

REPUBLICQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR
ET DE LA RECHERCHE SCIENTIFIQUE



UNIVERSITE SAAD DAHLEB DE BLIDA
FACULTE DES SCIENCES
DEPARTEMENT INFORMATIQUE



MEMOIRE DE FIN D'ETUDES
Pour l'obtention du Diplôme MASTER Informatique
Thème

**La Réalisation d'une architecture orientée service pour
l'aide à la décision avec l'approche SOA⁺^d.**
Application : gestion de stock

Organisme d'accueil :

Pharmacie : FRONCE PH ANOU

Promotrice :

Mme BOUMAHDHI Fatima

Réalisé par :

Mlle Boucherdid Fatma
Mme Belkacem Nabila

Promotion : 2014- 2015

MA-004-223-1

RESUME

RESUME

De nos jours, l'environnement concurrentiel des décideurs d'entreprises est de plus en plus rude et évolue rapidement. Cet accroissement est causé par plusieurs facteurs : la technologie de l'information et de communication, une complexité structurelle des décisions et une plus forte compétition.

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

L'absence des solutions architecturales pour résoudre ce problème a plongé les SI dans une situation de blocage vis-à-vis des exigences métiers.

Les architectures orientées services (SOA) offrent aujourd'hui un nouveau modèle pour résoudre ces problématiques. Elles permettent de mettre le SI au service des métiers, en procédant à son alignement avec les processus d'entreprise.

L'aide à la décision peut profiter tout à fait de l'architecture SOA or cette dernière n'inclut pas l'aspect décisionnel.

Dans le cadre de notre PFE, nous nous proposons d'inclure les aspects décisionnels dans l'architecture SOA et l'enrichir avec des informations spécifiquement d'aide à la décision. Nous avons retenu la gestion des stocks pour matérialiser la migration des aspects d'aide à la décision

dans une architecture SOA.

Mots clés : SOA, MDA, SoaML, Service, Orchestration, Aide à la décision, Service Web, Gestion des stocks, Méthode ABC, Modèle PARETO, Modèle Wilson.

ملخص

في الوقت الحاضر، تعد البيئة التنافسية لرجال اعمال المؤسسات في تزايد مستمر و هذا يعود الى عدة عوامل: تكنولوجيا المعلومات و الاتصالات، صعوبة و تعقيد هيكل القرارات و المنافسة الحادة.

نظرا لأهمية القرارات الاستراتيجية، تم اقتراح ادوات معلوماتية و نماذج مختلفة بهدف تسهيل المهمة على المسؤولين، لكن مع مرور الوقت واجهت الشركات مشكلة تكديس المكونات النموذجية الغير المتجانسة في نظم معلوماتها.

غياب الحلول الهيكلية لحل هذه المشكلة ادى بنظم المعلومات الى عدم قدرتها لتلبية متطلبات العمل.

البنية الخدمية تقدم نموذج جديد لحل مشكلة تكديس المكونات النموذجية الغير المتجانسة، حيث تجعل نظم المعلومات في خدمة الهيكليات التجارية.

يمكن لعملية صنع القرار الاستفادة بشكل كامل من البنية الخدمية لكن هذه الاخيرة لا تتضمن هذا الجانب.

العمل المقدم في هذه المذكرة يقترح ادماج الجانب المذكور اعلاه في البنية الخدمية وإثرائها بمعلومات تساعد على اتخاذ القرار. لدعم عملنا نقترح تطبيق هذه البنية على نظام ادارة المخزونات لنبين امكانية تطبيق جانب اتخاذ القرارات من خلال البنية الخدمية.

الكلمات الرئيسية: البنية الخدمية، البنية الموجهة بالنماذج، لغة تصميم البنية الخدمية، خدمة الواب، صنع القرارات، ادارة المخزونات، الطريقة(ABC)، (نموذج) PARETOU، (نموذج) WELS ON.

ABSTRACT

ABSTRACT

Nowadays, the competitive environment of business decision makers is becoming increasingly

harsh and rapidly changing. This increase is caused by several factors: the information technology and computers, a structural complexity of decisions and greater competition.

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

The absence of architectural solutions for this problem has plunged into an impasse opposite the business requirements.

Service-oriented architectures (SOA) now offer a new model for solving these problems. They help put IT (Information Technology) to serve the business, conducting its alignment with business processes.

The decision support can benefit fully from the SOA or the latter does not include the decision-making aspect.

Via the work presented in this thesis, we propose to include decision aspect in SOA and enrich it with information specific decision support through the migration of decision's aspects of an application of management inventory to the architecture mentioned above.

REMERCIEMENTS

Que seraient nos efforts sans l'aide du tout puissant que nul ne peut s'estimer suffisamment reconnaissant envers lui ; nous le remercions pour La résistance et la volonté qu'il nous a offert toute au long de notre parcours

Nous remercions jamais assez la personne qui nous a guidé, conseillé, orienté pas à pas dans notre travail ; notre sincère gratitude envers Mme Bou mahdi Fatima qui nous a apporté écoute et aide avec une patience prodigieuse ; que le tout puissant lui soit d'appuis là où ses pas prendront part dans la vie.

Nos gratitudes vont à l'ensemble des enseignants de BLIDA qui nous ont transmis leurs savoirs et leurs connaissances tout au long de nos cinq dernières années.

Nos remerciements vont aux membres du jury qui ont accepté de sacrifier une part de leurs temps pour examiner notre travail.

Enfin, à tous nos proches, familles et amis pour le soutien et leur présence.

A tous...

Merci Beaucoup

DEDICACES

C'est avec une immense fierté que je dédie ce modeste travail de fin d'études à ceux qui ont fait de moi ce que je suis aujourd'hui, grâce à leurs sacrifices, leurs tendresses et leur amour : ma chère maman et mon chère marie mes très chère parents que dieu me les garde

A toute ma petite famille : mes frères Idriss et lounis et Monhamed ali et sœurs Noria et son époux et Ibtissem et Amira et mon époux, pour leur soutien moral et leurs encouragements, un grand merci à eux

Mes anges et mes filles : Meryem et Assia.

Mon époux Arezki, et tout ma belle-famille.

A mes tentes : Malika, Ouardouche, Djamilia, Salha, Zahra.

A ma très chère maman.

Mon binôme : Fatima et toute sa famille.

Ma promotrice : Boumahdi Fatima.

A toutes les personnes qui nous ont prodigué des encouragements et se sont donné la peine de nous soutenir durant cette formation.

Mme BELKACEM NABILA

DEDICACES

C'est avec une immense fierté que je dédie ce modeste travail de fin d'études à ceux qui ont fait de moi ce que je suis aujourd'hui, grâce à leurs sacrifices, leurs tendresses et leur amour : mes très chères parents que Dieu me les garde

A toute ma petite famille : mon frère Mohammed, ma sœur Fatima, pour leur soutien moral et leurs encouragements, un grand merci à eux

Mes grands-parents, mes tantes, et mes oncles.

A mes cousines : Safia, Sabrina.

A ma meilleure amie : Zina.

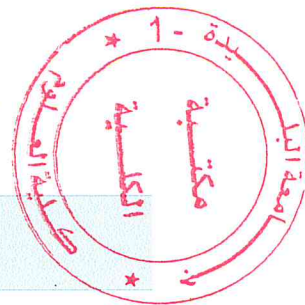
Mon très cher binôme : Nabila et toute sa famille.

Ma promotrice : Boumahdi Fatima.

A toutes les personnes qui nous ont prodigué des encouragements et se sont donné la peine de nous soutenir durant cette formation.

FATMA BOUCHARDID

Liste des abréviations



| | |
|-------------------------|---|
| DRL | Decision Rquirement Level |
| DMN | Décision Model Notation |
| DLL | Décision Logic Level |
| AJAX | Asynchronous Javascript And XML |
| API | Application Programming Interface |
| BPA | Business Process Automation |
| BPD | Business Process Diagram |
| BPEL | Business Process Execution Language |
| BPEL4Peo ple | Business Process Execution Language for People |
| BPM | Business Process Management |
| BPMI | Decision Logic Levele |
| BPMN | Business Process Modeling Notation |
| BPMS | Business Process Management System |
| B2BI | Business to Business Integration |
| DSI | Directeur des Systèmes d'Information |
| EAI | Enterprise Application Integration |
| EDI | Environnement de Développement Intégré |
| EJB | Enterprise Java Bean |

| | |
|-----------------|--|
| ERP | Enterprise Resource Planning |
| ESB | Enterprise Service Bus |
| HTML | Hyper Text Markup Language |
| HTTP | HyperText Transfer Protocol |
| IBM | International Business Machines |
| IT | Information Technology |
| JAX-RPC | Java API for XML based RPC |
| JBoss AS | JBoss Application Server |
| JBPM | Jboss Business Process Management |
| JPDL | JBPM Process Definition Language |
| JSF | JavaServer Faces |
| JSP | JavaServer Pages |
| J2EE | Java 2 Enterprise Edition |
| MDA | Model Driven Architecture |
| MOA | Maîtrise d'OuvrAge |
| DMN | Decision Requirements Diagram. |
| PGI | Progiciel de Gestion Intégré |
| POJO | Plain Old Java Object |
| RPC | Remote Procedure Call |

| | |
|--------------|--|
| RPC | Remote Procedure Call |
| RUP | Rational Unified Process |
| SDM | System Development Management |
| SAP | Systems, Applications, and Products |
| SGBDR | Système de Gestion de Bases de Données Relationnelles |
| SOA | Service Oriented Architecture |
| SOAP | Simple Object Access Protocol |
| TOGAF | The Open Group Architecture Framework |
| UDDI | Universal Description Discovery and Integration |
| UML | Unified Modeling Language |
| UP | Unified Process |
| WS | Web Service |
| WSDL | Web Services Description Language |
| XML | Extensible Markup Language |

Table des matières

Table des matières

| | |
|---|-----------|
| 1. Introduction..... | 11 |
| 2. Présentation du sujet | 12 |
| 2.1. L'entreprise concernée..... | 12 |
| 2.2. Problématique..... | 12 |
| 3. Organisation du mémoire..... | 14 |
| Partie 1 Concepts de base | 15 |
| Chapitre I Architecture Orientée Service SOA | 17 |
| 1.1 Introduction | 17 |
| 1.2 Les architectures orientés services..... | 18 |
| 1.2.1 Définition de l'architecture SOA..... | 18 |
| 1.2.2 Objectif de L'architectures SOA..... | 19 |
| 1.2.3 Vision de l'architecture SOA | 20 |
| 1.2.4 Avantages de l'architecture SOA | 21 |
| 1.2.5 La gouvernance SOA | 22 |
| 1.3 Les services et l'architecture SOA..... | 23 |
| 1.3.1 Les types de service | 24 |
| 1.3.2 Caractéristiques des services..... | 25 |
| 1.4 Les services web et l'architecture SOA | 28 |
| 1.4.1 Définition des services web | 27 |
| 1.4.2 Les standards de l'architecture orientée service web..... | 28 |
| 1.5 Les éléments de base du SOA..... | 29 |
| 1.5.1 Eléments organisationnels d'une SOA..... | 29 |
| 1.5.2 Eléments techniques d'une SOA..... | 30 |
| Conclusion | 31 |
| Chapitre II Méthodologie de développement SOA..... | 32 |
| 2.1 Introduction | 33 |
| 2.2 Stratégies de la conception des architectures SOA | 33 |

| | |
|--|-----------|
| 2.2.1 Top Down..... | 33 |
| 2.2.2 Bottom Up..... | 33 |
| 2.2.3 Outside In..... | 33 |
| 2.2.4 Middle Out..... | 34 |
| 2.3 Les méthodologies de développement des SOA | 35 |
| 2.3.1 SOMA (Service-Oriented Modeling and Architecture) d'IBM..... | 35 |
| 2.3.2 SOAF (Service Oriented Architecture Framework) | 37 |
| 2.3.3 La méthodologie CBDI-SAE | 38 |
| 2.3.4 La méthodologie SODM | 39 |
| 2.3.5 La méthodologie de Thomas Erl | 40 |
| 2.3.6 La méthodologie de S. Jones..... | 40 |
| 2.3.7 La méthodologie de Zimmerman SOAD | 41 |
| 2.4 Comparaison des méthodologies de SOA..... | 42 |
| Conclusion | 45 |
| Chapitre III La méthode SOA+d..... | 46 |
| 3.1 Description de la méthode SOA ⁺ d | 47 |
| 3.2 Concepts de base de l'approche SOA+d | 48 |
| 3.2.1 Language SoaML (Service Oriented Architecture Modeling Language) | 48 |
| 3.2.2 Modèle de décision et de notation..... | 49 |
| 3.3 Principes de SOA+d..... | 53 |
| 3.3.1 Principe d'ouverture | 53 |
| 3.3.2 Réutilisation de l'existant | 53 |
| 3.4 Description détaillée de la méthode proposée SOA+d..... | 54 |
| 3.4.1 Phase1 : La phase d'analyse..... | 54 |
| 3.4.2 Phase2 : Identifier et classer les services..... | 57 |
| 3.4.3 Phase3 : Modélisation de Services | 60 |
| 3.4.4 Phase4 : La réalisation | 62 |
| Conclusion | 64 |
| Partie 2 Étude de cas..... | 65 |

| | |
|---|------------|
| Chapitre IV : Conception et réalisation..... | 67 |
| 4.2 Etude de cas..... | 67 |
| 4.2.1 Phase1 : Analyse | 67 |
| 4.2.2 Phase 2 : Identifier et classer les services..... | 82 |
| 4.2.3 Phase 3 : Modélisation de services :..... | 83 |
| 4.2.4 Phase 4 : La réalisation | 93 |
| Conclusion : | 112 |

