.
- **Bouton « Supprimer toutes les règles »** : pour vider le fichier des règles.
- **Bouton « Annuler »** : pour quitter la page.

4.2.7. Le guide d'utilisation :

Il vise à expliquer le contenu de notre projet et contient aussi une explication de A à Z pour identifier des services candidats à partir d'un diagramme d'une plateforme e-learning modélisée avec BPMN 2.0.



Figure 5.15 : Guide d'utilisation

4.2.8. Exporter la Liste des services candidats :

Afin de sauvegarder les résultats de l'identification des services, notre système donne la main à l'utilisateur d'exporter ces résultats sous forme d'un fichier texte.

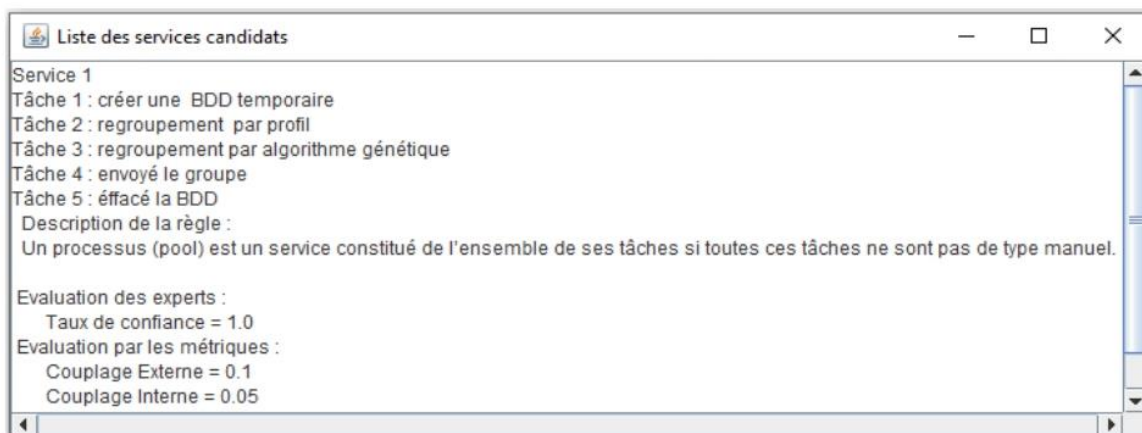


Figure 5.16 : Affichage des résultats d'identification dans un fichier texte.

5. Conclusion :

Nous avons présenté dans ce chapitre les différents modules de notre Framework d'identification des services d'une plateforme d'apprentissage classique ainsi que les étapes de suivre pour identifier un service à partir d'un diagramme BPMN.

Nous avons parlé aussi des outils de développement nécessaires pour la réalisation de notre projet. Nous avons aussi présenté l'interface de notre Framework service ainsi que les fonctionnalités offertes par ce dernier.

Conclusion générale et perspectives

Dans ces dernières années, l'architecture orientée services (SOA) est devenu de plus en plus apprécié dans le domaine de réingénierie logicielle, et ceci vue les avantages qu'elle apporte dans ce domaine tel que l'interopérabilité et la réutilisabilité et le plus important c'est la réduction des coûts.

L'architecture orientée services fonctionne dans le sens de la création de logiciels interopérables, elle fournit une séparation entre l'interface d'un service et sa mise en œuvre sous-jacente, de sorte que les consommateurs (applications) peuvent interagir avec le plus grand nombre de fournisseurs de services et que les fournisseurs peuvent facilement être échangés à la volée sans modification du code de l'application.

SOA présente le processus métier comme un service, BPM et SOA offrent une combinaison parfaite pour l'informatique d'entreprise. BPM fournit l'abstraction de plus haut niveau pour définir les processus métier, ainsi que d'autres capacités importantes de surveillance et de gestion de ces processus. Les services fournissent les fonctions qui prennent en charge ces processus. SOA fournit les capacités pour les services à combiner et pour soutenir et créer une entreprise agile et flexible. Le BPM sans SOA est utile pour créer des applications, mais difficile à étendre à l'entreprise. [44]

Nous avons suivi dans notre PFE l'approche top-down pour résoudre notre problématique qui consiste à la réingénierie d'une plateforme d'apprentissage classique vers une autre plus évoluée qui se base sur l'architecture orienté service (SOA). Où nous avons suiviez cinq (05) phases pour arriver en fin à identifier les services candidats à partir des diagrammes BPMN 2.0. Notre Framework applique des règles que nous avons déduits à partir des caractéristiques et des propriétés des services à savoir l'autonomie, la réutilisabilité, le couplage faible avec l'extérieur et la cohésion forte à l'intérieur de service afin d'identifier des services candidats à la programmation et à la réutilisation.