Conclusion générale

Annexe 1

Bibliographie

Liste des figures

| | |
|---|----|
| Figure 1:1 Exemple d'architecture SOA (IBM, 2006)..... | 19 |
| Figure 1:2 Les trois couches primaires d'une SOA (Erl & Upper Saddle River, 2005)..... | 20 |
| Figure 1:3 La gouvernance SOA (Vendeville, 2011)..... | 22 |
| Figure 1:4 Les types de services (Daehene, 2009)..... | 23 |
| Figure 1:5 Gestion de prêt en couplage fort (Audrey, 2013) | 24 |
| Figure 1:6 Gestion de prêt en couplage faible (Audrey, 2013). | 24 |
| Figure 1:7 Le contrat des services. | 25 |
| Figure 1:8 Service à forte granularité vers service à faible granularité (Adrien, 2008). | 26 |
| Figure 1:9 Modèle opérationnel de l'architecture orientée services. | 27 |
| Figure 1:10 Service métier, concept d'opération (Bourey, 2007)..... | 29 |
| Figure 1:11 les éléments techniques du SOA (Thai Tri, 2005) | 30 |
| Figure 1:12 : Intégration de l'ESB dans les principes d'une SOA. (Yann , 2007) | 31 |
| Figure 2:1 Stratégie d'intégration (Xebia IT Architects, 2007). | 34 |
| Figure 2:2 la méthodologie SOMA. (Arsanjani, Ghosh, Allam , & Abdollah, 2008)..... | 36 |
| Figure 2:3 La méthodologie SOAF (Erradi, Anand, & Kulkarni , 2006). | 38 |
| Figure 2:4 La méthode SOD-M | 39 |
| Figure 2:5 La méthodologie de SOAD (Zimmermann, Pal , & Clive , 2004). | 41 |
| Figure 2:5 La méthodologie de SOA ⁺ d .(Boumahdi ,et .,2014) | 42 |
| Figure 3:1 La description des phases de SOA ⁺ d avec la notation BPMN..... | 47 |
| Figure 3-2 Comment SOA + d intègre au processus de développement des services existant..... | 54 |
| Figure 3-3 Les différents choix technique proposés par SOA ⁺ d | 62 |
| Figure 4:1 Diagramme de cas d'utilisation pour la gestion des stocks pharmaceutique. | 69 |
| Figure 4:2 Les éléments graphiques de BPMN. | 70 |
| Figure 4:3 Le processus de gestion des stocks de la pharmacie. | 71 |
| Figure 4:4 Le sous processus d'analyser l'état des stocks. | 72 |
| Figure 4:5 Le sous processus Estimer les commandes..... | 74 |
| Figure 4:6 Le sous processus Lancer les commandes. | 75 |
| Figure 4:7 choix du fournisseur DMN. | 77 |
| Figure 4:8 Estimer la quantité a commandé DMN. | 79 |
| Figure 4:9 classification des médicaments DMN..... | 81 |
| Figure 4:10 The SOA+d application in PIM. | 83 |
| Figure 4:11 Architecture générale. | 88 |
| Figure 4:12 Le contrat de service « Exécuter Livraison ». | 89 |

| | |
|--|------------|
| Figure 4:13 Le contrat de service « Transmettre Procès Verbale Réception ». | 90 |
| Figure 4:14 Le contrat de service «Transmettre Demande D’approvisionnement ». | 91 |
| Figure 4:15 Le contrat de service « Signaler Problème Article». | 92 |
| Figure 4:16 L’architecture des composants techniques. | 93 |
| Figure 4:17 Les choix techniques de notre solution. | 96 |
| Figure 4:18 Architecture des composants technique de notre solution. | 96 |
| Figure 4:19 Service CRUD. | 97 |
| Figure 4:20 L’architecture des couches. | 98 |
| Figure 4:21 Exemple d'un fichier WSDL . | 100 |
| Figure 4:21 Les services CRUD et décisionnels. | 101 |
| Figure 4:22 Les étapes de transposition des processus métiers. | 102 |
| Figure 4:23 Définitions des rôles. | 102 |
| Figure 4:24 Partie du sous processus Analyse Stock. | 103 |
| Figure 4:25 Intégration des formulaires. | 104 |
| Figure 4:27 La fenêtre Accueil. | 105 |
| Figure 4:28 Espace pharmacien. | 106 |
| Figure 4:29 Espace magasinier. | 107 |
| Figure 4:31 services CRUD. | 108 |
| Figure 4:32Tableau d’estimation de quantité à commander. | 109 |
| Figure 4:34 Intalio Workflow «Authentification». | 110 |
| Figure 4:35 Intalio Workflow «Liste des tâches». | 110 |
| Figure 4:36 Formulaire de commande. | 111 |
| Figure 4:37 Intalio console. | 111 |

Liste des tableaux

| | |
|---|-----------|
| Tableau 2-1 La comparaison des méthodologies (Pascal Bou Nassar, 2012)..... | 44 |
| Tableau 3-1 Les composants de DRD (OMG, 2014). | 60 |
| Tableau 3-2 Résumé des différentes Services de SOA+ d et leurs caractéristiques..... | 62 |

Introduction générale

Introduction Générale

1. Introduction

Un système d'Information (SI) représente l'ensemble des éléments participant à la gestion, au traitement, au transport et à la diffusion de l'information au sein de l'organisation. C'est un composant indispensable pour la mise en œuvre de la stratégie de l'entreprise. Une des techniques de concevoir le SI qui privilégie les processus métiers et garantit l'interopérabilité, la flexibilité et la réutilisabilité, est l'architecture orientée service ou SOA (Service Oriented Architecture).

SOA est un ensemble de services métiers qu'une organisation veut offrir à ses partenaires et clients. Ces services étant décrits d'une façon standardisée permettant l'assemblage de Services. Ainsi l'identification des services d'une architecture orientée services constitue la première phase du cycle de vie de SOA.

L'élément clé de l'architecture SOA est le service web. En effet, les services web ne sont qu'un moyen de mettre en place une architecture orientée services et d'atteindre ses objectifs. Ils sont considérés comme étant l'évolution naturelle du web. Les services web facilitent non seulement les échanges entre les applications de l'entreprise mais surtout permettent une ouverture vers les autres entreprises.

Dans le cadre de ce PFE on a proposé un système d'information de Gestion de stock au en inclurons la notion du l'architecture SOA et l'aide à la décision au sein de cette architecture. En utilisant une nouvelle approche SOA +d qui devait bien évidemment respecter les principes de SOA et assurer que la logique métier et applicative seront en parfait alignement. La présente approche sera testée et validée sur un cas d'étude réel qui est la gestion de stock pharmaceutique hospitalière.

2. Présentation du sujet

2.1. L'entreprise concernée

L'entreprise concernée par la mise en œuvre est la pharmacie centrale de l'hôpital France Phonos de Blida. Cette hôpital possède l'une des plus grandes pharmacies de Blida cette dernière essaie de satisfaire les besoins des différents services qui sont au sein de cette hôpital ainsi que les hôpitaux de Blida. Elle appartient au service (CHU) **centre hospitalier universitaire** qui a été créé par un décret 86301 du 16.12.1986 portant de création de centre hospitalière universitaire de Blida.

2.2. Problématique

La problématique de communication entre les applications hétérogènes au sein des entreprises est un vieux défi de l'informatique. On a essayé de le résoudre en proposant diverses méthodes et technologies. Or l'empilement progressif des applications a conduit à une situation intenable de « silos ».

Les architectures orientées services (SOA) offrent aujourd'hui une formidable opportunité pour résoudre cette problématique, par conséquent elle est de plus en plus demandée et utilisée dans les entreprises, en contrepartie, ces dernières ont besoin des processus décisionnels dont la dimension est exclue dans ce type d'architecture.

Partant de ce constat, nous nous intéressons à l'élaboration d'une nouvelle approche d'intégration des aspects décisionnels dans l'architecture SOA, pour qu'elle soit utilisée d'une bonne manière.

Pour appuyer notre approche nous allons exploiter la nouvelle architecture à savoir SOA incluant les aspects décisionnels dans un système de gestion des stocks.

2.3. Objectifs de l'étude

L'objectif de ce mémoire est de pouvoir appliquer l'architecture SOA sur un système d'information de gestion de stock et mettre à profit les possibilités offertes par les services web en identifiant les processus métiers.

- Faire migrer un système d'aide à la décision vers une architecture technique de type SOA en utilisant l'approche soa+d.
- Identifier les mécanismes de passage entre une architecture fonctionnelle vers une architecture technique pour les systèmes d'aide à la décision
- .Mettre à profit les possibilités offertes par les web services.
- Appliquer l'architecture SOA sur un système décisionnel.

3. Organisation du mémoire

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

La première partie sera consacrée à l'étude théorique des grands concepts rencontrés lors de la réalisation de notre projet, structurée en trois chapitres. Le premier chapitre va traiter l'architecture orienté service, le second s'intéressera à la méthodologie de développement SOA et à la technologie des services web cœur de notre travail. Enfin, le troisième chapitre présentera la méthode soa +d qui sera appliqué sur notre processus métier tout au long de notre travail.

Nous aborderons dans la deuxième partie, la conception de notre solution avec la méthode SOA+d, adoptée pour la mise en œuvre d'une architecture orientée service, en utilisant les langages de modélisation SOAML et BPMN et DMN. Ensuite La conception de l'architecture orientée services par la dérivation des modèles élaborés, nous passerons à l'implémentation de la solution retenue au niveau de la conception après une description de l'environnement du travail. En fin une conclusion générale et des perspectives viendront clôturer ce mémoire.

Partie 1 Concepts de base

- **Chapitre I** : Architecture Orientée Service
- **Chapitre II** : les méthodologies de développement SOA
- **Chapitre III** : La méthode SOA^{+d}.

Chapitre 1 : Architecture Orientée Service SOA

1.1 Introduction

De nos jours, l'évolution rapide des entreprises et ses interactions fortes avec leurs environnements ont incité à de nouvelles exigences et une évolution de son système d'information.

L'intégration entre les différents systèmes d'information est devenue indispensable à l'aide de nouvelles technologies. Pour ce faire, l'introduction des nouvelles architectures informatiques est considérée comme une révolution pour aider le développement de ces systèmes étendus et complexes.

En effet, devant les exigences croissantes et continues dans le domaine informatique pour la conception des systèmes d'information, les architectures logicielles ne cessent d'évoluer. L'architecture orientée services a été introduite afin de rendre la conception des logiciels une tâche plus facile, et afin d'assurer la réutilisabilité de services et d'augmenter l'interopérabilité à grande échelle.

En effet, les systèmes d'information sont composés d'un ensemble de fonctions dont chacune est appelée service qui peut interagir avec son environnement à travers l'échange de messages

Cependant, avec l'évolution successive des exigences des entreprises, la valeur réelle d'un service deviendra insatisfaisante, d'où la nécessité d'une nouvelle approche pour la composition de services afin d'apporter une valeur ajoutée. Dans ce contexte, nous parlons alors de la composition de services dans l'architecture orientée service. Un service composé assemble divers services et coordonne les interactions au cours de leurs exécutions en vue d'utiliser leurs fonctions et réaliser des tâches plus complexes.

1.2 Les architectures orientés services

1.2.1 Définition de l'architecture SOA

Le concept de SOA est né du besoin de pouvoir gérer l'architecture des SI avec plus de souplesse et ainsi permettre un meilleur alignement stratégique des SI. Plusieurs définitions sont retenues :

« SOA est une forme d'architecture technologique qui adhère au principe d'orientation service. Lorsqu'elle est réalisée par le biais d'une plateforme de web services, SOA parvient à soutenir et promouvoir l'orientation service au sein des business process et domaines d'automation d'une entreprise. ». (Erl & Upper Saddle River, 2005) .

La SOA est un style d'architecture qui permet la réorganisation du Système d'Information. Elle permet l'encapsulation des fonctionnalités d'un système d'information en un ensemble de services faiblement couplés appartenant à la fois au niveau métier et au niveau technique de l'entreprise. Les services, munis d'un contrat d'utilisation et d'une interface de description, seront publiés dans des registres de services afin qu'ils puissent être invoqués par d'autres services. (Sodki , 2008).

Cette architecture à la mode répond aux problèmes de réutilisation d'outils (ou produits) des entreprises. Pour mieux comprendre sa définition, 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.

FIGURE 1.

Exemple d'architecture SOA**Applications opérationnelles**

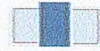
Système de prêt hypothécaire à taux fixe (progiciel)



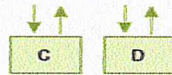
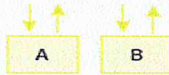
Système de prêt hypothécaire à taux variable (application J2EE sur mesure)



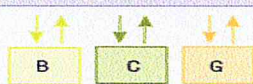
Système de crédit non garanti (.NET)



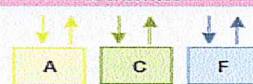
Éditeur de relevés de compte intégré (système principal)

**Décomposition en services invisibles** (tâches récurrentes – par exemple, créer un compte, vérifier la solvabilité)**Nouvelles fonctionnalités**

Nouveau système de services de crédit hybride conjuguant les services d'applications classiques



Nouveau portail WebCredit regroupant les services de sources internes et externes



Source : IBM Institute for Business Value.

Figure 1 1 Exemple d'architecture SOA (IBM, 2006).**1.2.2 Objectif de l'architecture SOA**

L'objectif principal des architectures orientées services est d'assurer un alignement entre les activités métier et le système d'information d'une manière à optimiser l'efficacité et l'agilité de l'entreprise. En effet, dans une SOA, l'entreprise se base sur le modèle du domaine métier qui décrit le fonctionnement des activités pour concevoir des services réutilisables en tirant parti des propriétés des services. Ceci permettra d'adapter rapidement le système d'information suite aux changements métier.

Dans les architectures orientées services, les nouvelles activités métier pourront être perçues comme étant des processus métier réalisés par des services. En effet, l'objectif principal dans une SOA est de découper les activités métier en des séries d'activités réutilisables qui pourront être complètement ou partiellement automatisées par des services métier. Les services reflètent à la fois l'image métier des activités et l'image technique des fonctions implémentées au niveau du système d'information.

1.2.3 Vision de l'architecture SOA

Les responsables métier percevront les services comme des fonctions métier réutilisables par différents clients ou partenaires. Les responsables techniques percevront les services comme étant des fonctionnalités applicatives, disponibles pour être réutilisées pour construire facilement et rapidement de nouvelles applications suite à des changements ou de nouveaux besoins métier.

La Figure I-2 illustre les concepts explicités ci-dessus. Dans cette figure, nous trouvons les processus métier qui représentent le cœur des activités métier de l'entreprise. La couche service représente l'interface entre les processus métier et les composants de l'architecture applicative. En particulier, nous avons représenté dans la couche infrastructure trois applications pour mettre en évidence la possibilité d'exposer les fonctionnalités des applications existantes dans l'entreprise en tant que services. Ceci permet d'optimiser la réutilisabilité des applications existantes dans la construction de nouvelles applications.

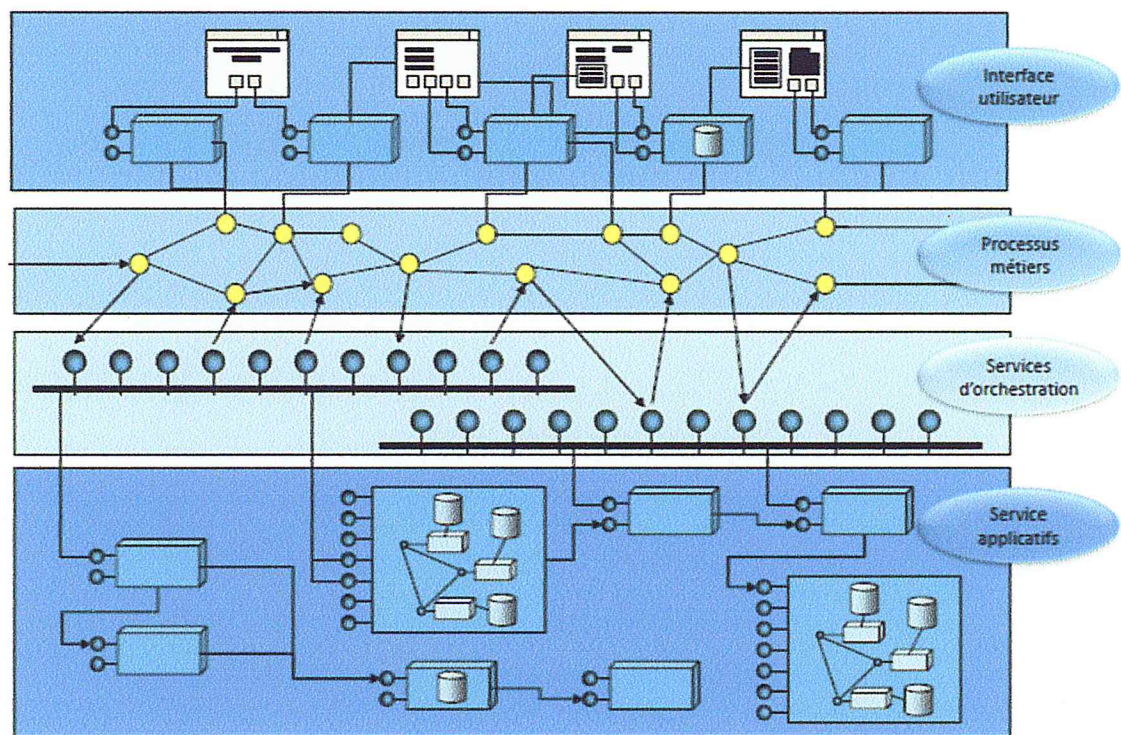


Figure 1 :2 Les trois couches primaires d'une SOA (Erl & Upper Saddle River, 2005).