Comme perspective à ce travail, je propose de lancer une nouvelle étude en suivant une approche Bottom-up pour l'identification des service puis de faire une étude comparatives entre nos résultats et ceux de la deuxième approche afin de conclure laquelle des approche est la plus optimale.

Liste des bibliographies

- [1] Su, Y.Z.: e-Learning: The Window of Knowledge in the Future World. *Information Pioneers Magazine* 53, 1–2 (2003)]
- [2] Chen, H.Z.: A Study and Practice of Involving Evaluation in a SCROM Learning Environment. Master's Thesis, Department of Information Management, Providence University, Taiwan (2007)
- [3] Wang, K.F.: Planning and Constructing a High School Collaborative Operation System Based on SOA. Master's Thesis, Department of Information Management, National Kaoshong First University of Science and Technology, Taiwan (2006)
- [4] Chung C. Chang and Kou-Chan Hsiao Department of Information Management, Chinese Culture University, Yang Ming Shan, Taipei 111, Taiwan zcz@faculty.pccu.edu.tw « Using SOA Concept to Construct an e-Learning System for College Information Management Courses »
- [5] M.Faïsa et E.Imane, mémoire de master2 « Etude comparative des différentes plates-formes de la formation en ligne « E-Learning » », Juin 2015.]
- [6] BEKRAR M, mémoire de l'ingénieur , « Protection de la vie privée à base d'agents dans un système d'E-Learning,mémoire », 2013/2014.
- [7] COUTURIER, Jean-François. Élaboration et expérimentation d'une méthodologie agile permettant la migration vers une architecture orientée services en PME à l'aide d'OpenUP. 2011. Thèse de doctorat. École de technologie supérieure.
- [8] Oracle, Oracle IT Modernization Series: The Types of Modernization, an Oracle White Paper; September 2008.
- [9] G. Lewis, E. Morris, D. Smith, L. Wrage, L. O'Brien, Service-Oriented Migration and Reuse Technique (SMART), Proceedings of 13th IEEE International Workshop on Software Technology and Engineering Practice (WSTEP'05), September 2005.
- [10] ARSANJANI, Ali. Service-oriented modeling and architecture: How to identify, specify, and realize services for your SOA, 2004. Retrieved August, 2011, vol. 12.

- [11] Christophe Heubès , « mise en œuvre d'une SOA : les clés de succès », 2007
- [12] S. Izza, L. Vincent, P. Burlat, H. Solignac et P. Lebrun, Intégration d'applications : Etat de l'art et perspectives, JIE 2004, les 2^{èmes} Journées d'Informatique pour l'Entreprise, université Saad Dahleb, Blida (Algérie) 2004.
- [13] P. Sarang, R. Loganathan, F. Jennings, M. Juric, SOA Approach to Integration: XML, Web services, ESB, and BPEL in real-world SOA projects. Packt Publishing 2007. ISBN: 13 978-1-904811-17-6
- [14] Tanguy Crusson, Business Process Management, de modélisation à l'exécution : Positionnement par apport aux Architectures Orientées Services, Rapport INTALIO, 2003.
- [15] Les Architectures Orientées services : le moteur de l'innovation, livre blanc de Si (Syntec informatique), juin 2007.
- [16] JorisVanGeet, Serge Demeyer: Lightweight Visualisations of COBOL Code for Supporting Migration to SOA –ERCIM 2007. Electronic Communication of the European Association of Software Science and Technology Volume 8 (2007)
- [17] Millard, David E., et al. "(Semantic web) services for e-learning." International Journal of Knowledge and Learning 4.2-3 (2008): 298-315.
- [18] M. Papazoglou, Van-Den-Heuvel, "Service Oriented Architectures: Approaches, Technologies and Research Issues," The International Journal on Very Large Data Bases, vol.16. 2007, pp.389-415
- [19] Grini Ahlem, bennour khadija , mémoire master « Un Framework pour l'automatisation de l'identification de service à base des représentations BPMN » encadré par Mr seridi ali , université 08 mai 45 guelma, juin 2017
- [20] Ali Seridi, Lynda Dib, Riad Bourbia article « Modernization of e-learning platforms towards a service-oriented architecture » J.Electical Systems vol 15-1 (2019) : 123-132
- [21] A. M. Riad, Hamdy K. El-Minir, Haitham A. El-Ghareeb article « Review of e-Learning Systems Convergence from Traditional Systems to Services based Adaptive and Intelligent Systems » January 2009
- [22] Développement professionnel continu E-Learning, Fiche technique méthode, 31 janvier 2013.