1.2.4 Avantages de l'architecture SOA

L'application des concepts expliqués précédemment permet de profiter de tous les avantages qu'on peut identifier autour d'une architecture orientée services (Elr T., 2008), à savoir :

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

1.2.5 La gouvernance SOA

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

Les dimensions multiples à considérer

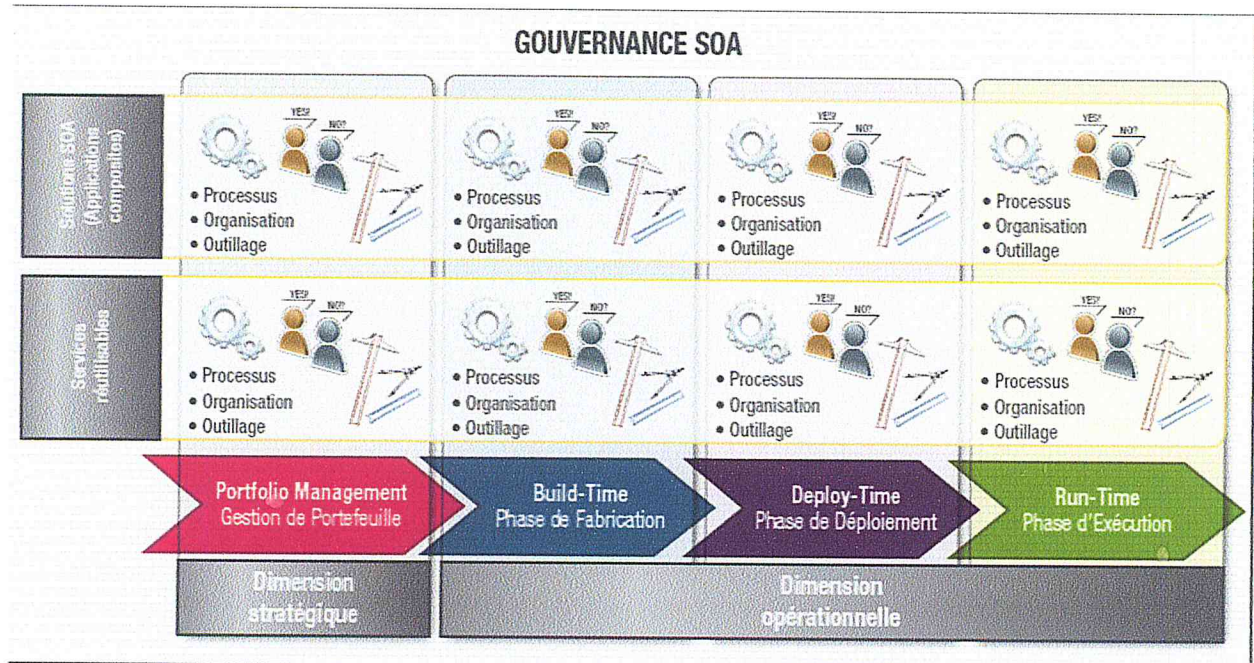
La gouvernance SOA recouvre un périmètre d'application très large, qui présente deux dimensions majeures : la stratégie et l'opérationnel.

Sur l'axe stratégique : le programme de gouvernance SOA doit proposer un cadre -- pour la gestion de portefeuilles de solutions SOA (connues sous le nom d'applications composites).

Sur l'axe opérationnel : il est nécessaire de contrôler et de sécuriser les trois temps forts de l'existence des solutions SOA et des services. Cela nécessite la mise en place de processus spécifiques de gestion des cycles de vie pour les solutions SOA et pour les services.

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

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



Matrice de synthèse des différents sujets à prendre en compte au niveau du programme de Gouvernance SOA

Figure 1:3 La gouvernance SOA (Vendeville, 2011)

1.3 Les services et l'architecture SOA

La notion de service est assez générale et existe dans plusieurs domaines métier et technologique. Par conséquent, la signification du terme varie selon le domaine et le contexte étudiés :

Dans le domaine de l'informatique et des solutions applicatives, l'Open Group définit les services comme étant « *une représentation logique d'une activité métier répétitive* » (Group, 2010). Dans ce même domaine, O. Perrin décrit les services comme étant « *les fonctions d'une application offertes sur le réseau. Ces services*

sont, de manière intrinsèque, distribués, hétérogènes, dynamiques et surtout faiblement couplés. (Godart , 2009).

Un service est un composant clef de l'soa. Il est basé sur des opérations qui offrent un ensemble d'actions spécifiques utilisables au sein de l'soa. En effet, ces opérations fournissent une ou plusieurs réponses suite à une interrogation à l'aide d'une requête.

Chaque service doit être indépendant des autres contextes ou services externes afin de garantir sa réutilisabilité et son interopérabilité dans le système d'information.

1.3.1 Les types de service

Il existe de type de service dans une entreprise (Daehene, 2009) :

1.3.1.1 Les services métiers

Il offre un service de type particulier et il correspond à un périmètre fonctionnel que l'on souhaite exposer à des consommateurs indépendamment des choix d'architecture applicative.

Les services métiers sont divisés en trois principaux groupes :

- ✓ **Service de persistance** : qui a le rôle de créer, rechercher, mettre à jour et supprimer de l'information des référentiels du SI
- ✓ **Service fonctionnel** : sert à encapsuler tout ou partie des règles de gestion et des traitements métier. Il s'appuie en général sur des services de persistance pour accéder aux informations métier dont il a besoin. Le service fonctionnel est caractérisé par une granularité moyenne à forte et une réutilisabilité moyenne à forte.
- ✓ **Service applicatif** : Sert à coordonner les appels aux services fonctionnels. Sa granularité est forte et sa réutilisabilité est nulle.

1.3.1.2 Les services techniques

Ils donnent l'accès à une ressource technique donnée (Messagerie, imprimante, Base de données relationnelle...etc.). Les services techniques sont génériques, c'est-à-dire lié à une catégorie de ressources. Ils sont caractérisés par une granularité moyenne à forte et une réutilisabilité forte

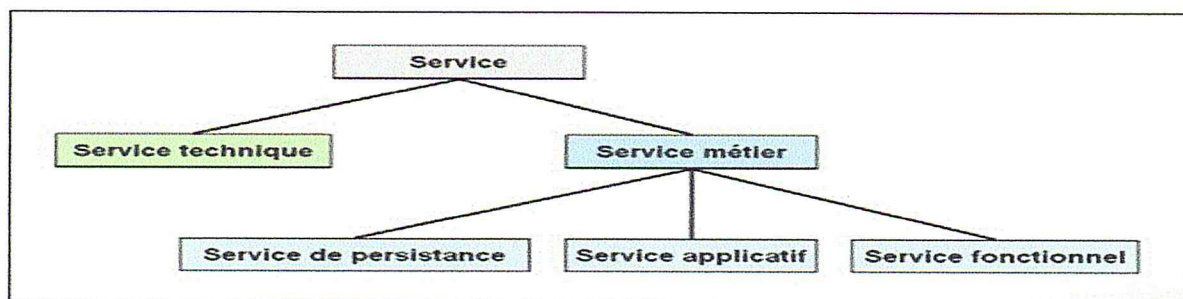


Figure 1 :4 Les types de services (Daehene, 2009) .

1.3.2 Caractéristiques des services

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

1.3.2.1 Couplage faible

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

- a- Un service ne peut pas appeler un autre service
- b- Une meilleure prise en compte de l'interopérabilité

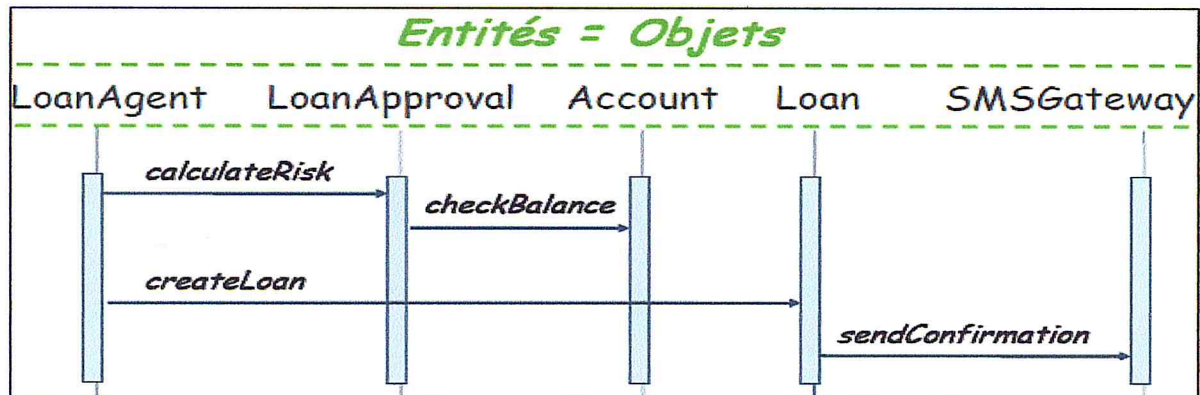


Figure 1:5 Gestion de prêt en couplage fort (Audrey, 2013) .

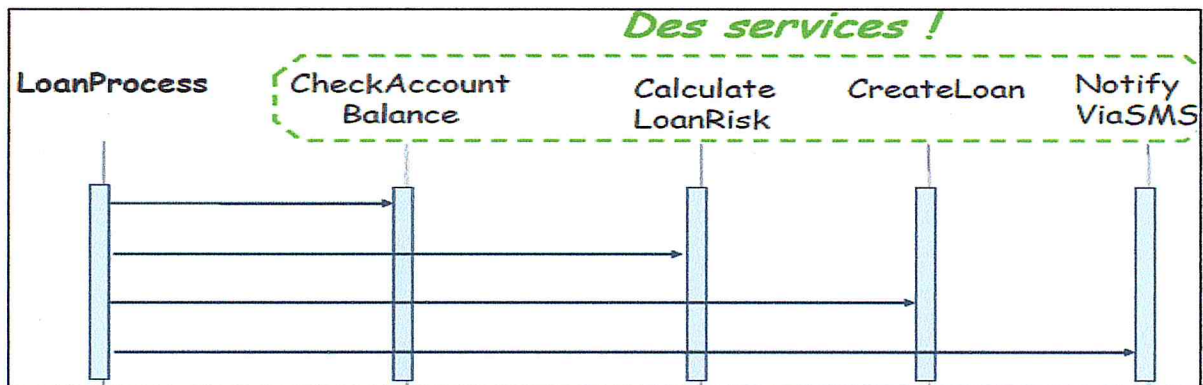


Figure 1 :6 Gestion de prêt en couplage faible (Audrey, 2013).

1.3.2.2 Découverte, sélection et consommation

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

1.3.2.3 Réutilisation

Cette propriété devra être prise en compte dès les premières phases de conception des services. En effet, les services doivent encapsuler une logique de traitement suffisamment générique pour être utilisés dans des contextes d'utilisation différents (Erl & Upper Saddle River, 2005). Cependant, le degré de réutilisation varie selon la portée du service et de sa granularité.

1.3.2.4 Le contrat des services

Le contrat des services permet de décrire les services d'une façon normalisée. Il permet aux consommateurs de juger de l'utilisabilité du service pour le sélectionner. De plus, il permet aux consommateurs de connaître les capacités du service, ses exigences et la manière dont il fonctionne.

Dans (Erl T. , 2008), T. Erl définit le contrat des services comment un ensemble de documents contenant :

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

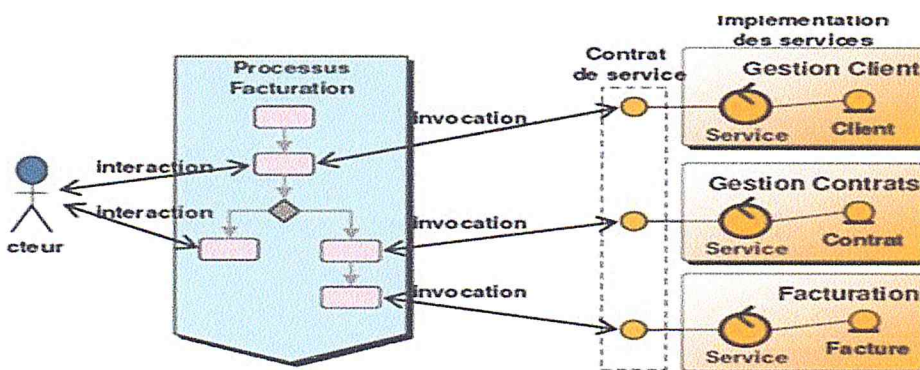


Figure 1 :7 Le contrat des services.

1.3.2.5 Granularité

Un enjeu de la SOA est de trouver la bonne granularité des services proposés par une application. En effet, un service à granularité trop fine n'offre que peu d'intérêt au niveau métier. Des services renvoyant uniquement le nom d'un client, ou juste son adresse n'ont guère de valeur ajoutée. Des services à granularité plus forte, créés à partir de plusieurs composants structurés, offrent plus d'intérêt dans la réalisation d'un processus global. (Adrien, 2008).

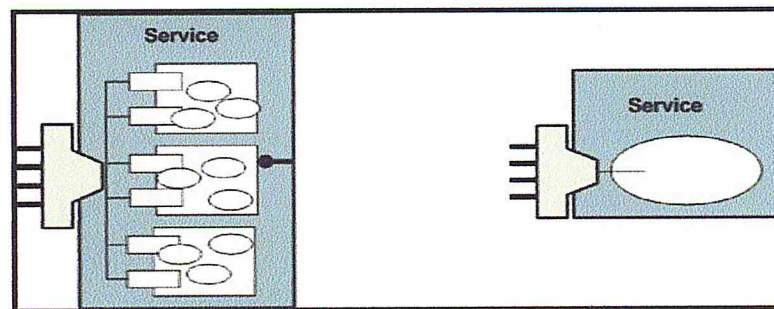


Figure 1 :8 Service à forte granularité vers service à faible granularité (Adrien, 2008).

1.3.2.6 Autonomie

Lors d'un travail d'architecture applicative, les services sont identifiés en décomposant les traitements métier. Ces services qui forment l'inventaire des services, devront être autonomes. L'autonomie hérite de la propriété 'couplage faible' et signifie que le service peut évoluer indépendamment des autres services de l'inventaire. Nous distinguons l'autonomie dans la phase de la conception et celle de la phase d'exécution.