- [23] Stefan Hüseemann, Mise en place d'une architecture de type SOA pour un projet informatique, Travail de bachelor, Université de Fribourg, Suisse, Janvier 2011.
- [24] Zakia Imane Kazi-Aoul, « Une architecture orientée services pour la fourniture de documents multimédia composés adaptables », l'École Nationale Supérieure des Télécommunications de Paris, 2008.
- [25] S. Jarzabek, « *Software reengineering for reusability* », *Computer Software and Applications Conference, 1993. COMPSAC 93. Proceedings., Seventeenth Annual International*, 1-5 nov 1993, p. 100-106 (ISBN 0-8186-4440-0, DOI 10.1109/COMPSAC.1993.404221)
- [26] Moon Ting Su, Chee Shyang Wong, Chuak Fen Soo, Choon Tsun Ooi et Shun Ling Sow ("Service-Oriented E-Learning System" IEEE, 2007).
- [27] Chung C. Chang et Kou-Chan Hsiao (Article « A SOA-Based e-Learning System for Teaching Fundamental Information Management Courses » *Journal of Convergence Information Technology*, Volume 6, Number 4. April 2011)
- [28] Peter Westerkamp ET Leonardo-Campus (« E-Learning as a Web Service » janvier 2003)
- [29] K. Palanivel ET S. Kuppuswami (« Service-Oriented Reference Architecture for Personalized E-learning Systems (SORAPES) » janvier 2011)
- [30] Lian Kei SOO, Eng Thiam YEOH & Sin Ban HO (Searching & Ranking Learning Objects in a Service Oriented Architecture for E-learning) *Proceedings of the 5th International Conference on IT & Multimedia at UNITEN (ICIMU 2011) Malaysia*
- [31] S. Sagayaraj, N.S. Rajalakshmi et M. Poovizhi (A SOA BASED E-LEARNING SYSTEM FOR TEACHING FUNDAMENTAL INFORMATION'S OF COMPUTER SCIENCE COURSES) Vol 01, Issue 02, December 2012 *International Journal of Web Technology*, <http://iirpublications.com> ISSN: 2278-2389
- [32] Marinela Mircea (SOA adoption in higher education: a practical guide to service oriented virtual learning environment) *Procedia - Social and Behavioral Sciences* 31 (2012) 218 – 223

- [33] SU, Moon Ting, WONG, Chee Shyang, SOO, Chuak Fen, et al. Service-oriented e learning system. In : Information Technologies and Applications in Education, 2007. ISITAE'07. 2007. p. 6-11.
- [34] Chantal Morley, Marie Bia-Figueiredo, Yves Gillette ,Livre « PROCESSUS MÉTIERS ET SYSTÈMES D'INFORMATION Gouvernance, management, modélisation » , 3e édition Dunob, Paris, 2011
- [35] Livre « Modélisation des processus métiers PowerAMC™ 16.0 » ID DU DOCUMENT : DC31020-01-1600-01, DERNIERE REVISION : Juillet 2011
- [36] Younes Lemrabet, « Proposition d'une méthode de spécification d'une architecture orientée services dirigée par le métier dans le cadre d'une collaboration inter- organisationnelle », thèse de doctorat, 7 juin 2012.
- [37] ADLA BENTELLIS Epouse FEDJKHI ,thèse doctorat «Intégration des Applications d'Entreprises /Une approche basée objectif pour la gestion des processus métier flexibles » ,Université Mentouri Constantine, le 07/02/2010.
- [38] SINI Ghanima, thèse doctorat « Méthode et outils pour la gestion des workflow – Modélisation ontologique des processus pour l'analyse », 07/03/2013.
- [39] Marie-Hélène Gentil « Management des processus, Bonnes pratique et retours d'expérience », université de Bordeaux.
- [40] Seridi Meryem, Benredjem Ahlam, « Implémentation de l'approche processus dans une organisation étude de cas 'les moulins amor ben amor' », juin 2016, dirigé par Me Djakhdjakha Lynda.
- [41] Mémoire du Master MALTT « BPMN 2.0 POUR LA MODÉLISATION ET L'IMPLÉMENTATION DE DISPOSITIFS PÉDAGOGIQUES ORIENTÉS PROCESSUS » par Julien Da Costa, GENÈVE, FÉVRIER 2014
- [42] Aïcha BAKKI ,THESE COTUTELLE DE DOCTORAT « Modèle et outil pour soutenir la scénarisation pédagogique de MOOC connectivistes » ,Thèse présentée et soutenue à

Agadir (Maroc), le « 12/12/2018 » Unité de recherche : Laboratoire Informatique de l'Université du Mans (LIUM) Thèse N° : 2018LEMA1033

[43] ABDELLATIF, Iheb. Vers une démarche d'aide à la décision pour l'identification des services d'une architecture orientée services. 2011. Thèse de doctorat. École de technologie supérieure.

[44] Dr. Gopala Krishna Behara, bptrends, Article « BPM and SOA : A Strategic Alliance », mai 2006.

[45] Plouin, Guillaume, Bruno Penneç, Pascal Grojean et Cyril Rognon. 2007. « SOA Perception des entreprises françaises ». Paris

[46] Wing, Hong Lam, et Shankararaman Venky. 2007. Enterprise Architecture and Integration: Methods, Implementation and Technologies IGI Global, 344 p.

[47] Klein, Scott 2007. Professional WCF Programming: .NET Development with the Windows Communication Foundation. Wrox press, 450 p.

[48] PORTER, Michael E. et DE LAVERGNE, Philippe. L'avantage concurrentiel. 1986.

[49] Gartner Inc. (2005) Business Process Management Suites Enhance the Control and Management of Business Processes.

[50] DEBAUCHE, Bernard et MÉGARD, Patrick. Business Process Management, éd. 2004.

[51] Manon Froger. Une approche d'accompagnement de la maturation BPM d'une entreprise et de la formalisation de ses processus métiers. Autre [cs.OH]. Ecole des Mines d'Albi-Carmaux, 2020.

[52] Chinosi, M., & Trombetta, A. (2012). BPMN: An introduction to the standard. Computer Standards & Interfaces, 34(1), 124–134. doi:10.1016/j.csi.2011.06.002

Liste des webographies

- [w1] <http://ludovictresson.com/skill/reingenierie-logicielle-refactoring.html> (Dernière navigation le : 30/09/2020)
- [w2] <https://www.w3.org/TR/2001/NOTE-wsdl-20010315> (Dernière navigation le : 30/09/2020)
- [w3] Approche processus disponible sur : <http://www.pqb.fr/plateform.php?i&if=33&ch=551#>. (Dernière navigation le : 24/09/2020)
- [w4] [Http : //blog.xebia.fr/2009/03/04/soa-du-composant-au-service-le-contrat-standardise/](http://blog.xebia.fr/2009/03/04/soa-du-composant-au-service-le-contrat-standardise/), (Dernière navigation le : 30/09/2020).
- [w5] [Http : //www.dotnet-france.com/Documents/WCF/Introduction%20a%20WCF.pdf](http://www.dotnet-france.com/Documents/WCF/Introduction%20a%20WCF.pdf), 17/05/2015 (Dernière navigation le : 30/09/2020).
- [w6] <http://www.redsen-consulting.com/2011/07/concepts-fondamentaux-soa>. (Dernière navigation le : 30/09/2020)
- [w7] http://tvquality.verollet.fr/files/Reingenierie_Rapport_Tomier_Verollet.pdf 11/06/2019 (Dernière navigation le : 30/09/2020)
- [w8] http://www.journaldunet.com/solutions/0403/040323_reengineering.shtml 09/05/2019 (Dernière navigation le : 30/09/2020).
- [w9] <http://www.ricoh.fr/common-business-terms/business-process-re-engineering/> 14/05/2019 (Dernière navigation le : 30/09/2020).
- [w10] <https://tecfa.unige.ch/proj/bpmn2015/> (Dernière navigation le : 30/09/2020).
- [w11] <http://www.differencebetween.net/technology/difference-between-uml-and-bpmn/> (Dernière navigation le : 30/09/2020).
- [w12] <https://bizagi-process-modeler.fr.softonic.com/> (Dernière navigation le : 01/10/2020).
- [w13] <https://www.memoireonline.com/12/08/1644/SOA--Definition-Utilisation-dans-le-monde-de-la-banque-et-methodologie-de-test.html> (Dernière navigation le : 01/10/2020).
- [w14] The Open Group : <https://www.opengroup.org/soa/source-book/soa/p1.htm>, consulté 24 septembre, 2020.