Après avoir présenté les définitions et les propriétés des services, nous abordons dans la section suivante les architectures orientées services.

1.4 Les services web et l'architecture SOA

Nous distinguons une application qui utilise des services web et une application basée sur une architecture orientée services. En effet, les services Web représentent l'implémentation la plus utilisée des SOA à travers le Web. Il existe cependant d'autres implémentations SOA telles que les applications à base de composants EJB (Enterprise Java Beans).

Une SOA est un modèle de conception basé sur le concept d'encapsulation de la logique de l'application par des services interagissant via des protocoles de communication communs. (Erl T., 2004).

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

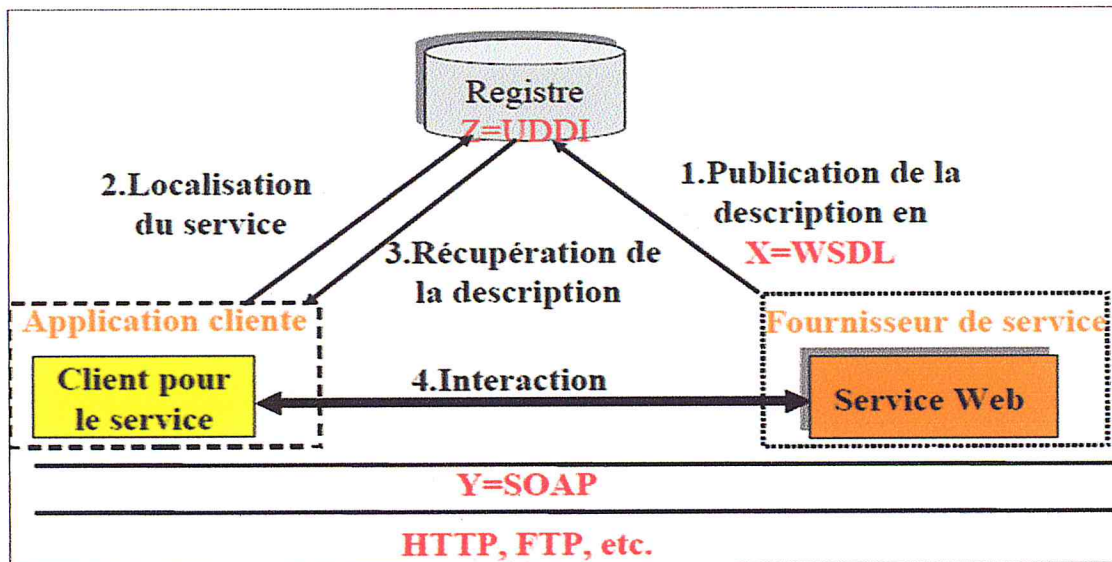


Figure 1:9 Modèle opérationnel de l'architecture orientée services.

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

1.4.1 Définition des services web

Un service web est une agrégation de fonctionnalités publiées pour être utilisées. Il utilise Internet comme conduit pour réaliser une tâche. Il est semblable à un processus métier virtuel qui définit des interactions au niveau application." (Edmond & Ter Hofstede , 2002).

Un service web est un système logiciel conçu pour supporter les interactions entre application à travers le réseau. Les services web offrent un moyen standard d'interopérabilité entre différentes applications qui s'exécutent sur une variété de machine/plateforme. Ils sont caractérisés par leur grande interopérabilité et extensibilité, ainsi qu'une description interprétable automatiquement par la machine grâce au standard XML. Ils peuvent être combinés d'une façon faiblement couplée

afin de réaliser des opérations complexes. Les programmes offrant des services simples, peuvent interagir ensemble afin de mettre en place des services sophistiqués avec des valeurs ajoutées.

1.4.2 Les standards de l'architecture orientée service web

✦ **Transport : SOAP (Simple Object Protocol communication)**

SOAP est un protocole de transmission de messages fondé sur XML pour l'échange d'informations entre systèmes hétérogènes en environnement décentralisé. Il définit un ensemble de règles pour structurer des messages principalement pour exécuter des dialogues requête /réponse.

✦ **Découverte : UDDI (Universal Description Discovery and Integration)**

C'est un annuaire de publication de services, il se repose sur le protocole SOAP et de ce fait, les requêtes et les réponses sur l'annuaire sont des messages SOAP, UDDI comporte plusieurs catégories de données :

- ✓ **Pages blanches** : elles recensent les entreprises et contiennent des informations telles que le nom de l'entreprise, ses coordonnées, des descriptions accessibles aux clients, et des identifiants permettant de la retrouver par recherche.
- ✓ **Pages jaunes** : elles contiennent, au format WSDL, la description des services déployés par l'entreprise. Les services sont répertoriés par catégories.
- ✓ **Pages bleues** : elles fournissent des informations techniques détaillées sur les services

✦ **Description : WSDL (Web Services Description Language)**

C'est une interface qui cache le détail de l'implémentation du service, permettant une utilisation indépendante de la plate-forme et de langage utilisés. Il regroupe au format XML toutes les informations nécessaires pour interagir avec le web service. Il y a deux types de documents WSDL :

- ✓ Le document WSDL décrivant l'interface du service.
- ✓ Le document WSDL décrivant l'implémentation du service.

L'infrastructure des services web s'est concrétisée autour de trois spécifications considérées

Comme des standards, à savoir SOAP, UDDI et WSDL . Nous les détaillons dans les sections suivantes.

1.5 Les éléments de base du SOA

Ils doivent être compris de deux niveaux, technique et métier (organisationnel).

1.5.1 Eléments organisationnels d'une SOA

Selon F.Vernadat, (Touzi, 2007) les éléments métiers du SOA sont définis comme suit :

1.5.1.1 Service métier

C'est une fonctionnalité de l'entreprise qui apparaît atomique du point de vue de l'appelleur de service. Un service doit être identifié d'une manière unique dans l'entreprise.

1.5.1.2 Evènement métier

C'est un fait qui se produit en relation avec les opérations de l'entreprise. C'est un changement dans l'état de l'entreprise, qui doit avoir une réponse. Il existe plusieurs types d'évènements : un évènement sollicité (réception d'une nouvelle requête, etc.), un évènement d'exception (panne de machine, etc.), évènement programmé (délai de temps passé, etc.) Ou un évènement de synchronisation (début ou fin d'une activité)

1.5.1.3 Processus métier

Un processus métier est un séquençement partiellement ordonné d'un ensemble d'activités et/ou de services dans le but de réaliser un objectif de l'entreprise.

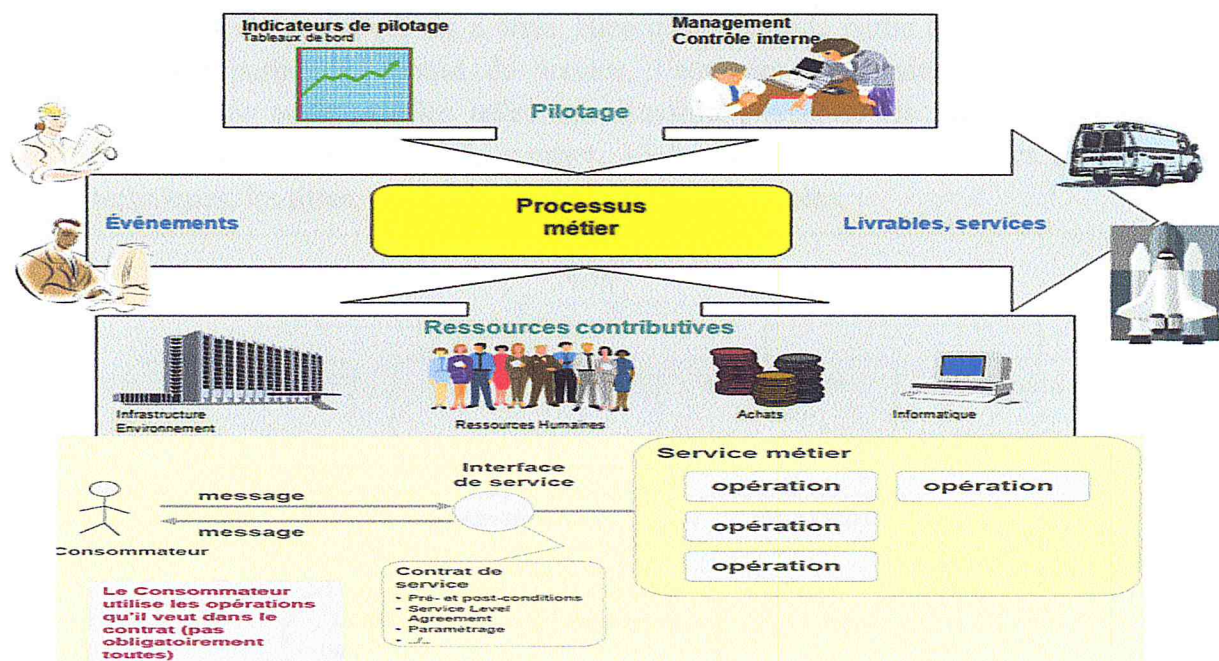


Figure 1:10 Service métier, concept d'opération (Bourey, 2007).

1.5.2.4 Infrastructure de SOA (ESB)

Un ESB est parfois considérée comme une nouvelle génération d'EAI (Intégration d'applications d'entreprise) Tout service pourra être branché sur le bus de service pour faire partie d'un écosystème de services.

L'ESB doit répondre aux fonctionnalités suivantes :

- ✦ **Connectivité** : Le but primaire de l'ESB est de relier l'ensemble des éléments de SOA
- ✦ **Hétérogénéité de technologie** : la réalité des entreprises est caractérisée par des technologies hétérogènes. En conséquence, l'ESB doit pouvoir relier les composants qui sont basés sur différents langages de programmation.
- ✦ **Hétérogénéité des concepts de communication** : en raison des conditions divergentes de différentes applications, l'ESB doit permettre différents modes de communication.
- ✦ **Les caractéristiques techniques** : bien que le but de l'ESB soit principalement communication, il doit également fournir des fonctionnalités techniques.

Pour bien expliquer ce concept Yann (Yann, 2007) propose le schéma suivant

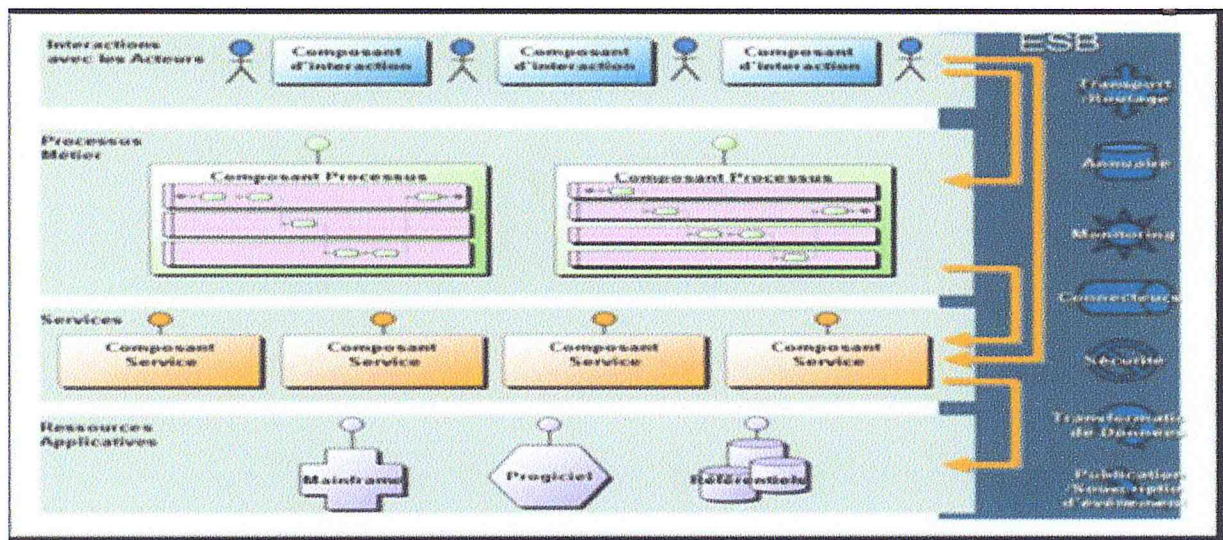


Figure 1:12 : Intégration de l'ESB dans les principes d'une SOA. (Yann , 2007)

Conclusion

L'architecture orientée service (SOA) qui est une approche architecturale de dernière génération qui permet de passer d'une vision « application » à une vision « services », elle intervient au niveau logique. Au niveau logique, il s'agit d'identifier, sur le plan organisationnel, les fournisseurs et consommateurs d'informations et d'identifier, sur le plan fonctionnel, les différentes unités fonctionnelles pouvant être transformées en services.

Chapitre 2 : Méthodologie de développement SOA

2.1 Introduction

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

2.2 Stratégies de la conception des architectures SOA

2.2.1 Top Down

L'approche « top down » suit une logique descendante. Elle consiste à commencer l'intégration par la définition des processus métiers puis à la définition des services métiers nécessaire au fonctionnement des processus. Cette logique est poursuivie dans le but d'aligner le système d'information sur le métier de l'entreprise [Erl 2006]]. Cette approche permet de piloter l'architecture par les besoins métier tout en minimisant les redondances des services [Xebia IT Architects, 2007].

2.2.2 Bottom Up

La seconde stratégie d'intégration, nommé « bottom up » opère dans le sens inverse de la précédente approche. Elle est réalisée dans les entreprises possédant déjà un système d'information. La conception se déroule de manière ascendante en commençant par identifier les fonctions existantes du SI que l'on peut réutiliser en tant que service [Erl 2006]. En se basant sur un système déjà présent, l'entreprise peut récolter un gain sur les coûts d'implémentation. Ce type d'intégration est meilleur marché qu'une intégration top down. [[Xebia IT Architects, 2007].

2.2.3 Outside In

Cette approche préconise de mener en parallèle :

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

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

2.2.4 Middle Out

Cette approche peut être introduite par la phrase « BPM et SOA sont les deux faces de la même pièce ». Assertion qui fait référence à un autre débat stérile ayant fait couler beaucoup d'encre et qui rappelle qu'il est impossible de dissocier la définition des processus métiers de l'établissement du catalogue de services.

Par opposition à l'approche Outside In, cette méthode propose de commencer « In the middle », c'est-à-dire là où le métier et les IT parlent le même langage (ou en tout cas presque). Elle s'attaque donc d'emblée à ce qui reste un des principaux freins à l'adoption des SOA au sein des DSI françaises : La compréhension du métier de l'entreprise par les maîtrises d'œuvre et inversement, la compréhension des contraintes IT par les maîtrises d'ouvrage.

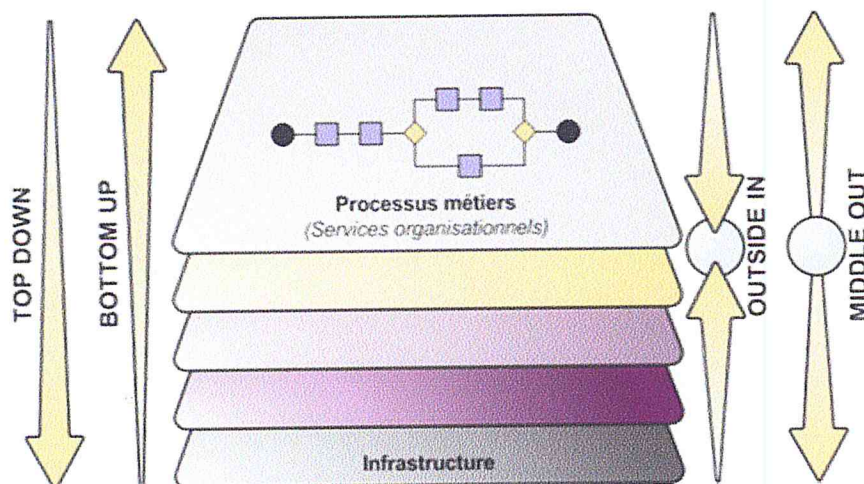


Figure 2:1 Stratégie d'intégration (Xebia IT Architects, 2007).

2.3 Les méthodologies de développement des SOA

Plusieurs méthodologies ont été proposées pour concevoir et mettre en œuvre des SOA

2.3.1 SOMA (Service-Oriented Modeling and Architecture) d'ibm

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

2.3.1.1 La phase d'identification des services

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

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

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

2.3.1.2 La phase de spécification

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

2.3.1.3 La phase de réalisation

Cette phase se charge d'implémenter l'ensemble des services. Le schéma si dessous illustre Toutes les étapes de SOMA.

un livrable clé SOMA, qui couvre la syntaxe et les sémantiques de l'invocation de service, ainsi que d'autres caractéristiques opérationnelles tel que la possession de service, les dépendances et le visionnement (Zimmermann, Pal , & Clive , 2004). La réalisation des services et des composants est assistée par des outils et des techniques telles que les *patterns* (par exemple les *patterns* d'ibm pour le *e-business* peuvent être utilisés).

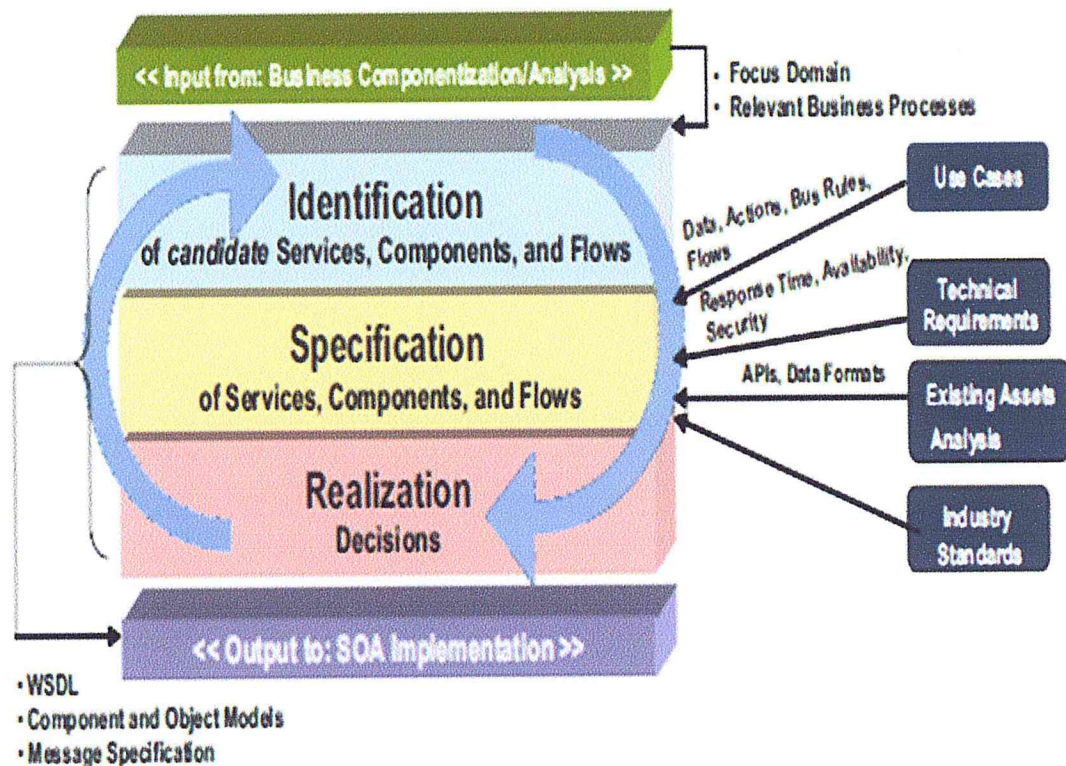


Figure 2:2 la méthodologie SOMA. (Arsanjani, Ghosh, Allam , & Abdollah, 2008).

2.3.2 SOAF (Service Oriented Architecture Framework)

La méthodologie SOAF (Service Oriented Architecture Framework) est une méthodologie réalisée par (Erradi, Anand, & Kulkarni , 2006). Contrairement aux méthodes propriétaires précédentes, cette méthodologie détaille les artéfacts pour chacune des phases. C'est à première vue la méthodologie disponible la plus complète. SOAF se décompose en cinq phases.

2.3.2.1 Extraction de l'information

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

2.3.2.2 Identification des services

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

2.3.2.3 Définition des services

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

2.3.2.4 Réalisation des services

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

2.3.2.5 Feuille de route et planification

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

La Figure 1.13 SOAF se réfère à des artefacts liés à des activités d'architecture d'entreprise (applications existantes, portefeuille, modèles et bonnes pratiques) sans pour autant donner davantage de détails sur la réalisation de ces activités. Cette méthodologie n'est enchâssée ni dans une méthodologie de développement complète ni dans un cycle incluant des disciplines d'entreprises.

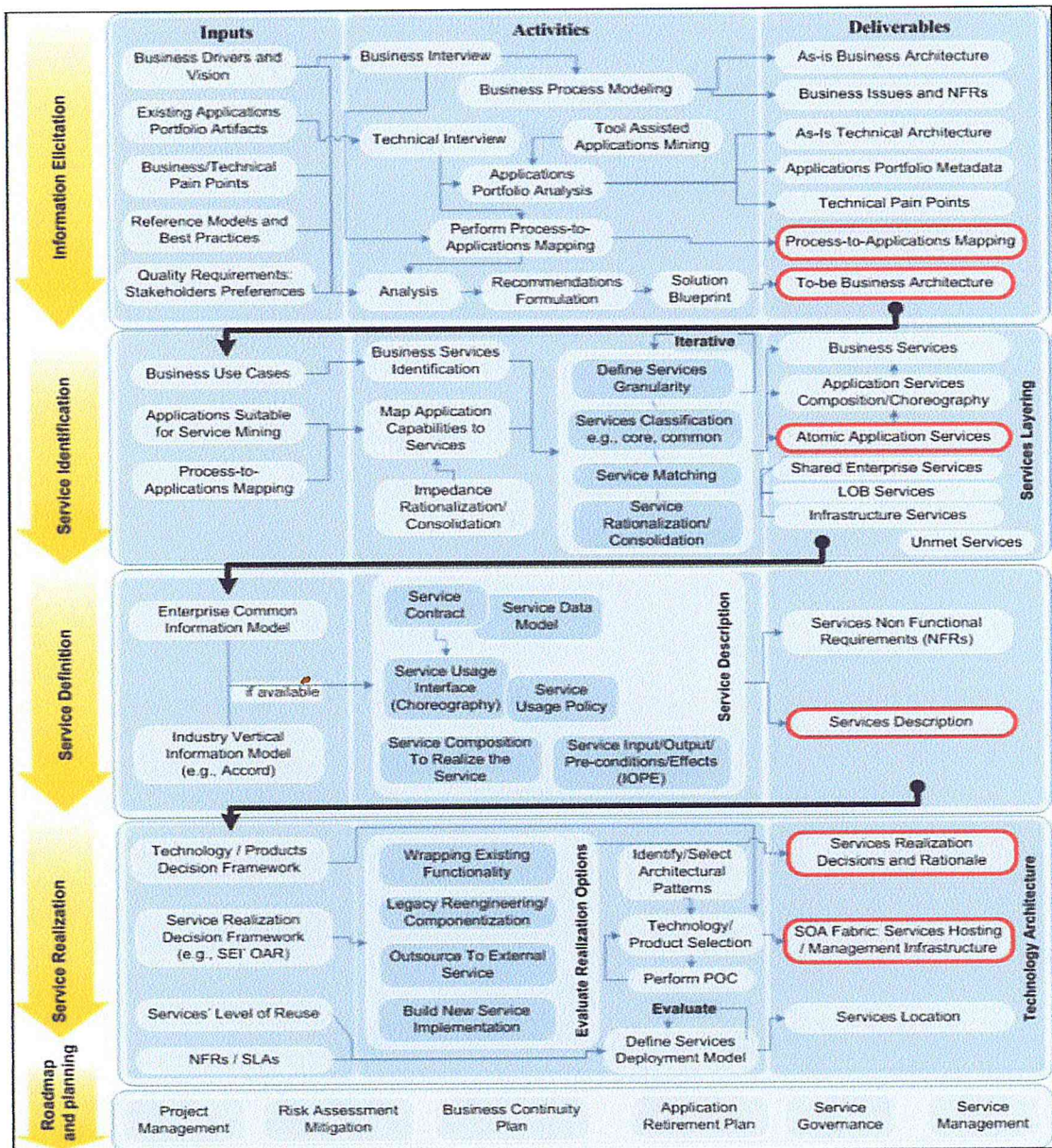


Figure 2 :3 La méthodologie SOAF (Erradi, Anand, & Kulkarni , 2006).

2.3.3 La méthodologie CBDI-SAE

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

2.3.4 La méthodologie SODM

Dans (Papazoglou & Van Den Heuvel , 2006), Papazoglou propose une méthodologie de développement de service (SODM) qui couvre la totalité du cycle de vie des services, tout en tenant compte des points de vue du fournisseur et des consommateurs des services. Cette méthodologie se base sur des modèles de développement comme le RUP (Rational Unified Process), la CBD (Component Based Développement) et le BPM (Business Process Modelling). La méthodologie utilise un processus itératif et incrémental incluant une phase préparatoire et huit phases principales (l'analyse, la conception, la construction, les tests, l'approvisionnement, le déploiement, l'exécution et la supervision des services)

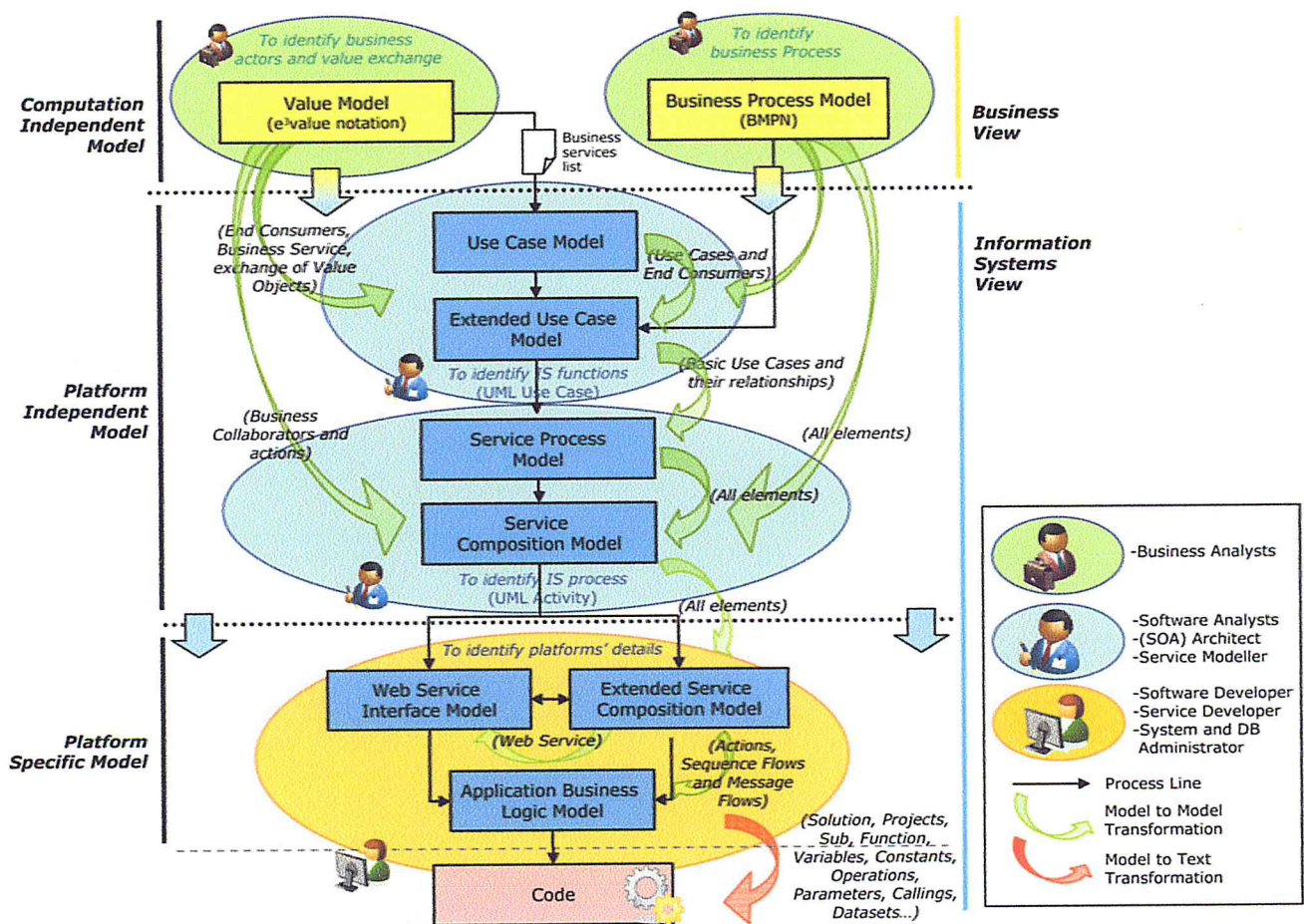


Figure 2:4 La méthode SOD-M.

2.3.5 La méthodologie de Thomas Erl

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

2.3.5.1 Analyse orientée service

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

2.3.5.2 Conception orientée services

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

2.3.5.3 Développement du service

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

2.3.5.4 Test du service

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

2.3.5.5 Déploiement du service

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

2.3.5.6 Administration du service

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

2.3.6 La méthodologie de S. Jones

Dans (Jones & Morris, 2005), S. Jones présente une méthodologie de conception qui met en évidence les propriétés métier d'une architecture orientée services. Il souligne

L'importance des concepts métier et se focalise sur les éléments {Qui, Quoi, Comment et Pourquoi ; dans L'identification des services métier selon une approche descendante. Toutefois, l'auteur n'aborde pas les aspects technologiques des architectures orientées services.

2.3.7 La méthodologie de Zimmerman SOAD

(Zimmermann, Pal , & Clive , 2004)proposent une approche interdisciplinaire, nommée SOAD (Service-Oriented Analysis and Design) pour *Service Oriented Analysis and Design*, En tant que discipline de modélisation qui est construite sur les fondations bien établies de l'analyse et de la conception orientée objet (*Object Oriented Analysis and Design*, OOAD), l'architecture d'entreprise (*Enterprise Architecture*, EA) et la gestion des processus d'affaires (*Business Process Management*, BPM). La figure ci-dessous illustre l'approche SOAD et les éléments sur lesquels elle se base. (Boukadi & Khouloud, 2009) .

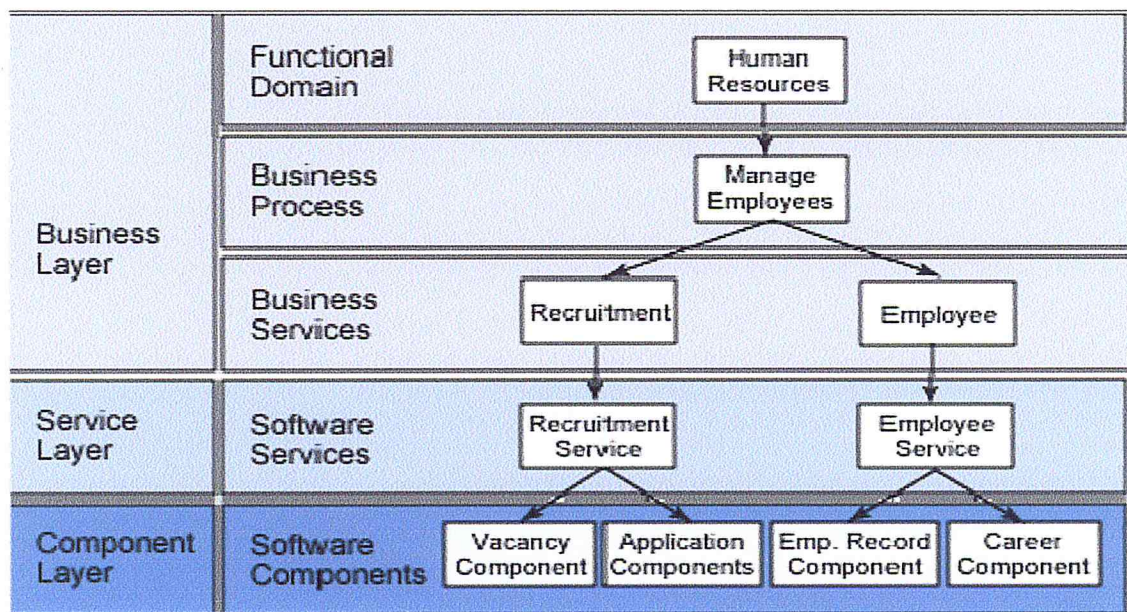


Figure 2:5 La méthodologie de SOAD (Zimmermann, Pal , & Clive , 2004).

(Zimmermann, Pal , & Clive , 2004) Considèrent que les trois disciplines de modélisation OOAD, EA et BPM fournissent des pratiques de haut niveau qui peuvent être combinées pour fournir le support requis au paradigme SOA. Mis à part la reprise de notions déjà existantes, SOAD a apporté des innovations en ce qui concerne les répertoires de service,

L'orchestration de service et le bus de services d'entreprise. Cependant, les notations et les processus de SOAD restent à être définis en détail.

2.3.8 La méthode SOA⁺

SOA⁺ est une démarche méthodologique à quatre phases mettant en exergue une démarche de construction de l'architecture de services au sein de l'entreprise : la construction de la SOA métier, la construction de la SOA Information et la construction de la SOA Décision. Chaque phase de la méthode SOA⁺ peut comporter une ou plusieurs étapes et peut être Réalisée soit par un intervenant métier, un intervenant d'information, ou soit par un intervenant de décision soit par les trois au même temps.

La méthode proposée par (Boumahdi, et al., 2014) suit une approche descendante pour découvrir les services intervenant dans SOA⁺. Comme nous montre **figure 2.5**, SOA⁺ s'articule autour de quatre phases : analyse, identification et catégoriser services catégorisés, modélisation de services et la phase de réalisation.

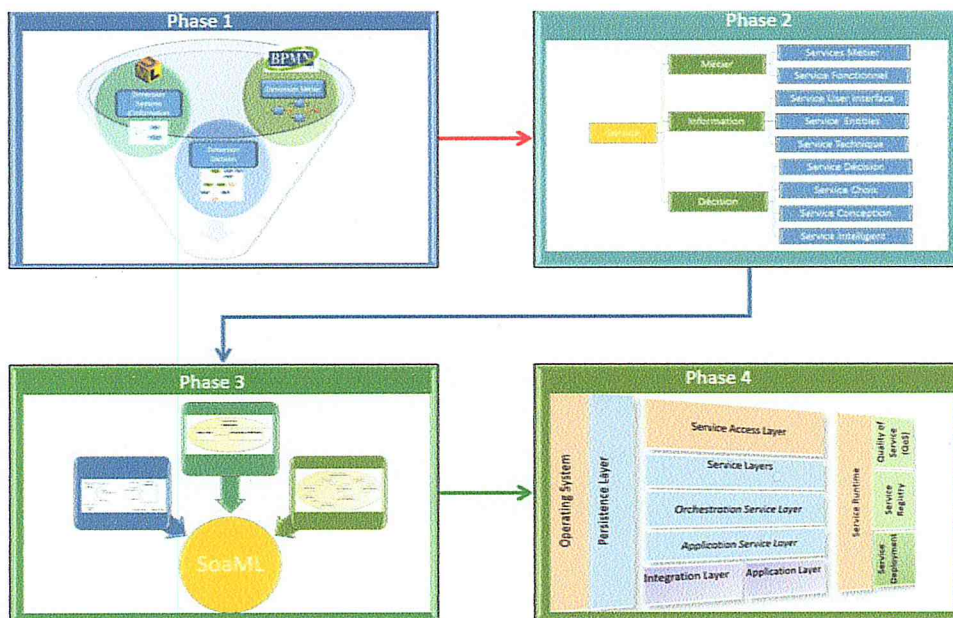


Figure 2:5 La méthodologie de SOA⁺. ((Boumahdi, et al., 2014).

2.4 Comparaison des méthodologies de SOA

Pour comparer les différentes méthodologies de conception, nous avons retenu différents critères : (Pascal Bou Nassar, 2012)

Critère 1 : Flexibilité

Ce critère décrit la capacité de la méthode à adapter l'architecture orientée services aux nouveaux besoins métier. Ce critère nous permet de différencier les méthodologies qui utilisent des approches agiles dans le développement pour plus de flexibilité des autres qui adoptent des approches plus rigides limitant leur flexibilité.

Critère 2 : Accessibilité

L'accessibilité dénote l'accès aux détails de la spécification de la méthode. Ce critère nous permet de différencier les méthodologies propriétaires dont les spécifications ne sont pas accessibles et celles non propriétaire qui sont accessibles en détail.

Critère 3 : Approche d'identification des services

Ce critère est en rapport avec les approches utilisées dans la conception des architectures orientées services : Approche Ascendante (AS), Approche Descendante (AD), Approche 'Outside In' (IO) et Approche 'Middle Out' (MO). Ce critère nous permet de déterminer comment la méthodologie aborde l'identification des services et la réutilisation des ressources.

Critère 4 : Portée

Ce critère décrit les phases du cycle de vie des services couvertes par la méthodologie. Dans notre travail, nous proposons d'intégrer la sécurité dans la totalité des phases du cycle de vie des services. Par conséquent, il est important de déterminer si les méthodologies couvrent la totalité de ce cycle de vie, ou bien se limitent à des phases particulières.

Critère 5 : Richesse de la méthodologie

Ce critère dénote la richesse de la méthodologie en termes de guides d'utilisation, de directives et de bonnes pratiques pour évaluer l'autonomie de la méthodologie et déterminer si elle présente un support facilitant la conception des services.

Critère 7 : Prise en compte de l'aspect décisionnel

Ce critère décrit la capacité de la méthode à adapter l'architecture orientée services aux nouveaux besoins métier qui est l'aspect décisionnel. Ce critère nous permet de différencier les méthodologies qui utilisent l'aspect décisionnel et de l'absence dans les autres méthodologies.

La comparaison des méthodologies selon ces critères est résumée dans le Tableau 2-2 (Pascal Bou Nassar, 2012).

| critère | Méthodologies de conception | | | | | | |
|--|-----------------------------|-----------------------|-----------------------|-----------------------|--|-----------------------|----------------------|
| | SOMA | CBD ISAE | SOAF | SODM | T.ERL | S. JOHNS | SOA+d |
| Flexibilité | ++ | +++ | + | ++ | + | SO | +++ |
| Accessibilité | Non accessible | Non accessible | Version2 oui | accessible | accessible | accessible | accessible |
| Approche d'identification des services | Approche Out side -In | Approche Out side -In | Approche Out side -In | Approche Out side -In | Approche Descendante | Approche Descendante | Approche Descendante |
| Portée | Analyse et conception | Tous les cycle de vie | Analyse et conception | Tous les cycle de vie | Analyse et Conception et implémentat ion | Analyse et Conception | Tout le cycle de vie |
| Richesse de la méthodologie | +++ | +++ | ++ | ++ | +++ | + | +++ |
| Aspect décisionnelle | NON | NON | NON | NON | NON | NON | OUI |

+ : niveau mineur de flexibilité (capacité de la méthode à adapter l'architecture orientée services aux nouveaux besoins métier) / richesse de la méthode en termes de guides d'utilisation, de directives et de bonnes pratiques.
 ++ : niveau moyen de flexibilité / richesse
 +++ : niveau majeur de flexibilité / richesse
 SO : Sans Objet (Le critère ne s'applique pas)

Tableau 2-1 La comparaison des méthodologies selon des critères.

On remarque que CBDI-SAE et SODM couvrent la totalité des phases du cycle de vie des services.

- ✦ CBDI-SOA est une méthodologie assez riche. Toutefois, elle n'est pas accessible et ne prend pas en compte la sécurité.
- ✦ La méthodologie SODM est accessible, mais ne prend pas en considération l'aspect décisionnel.

Les méthodologies restantes se limitent à l'étude d'une phase particulière du cycle de vie des services et ne prennent pas en compte les aspects de la décision.

Cette étude montre qu'aucune des méthodologies étudiées n'est complète et ne répond à nos besoins pour développer des architectures orientées services qui prennent en charge l'aspect décisionnel. Pour cela, nous proposons de développer une méthodologie permettant d'assurer la conception et le déploiement des architectures orientées services décisionnel

Conclusion

Les architectures orientées services représentent un style d'architecture permettant d'améliorer l'agilité du système d'information. Cependant la mise en place d'une telle architecture nécessite une préparation approfondie dans l'objectif d'atteindre la finalité d'une SOA qui est l'alignement métier et applicatif.

Dans ce deuxième chapitre, nous avons abordé les concepts, les modèles et les standards dans le domaine des architectures orientées services. Nous avons d'abord recherché des modèles de référence permettant de définir les concepts d'une SOA, puis des architectures de référence, des standards et des technologies permettant de réaliser une SOA. En abordant les concepts autour de la SOA, nous avons pu identifier des solutions intéressantes permettant d'améliorer l'architecture SOA et d'introduire l'aspect décisionnel

Chapitre 3 : La méthode SOA^{+d}

3.1 Description de la méthode SOA^{+d}

SOA^{+d} est une démarche méthodologique à quatre phases mettant en exergue une démarche de construction de l'architecture de services au sein de l'entreprise : la construction de la SOA métier, la construction de la SOA Information et la construction de la SOA Décision. Chaque phase de la méthode SOA^{+d} peut comporter une ou plusieurs étapes et peut être réalisée soit par un intervenant métier, un intervenant d'information, ou soit par un intervenant de décision soit par les trois au même temps.

La méthode proposée par (Boumahdi, et al., 2014) suit une approche descendante pour découvrir les services intervenant dans SOA^{+d}. Comme nous montre la figure ci dessous, SOA^{+d} s'articule autour de quatre phases : analyse, identification et catégoriser services catégorisés, modélisation de services et la phase de réalisation.

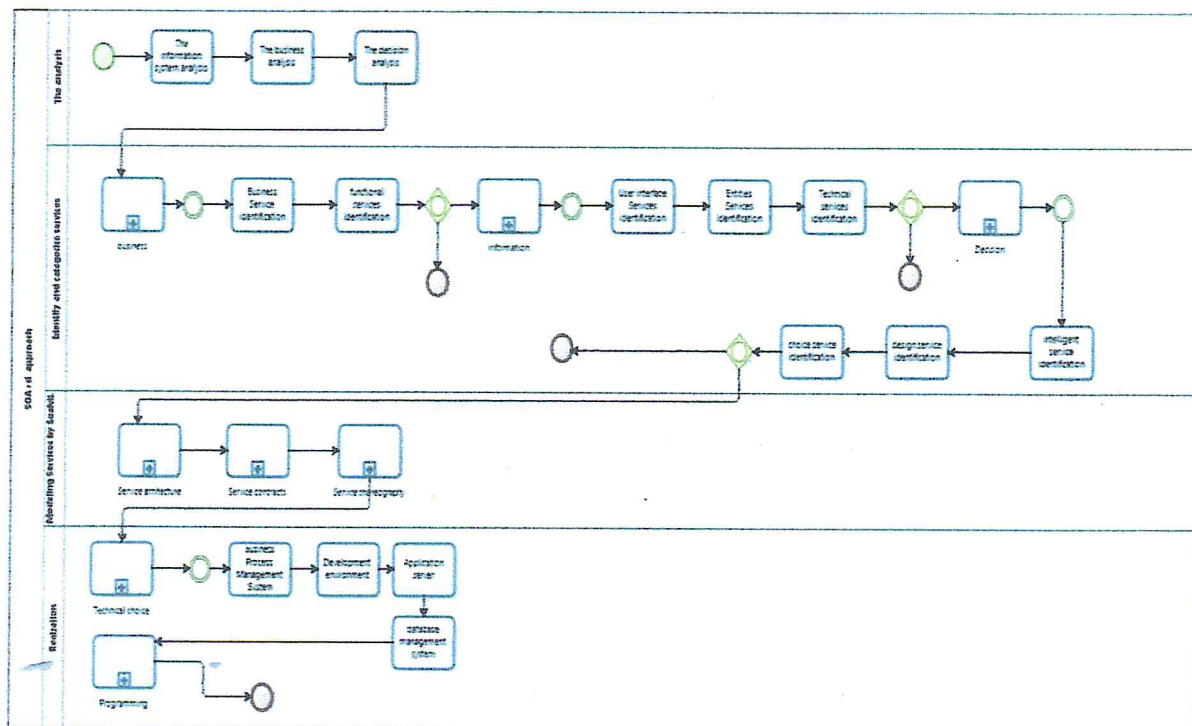


Figure 3:1 La description des phases de SOA^{+d} avec la notation BPMN

3.2 Concepts de base de l'approche SOA+d

Dans la section suivante, la notation DMN et le langage SoaML seront présentés. Ces concepts sont utilisés pour définir l'approche SOA+d.

3.2.1 Language SoaML (Service Oriented Architecture Modeling Language)

Depuis le mois de Janvier 2009, Services Oriented Architecture Modeling Language (**Soaml**) est devenu une spécification de l'Object Management Group (**OMG**), qui fournit un Métamodèle pour la conception de services au sein d'une architecture orientée services.

La spécification étend Unified Modeling Language (**UML2.0**), permettant ainsi une approche visuelle et sémantiquement rigoureuse pour la définition des services. Le profil UML pour **Soaml** présente de l'intérêt pour les analystes métier, les architectes et les promoteurs tout au long du processus d'établissement d'une architecture orientée services.

Soaml se base sur un ensemble de composants qui s'adaptent au niveau métier de l'entreprise :

- ✦ **Les participants** : sont utilisés pour définir les fournisseurs et les consommateurs de services dans un système, ils peuvent jouer le rôle d'un fournisseur de service, d'un consommateur de service ou les deux.
- ✦ **L'architecture de services** : est utilisée pour définir la manière dont les participants s'échangent les services.
- ✦ **Les contrats de services** : définis par des interfaces les contrats sont utilisés pour décrire les modèles d'interaction entre les participants. Un contrat est utilisé pour modéliser un accord entre deux ou plusieurs parties.

2.1.1. Objectif du langage SoaML

Le but de Soaml est en effet de fournir aux utilisateurs du langage UML (Unified Modeling Language) les moyens de modéliser une architecture orientée services - comprenant donc des notions de consommateurs et de fournisseurs de services, ainsi que la notion de contrats.

- 💧 Sépare les préoccupations de l'architecture administrative de la technologie mise en œuvre et de la réalisation.
- 💧 SoaML prend en charge une gamme complète de modélisation de service, le contrat de service et d'interface de service ainsi que la spécification du service lui-même.
- 💧 Compatibilité entre UML et BPMN du processus métiers
- 💧 Modélisation de haut en bas, bas vers le haut ou meet-in-the-middle.

2.1.2. Les approches de Soaml

On dispose de trois approches pour identifier les services intervenants dans un système (Casanave, 2012) :

- ✦ Concevoir le système comme étant une communauté de participants (fournisseurs et consommateurs de services) et les contrats qui spécifient la manière avec laquelle ils devront interagir.
- ✦ Organiser de manière hiérarchique un certain nombre de fonctionnalités mutualisées en habilités d'un service. La hiérarchie doit refléter les dépendances.
- ✦ S'inspirer de la logique métier en termes de processus pour identifier les habilités nécessaires pour accomplir une tâche particulière ainsi que les rôles joués par chaque participant.

3.2.2 Modèle de décision et de notation

Le Groupe de gestion d'objets a récemment (01-02-2014) adopté le modèle De précision et de notation en tant que norme de bêta. Le but de DMN est de normaliser les notations (et méta modèle associé) pour la modélisation des décisions. DMN fournit une construction couvrant les exigences en matière de décision et de la modélisation de la décision logique.

3.2.2.1 Les exigences en matière de décision

Se compose d'une décision exigences graphique (GRD) représenté dans un plusieurs schémas de prescription décision (DRA). Un GRD modélise un domaine de la prise de décisions, montrant les éléments impliqués les dépendances entre eux (OMG, 2014).

Les éléments modélisés sont des décisions, domaines de la connaissance de l'entreprise et les données saisie. La notation pour tous les composants d'un diagramme des besoins de décision (DRD) est résumée dans le tableau.

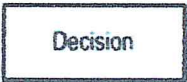
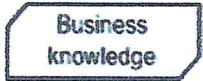
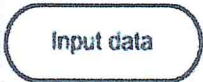
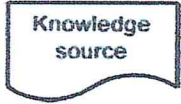



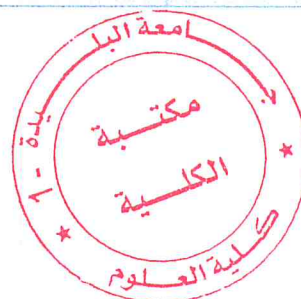
| Composants DRD | Description | Notation | |
|----------------|--|--|---|
| Eléments | <i>Décision</i> | Une décision désigne le fait de déterminer une sortie de plusieurs entrées, en utilisant la logique de décision qui peut faire référence à un ou plusieurs modèles de connaissances métiers. |  |
| | <i>Model de Connaissances D'entreprise</i> | Un modèle de connaissance de gestion désigne une fonction encapsulant les connaissances de l'entreprise, par exemple, les règles d'entreprise, une table de décision, modèle analytique ou un algorithme. |  |
| | <i>Données d'entrée</i> | Un élément de données d'entrée indique les informations utilisées comme entrée par une ou plusieurs décisions. Lorsqu'elle est placée dans un modèle de connaissance, il désigne les paramètres du modèle de connaissance. |  |
| | <i>Source de connaissances</i> | Une source de connaissances désigne une autorité pour un modèle de connaissance de gestion ou de décision. |  |
| Exigences | <i>Obligation d'information</i> | Une obligation d'information représente les données d'entrée ou une décision de sortie utilisée comme une des entrées d'une décision |  |
| | <i>Connaissances exigées</i> | Une exigence de connaissances désigne l'invocation d'un modèle de connaissance d'entreprise |  |
| | <i>Autorité Exigence</i> | Une exigence de l'autorité désigne la dépendance à l'égard d'un élément DRD sur une source de connaissances ou la dépendance d'une source de connaissances sur les données d'entrée |  |

Tableau 3-1 Les composants de DRD (OMG, 2014).



3.2.2.2 Le niveau logique de décision

Les composants du niveau d'un DRD peuvent être décrits, en utilisant seulement des concepts métier. Ce niveau de description est souvent suffisant pour l'analyse métier d'un domaine de la prise de décisions, d'identifier les décisions métier impliquées, leurs relations, les domaines de la connaissance métier et les données requises et les sources de la connaissance de l'entreprise (OMG, 2014).

Objectif de DMN :

- DMN crée un pont normalisé de l'écart entre la conception de la décision de l'entreprise et son implémentation (OMG, 2014).
- DMN, comme une spécification de l'it, il confirme qu'il y a demande d'un nouveau type de produit de logiciel destiné à la gestion et la modélisation de la décision.
- Notation commune qui est facilement compréhensible par tous les utilisateurs professionnels, des analystes métier qui doivent créer des exigences de la décision initiale et puis plus détaillée des modèles de décision, pour les développeurs techniques responsables pour automatiser les décisions dans le processus et enfin, pour les gens métiers qui permettra de gérer et de contrôler ces décisions.

Les sections suivantes présentent brièvement les concepts proposés par SOA⁺d. Ces notions sont organisées selon le métier, le système d'Information et les vues décisionnel, qui sont intégrés afin d'assurer la relation entre les processus métier, le niveau des informations système et décision.

| <i>Service de nature</i> | <i>Description</i> | <i>Style d'interaction</i> | <i>Granularité</i> | <i>Composition</i> |
|---|---|----------------------------|--------------------|--------------------|
| Niveau de l'entreprise | | | | |
| Services métier : le service est une brique réutilisable sur le plan commercial. Il correspond à des fonctionnalités métier. Il fait appel aux services fonctionnels | Service abstrait. Participer au processus métier | <i>Synchrone</i> | 4/4 | Service composite |
| Service fonctionnel : comme par exemple les services offrant les fonctionnalités liées à la gestion d'une commande de client, ou le calcul d'un montant de la commande. | Service abstrait. Orchestre plusieurs services d'informations et de décisions. | <i>Synchrone</i> | 3/4 | Service composite |
| Niveau d'information | | | | |
| Services Interface utilisateur permettent de gérer les communications et le dialogue avec les acteurs. | Service concret Encapsule les applications métiers | <i>Synchrone</i> | 3/4 | Service composite |
| Services d'entités (CRUD) : les services qui donnent accès à l'information relativement aux objets de commerce. En général, ce sont les services qui effectuent des opérations de création, lecture, mise à jour et suppression (CRUD). | Service concret Service exposée par un objet métier. | <i>Asynchrone</i> | 2/4 | Service atomique |
| Services techniques : rassembler les services de bas niveau et permettre de gérer l'infrastructure du système d'information de l'entreprise (exemple : les services liés à la gestion du réseau, courrier électronique et de l'édition). | Service concrète infrastructure de services de SI | <i>Asynchrone</i> | 1/4 | Service atomique |
| Niveau de décision | | | | |
| Décision : construire un service de décision indépendante en utilisant les méthodes, les algorithmes et techniques pour proposer une solution. | Service abstrait Participer au processus métier | <i>Synchrone</i> | 3/4 | Service composite |
| Service de conception : consistent dans la construction et étudier les solutions, à l'aide de l'information qui résulte de service intelligente. | Service concret Service expose la méthode de solution. | <i>Asynchrone</i> | 2/4 | Service composite |
| Service intelligent^{td} : est un service qui cherche les informations pertinentes en relation avec le problème à résoudre | Service concret Service inclut les exigences de la décision. | <i>Asynchrone</i> | 1/4 | Service atomique |

Tableau 3-2 Résumé des différents Services de SOA^{td} et leurs caractéristiques.

SOA^{+d} définit un ensemble de concepts organisés selon trois points de vue :

- Métier, mettant l'accent sur les caractéristiques et les besoins de l'entreprise dans laquelle l'it sera construite.
- La vue de système d'informations, se concentré sur les fonctionnalités et les processus qui doivent être mis en œuvre pour satisfaire les exigences de l'entreprise et de la décision dans le SI.
- Décision : recueillir des fonctionnalités offertes par le cadre décisionnel du processus métier et ils sont utilisés par le système d'information.

3.3 Principes de SOA+d

SOA+d repose sur un ensemble de principes fondateurs :

3.3.1 Principe d'ouverture

Il impose de s'inscrire dans le cadre d'utilisation des standards. En effet, SOA^{+d} s'adresse aux petites et moyennes entreprises pour lesquelles, il est indispensable de proposer une solution basée sur des standards. L'utilisation des standards constitue une boîte à outils à laquelle s'ajoute une démarche guidant la construction de l'architecture de services à chaque étape. En effet et comme nous montre la figure les standards UML, BPMN, DMN et soaml seront utilisés dans le cadre de la méthode SOA^{+d}.

3.2.2 Réutilisation de l'existant

Le patrimoine applicatif de l'entreprise, résulte de l'empilement de générations successives d'applications, de la stratification des logiciels, comportant des redondances et manquant de cohérence. La complexité croissante de ce patrimoine existant génère des difficultés de plus en plus grandes pour faire évoluer le système d'information en adéquation avec les attentes des différents métiers de l'entreprise. Néanmoins, il n'est pas économiquement envisageable de le refondre complètement. Par conséquent, SOA+d supporte la vision de la réutilisation ; il faut au maximum ré-exploiter l'existant quand c'est possible. La réutilisation des connaissances dans le processus de conception est une stratégie majeure pour diminuer les coûts, réduire les délais et améliorer la qualité du produit.

En effet, SOA+d propose une nouvelle méthode pour le développement de l'architecture SOA. Elle prend en compte trois points de vue qui doivent être analysés afin de développer l'architecture SOA, Les visions métier et information, notre contribution consiste dans la proposition de la troisième vision qui est la décision.

Comme le montre la figure 3.2, et dans le but de réutiliser les connaissances existantes, SOA+d a été défini pour intégrer les concepts de conception utilisés pour le développement des SOA, en ajoutant des éléments spécifiques aux décisions.

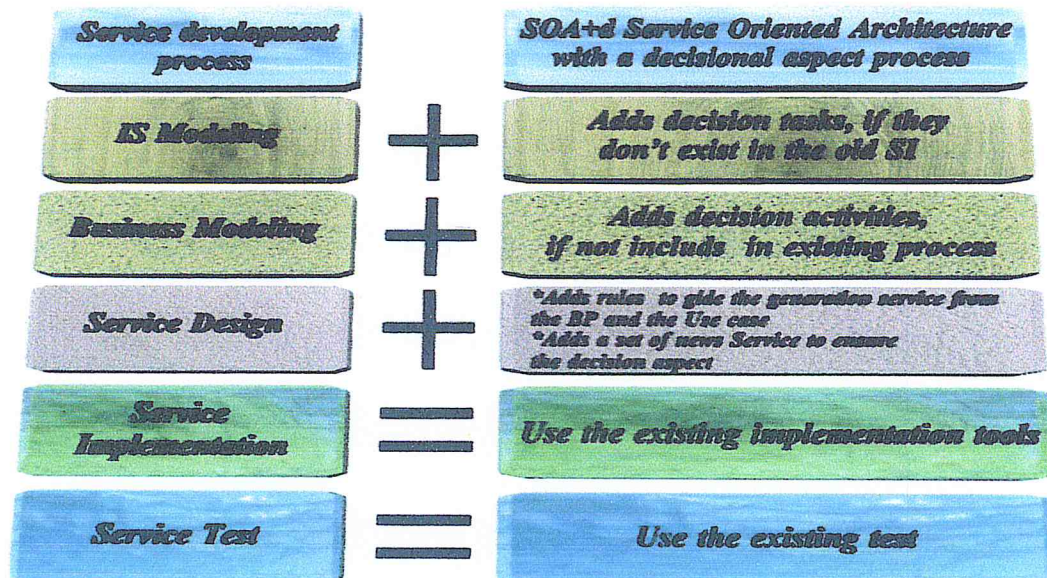


Figure 3-2 Comment SOA + d intègre au processus de développement des services existant

3.4 Description détaillée de la méthode proposée SOA+d.

3.4.1 Phase 1 : La phase d'analyse

La première phase de SOA+d est réalisée conjointement par les intervenants métier, les intervenants d'information et les décideurs de l'entreprise. Elle comporte trois étapes majeures : l'élaboration du modèle métier, l'étude du système d'information et la construction d'une ou des solutions aux problèmes décisionnels. La phase d'analyse est définie en se basant sur un ensemble de modèles permettant de décrire les exigences métiers.

Principalement, cette phase se compose des modèles des cas d'utilisation, BPMN et des notations DMN. Les modèles des cas d'utilisation permettent la formalisation des besoins des utilisateurs associés aux différents acteurs interagissant avec le système. Les modèles BPMN permettent la description des différents scénarios de réalisation des cas d'utilisations. A la fin, le DMN décrit la conception des solutions des décisions.

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

3.4.1.1 Activité 1 : l'analyse du système d'information

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

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

3.4.1.2 Activité 2 : l'analyse métier

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

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

La situation existante avant la prise de l'aspect décision est représentée avec les modèles As-Is ; alors que la nouvelle situation est représentée avec les modèles To-Be. Jackson (1995) explique que les modèles As-Is décrivent les propriétés possédées par le système (propriétés indicatives) alors que les modèles To-Be décrivent les propriétés que le

nouveau système possédera (propriétés opératives). Dans le cas où les participants n'auraient jamais traité les problèmes de décision, il faut dresser une cartographie de l'existant (As-Is) et de la cible (to-be). Sinon, si les participants prennent des décisions déjà il faut prendre en considération les modèles existants.

✦ La modélisation des processus métier

Nous adoptons le langage BPMN 2.0 dont la sémantique est proposée par l'omg. Cette activité est l'une des activités clés dans le développement de services Métier (SM), car elle est l'entrée principale nécessaire pour comprendre et décrire le processus métier dans l'organisation. Nous utilisons BPMN pour les spécifier, qui fournit des éléments tels que les couloirs (pool, voie), objets de flux (activités, passerelles), élément de connexion (séquence, message) et des artefacts (groupes, données) pour modéliser le processus métier.

3.4.1.3 Activité 3 : l'analyse de décision

Nous définissons trois étapes dans l'analyse de décision :

3.4.1.3.1 Identifier les Décisions

Soa⁺ propose d'identifier les problèmes de décision, de la manière suivante : à partir de cas d'utilisation et le BPMN nous déterminons les mots ou les indices de décisions tels que : calculer, choisir, prévoir, déterminer, valider, évaluer la sélection et, bien sûr, décider. Cette tâche ou sous-processus sont considérés comme un service de décisions candidats.

3.4.1.3.2 Spécifier les besoins de la décision

Aller au-delà des simples descriptions des décisions pour spécifier les exigences détaillées de la décision. Le but de cette étape est la spécification des informations et des connaissances

Nécessaires pour prendre des décisions et de les combiner dans un graph des besoins de la décision. Le diagramme des besoins de la décision de DMN est une notation représentant des éléments importants de la prise de décision et leurs dépendances (OMG, 2014). Les éléments importants sont les décisions, connaissance métier, source de connaissance métier, et les données d'entrée

3.4.1.3.3 Spécifier le niveau logique de la décision

Le Niveau Logique de Décision de DMN (NLD) est l'étape pour spécifier la logique ou le raisonnement utilisé pour trouver une solution au problème de décision. Chaque décision dans

un diagramme des besoins de décision de DMN peut avoir une expression de valeur, des règles métier, modèle de connaissance. Un modèle de connaissance métier peut être les spécifications de tables de décision, des modèles analytiques ou un algorithme.

3.4.2 Phase2 : Identifier et classer les services

Cette phase est basée sur les cartographies déjà publiée par (Boumahdi & Chalal, soada: Service Oriented Architecture with a Decision Aspect, 2013) pour identifier les services (métier, information et décision). Cette activité vise à identifier les services nécessaires à l'exécution du processus métier en cours de développement et c'est la première clé dans notre méthode. Les principaux entrées de cette phase est le BPMN, le cas de l'utilisation et de le DMN ont été spécifiés précédemment.

La découverte de services vise à apporter un soutien dans l'identification des services qui seront utilisés par les processus métier. Afin d'identifier et de décrire les services à partir des informations fournis par les cas d'utilisation, BPMN et la DMN, nous proposons des règles de passage comme un guide pour identifier les services. Les règles proposées par (Boumahdi, et al., 2014) sont:

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

| Cas d'utilisation | Service | Description | Exemple |
|--------------------------|---------|---|---------|
| Cas d'utilisation global | Métier | Chaque cas d'utilisation global (CU) correspond à la fonction métier. | |

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

| Cas d'utilisation | Service | Description | Exemple |
|------------------------|---------|---|---|
| Cas d'utilisation CRUD | CRUD | Dans le cas d'utilisation, si l'association Extend a ces objectifs : Créer, Lire, Mise à jour et Supprimer, le service sera présenté comme un service CRUD. | <pre> usecaseDiagram usecase CV_CRUD as CV CRUD usecase Update_CV as Update CV usecase Submit_CV as Submit CV usecase Delete_CV as Delete CV CV_CRUD ..> Update_CV : <<extend>> CV_CRUD ..> Submit_CV : <<extend>> CV_CRUD ..> Delete_CV : <<extend>> </pre> |

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

| BPMN | Service | Description | Exemple |
|----------|-------------|---|--|
| Activité | Fonctionnel | Chaque activité identifiée dans le processus métier BPMN peut donner lieu à un service fonctionnel. | <pre> graph LR A[Pre-select candidates for interview] --> B[Interview pre-selected candidates] B --- C[Functional service] </pre> |

Règle de Mapping 4: Tâche de BPMN vers service Technique

| BPMN | Service | Description | Exemple |
|-------|-----------|---|---|
| Tâche | Technique | Chaque tâche identifiée dans le modèle de processus métier peut donner lieu à un service technique. | <pre> graph LR A[Publish on corporate website] --- B[Technical service] C[Distribute on job boards] --- B D[Advertise on social media] --- B </pre> |

Règle de Mapping 5 : Indice de Décision vers Service Décision

| Cas d'utilisation+ BPMN | Service | Description | Exemple |
|-------------------------|----------|--|---------|
| Indice de Décision | Décision | A partir du cas de d'utilisation le BPMN, déterminer l'indice des décisions (mot décision). Cette tâche ou sous-processus sont considérés comme un service de décisions. | |

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

| DMN | Service | Description | Exemple |
|-----------|----------|---|---------|
| Décisions | Décision | Chaque décision du diagramme des exigences de décision correspond au service de décision. | |

Règle de Mapping 7: Connaissance Métier DMN vers service de conception

| DMN | Service | Description | Exemple |
|---------------------|------------|--|---------|
| Connaissance métier | Conception | Chaque connaissance métier à partir du diagramme des exigences de décision correspond au service de la conception. | |

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

| DMN | Service | Description | Exemple |
|-------------------------|---------|--|---------|
| <i>Knowledge Source</i> | Choice | Chaque source de connaissances à partir du diagramme de décision des exigences correspond au service de choix. | |

Règle de Mapping 9: Données en entrée du DMN vers Service Intelligent

| DMN | Service | Description | Exemple |
|--------------------------|-------------|---|---------|
| <i>Données en entrée</i> | Intelligent | Chaque données en entrée à partir du diagramme des exigences de décision correspond au service intelligent. | |

3.4.3 Phase3 : Modélisation de Services

Cette phase se concentre sur la spécification des services (modèle de données, interfaces, etc.) Dans cette phase, les services doivent être modélisés avec formalisme, nous adoptons une spécification à la base du langage soaml (OMG, 2012) qui offre un haut niveau d'abstraction, il est alors nécessaire d'affiner les services pour les rendre propres à une plateforme donnée. SOA+d propose le développement des diagrammes suivants :

✓ *Diagramme de Participants*

Le diagramme de participants sert à définir les fournisseurs et les consommateurs dans un système. Un participant peut jouer le rôle de fournisseur de services, consommateurs, ou les deux au même temps. Comment identifier les participants ?

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

✓ *Diagramme d'Architectures de services*

Est une description de la manière dont les participants (acteurs) s'échanges des services exprimés par les contrats de services. Nous définissons les les étapes de la modélisation des architecture des services comme suit :

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

✓ *Contrats de Service*

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

3.4.4 Phase4 : La réalisation

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

La figure 3.3 illustre les différents choix technique qui doivent prendre en considération pour implémenter une architecture SOA. Dans les sections ci-dessous, les éléments d'architecture sont mappés au Système Open Source (SOS) outils/produits proposés.

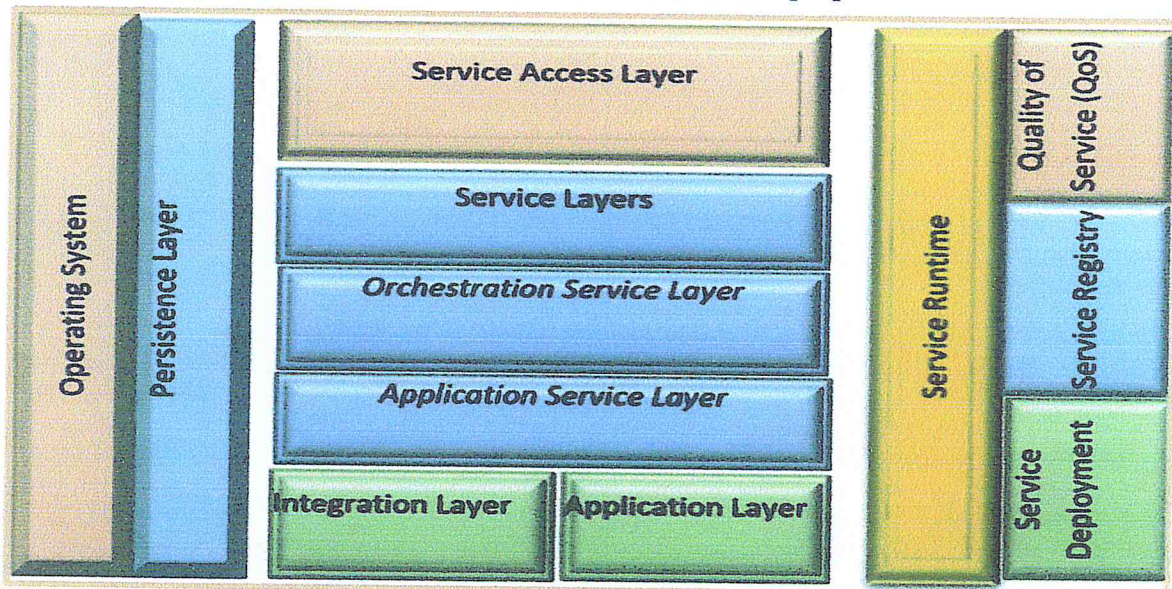


Figure 3-3 Les différents choix technique proposés par SOA^{td}.

3.4.4.1 Les couches techniques

- Couche Accès Service :** est la couche interface entre le client final et le système. Nous pouvons utiliser le Serveur Apache HTTP pour implémenter cette couche.
- Couche Service :** cette couche définit et expose les fonctionnalités métier et, application prises en charge par le système en tant que services. Sur la base de l'abstraction et de l'objectif, cette couche peut être encore divisée en :
- Couche service orchestration :** Cette couche est fondamentale dans la réalisation de certains des principaux avantages de l'architecture SOA – la composition,

la réutilisation et l'abstraction. Les services de cette couche définissent et exposent les processus métier du système. Comme outil nous pouvons utiliser :

- ✦ Jbpm de jboss permet la création de processus métier qui coordonnent entre les personnes, les applications et les services.
- ✦ Le Moteur activebpel est capable d'exécuter des définitions de processus créés pour le Langage Business Process Execution Language.
- **Couche Service Application** : Les services de cette couche sont les «travailleurs». Ils représentent l'application logique réelle nécessaire pour mettre en œuvre les processus métier. Nous pouvons implémenter cette couche en utilisant Apache Axis, qui est une implémentation de SOAP et permet le développement et le déploiement de services (par le biais d'un serveur d'application J2EE)
- **Couche Intégration** : Cette couche intègre le système orienté services pour les systèmes existants. Service Mix est un exemple d'outils de mise en œuvre de cette couche.
- **Couche Application** : Cette couche accueille les implémentations des services.
- **Couche Persistance** : Il s'agit généralement de la couche de base de données, où les données fonctionnelles et non fonctionnelles sont conservées temporairement (cache) ou d'une manière permanente (durée déterminée par les exigences métier et Techniques). Pour implémenter cette couche, nous pouvons utiliser: Apache Derby, postgresql, mysql, ou Hibernate.
- **Système d'exploitation** : c'est le système d'exploitation sur lequel les différentes piles du système orienté services sont hébergés. Comme logiciel Open Source, nous pouvons utiliser Linux.

- ◆ **Service Exécution** : cette pile représente les composants nécessaires pour déployer et «exécuter» les services. Les éléments de base de cette pile sont :
 - ✦ **Service Registry** : Service Regitre : Le registre est le référentiel central où tous les services sont définis, exposés en utilisant les spécifications standards, et découverts
 - ✦ par les deux systèmes internes et externes. Comme outil nous pouvons utiliser , qui est une implémentation Java de la spécification UDDI pour les services.
 - ✦ **Qualité de Service** : cette composante représente la fourniture de capacités de transactions sécurisées et fiables à des niveaux de performance optimaux.
 - ✦ **Déploiement de services** : Il s'agit de la plate-forme de déploiement de services. Cette plate-forme doit supporter les fonctionnalités d'une architecture orientée services. Nous pouvons utiliser les outils suivants:
 - **Logicblaze FUSE** : est une plate-forme SOA complète. Les composants comprennent une plate-forme de messagerie, bus de services d'entreprise, et les fournisseurs de services d'exécution nécessaires pour le routage, l'orchestration, les transformations, la sécurité et le service des recherches.
 - **Java Open sourceapplication Server (jonas)** : est un serveur d'applications J2EE avec un support pour les services Web par Apache Axis.
 - **Jboss Application Server** est un serveur d'applications J2EE capable d'accueillir des services Web.

Conclusion

Nous avons présenté dans ce chapitre notre approche SOA^{+d}, une démarche top down, qui répond au manque de méthodes pour la construction de la SOA avec un aspect décisionnel. La démarche SOA^{+d} a été décrite grâce à un processus de quatre phases qui guide les analystes dans le processus d'identification et de modélisation des services.

Partie 2 Etude de cas

➔ **Chapitre IV : Conception et réalisation.**

Chapitre 4 : Conception et réalisation

4.1 Introduction

Ce chapitre décrit une étude de cas qui permet d'appréhender, par la pratique, les différentes phases de la méthode SOA^{td}. La première partie de cette section est consacrée à la présentation du contexte dans lequel nous avons appliqué notre méthode. Quant à la deuxième, elle mettra en évidence l'application de la méthode SOA^{td} au système d'information de l'étude de cas. Ce qui permet en retour de tester l'applicabilité et la pertinence de la démarche scientifique dans le domaine gestion de stock.

4.2 Étude de cas

4.2.1 Phase 1 : Analyse

Cette étude a été réalisée pour l'hôpital Frantz Fanon, situé dans la région du Nord de l'Algérie. Étant donné que les dépenses de médicaments sont le principale composante des dépenses de l'hôpital, contrôle des stocks pharmacie est un domaine intéressant à étudier. Quelque raison du choix de la gestion du stock de la pharmacie :

- ✦ Le PFE est cruciale pour le succès de la plupart des hôpitaux,
- ✦ Le rôle majeur de la décision dans les hôpitaux,
- ✦ La PFE comprend des activités commerciales, qui sont liées à des activités de la décision.

La méthode proposée suit une approche vers le bas (approche descendante) pour découvrir les services intervenant dans SOA + d. Elle se compose de six étapes.

4.2.1.1 Analyse du système d'information

Cette phase consiste à élaboration le modèle métier, l'étude du système d'information et de trouver des solutions aux problèmes décisionnels, ainsi notre processus de gestion de stock de pharmacie est composé de 4 étapes figure II-2 comme suit :

A. Approvisionnement : permet de répondre aux questions : Quoi acheter ? Et, En quelle quantité ? Cette étape inclus :

- ✦ **Estimation des besoins** : Les différents services de l'hôpital établissent des demandes en fournitures et matériels indispensables à leur bon fonctionnement, en fonction des besoins établis au préalable. Compte tenu des objectifs te du

volume des activités, ces besoins sont évalués en tenant compte des consommations, et des actions envisagées.

- ✦ **La demande d'approvisionnement** : Après une évaluation exhaustive des besoins du service, le responsable élabore une expression de besoins qu'il transmet au service chargé de l'approvisionnement qui se charge de la vérification de la demande et du déclenchement de la procédure d'achat.

B. L'appel d'offres : dans cette étape, nous répondrons à la question : De qui acheter ? L'objectif de cette étape est d'obtenir le bon produit, la bonne quantité au bon prix au bon moment avec le meilleur fournisseur. Elle comprend :

- ✦ **La recherche et le choix du fournisseur** : A ce niveau, le responsable des approvisionnements initie la procédure de passation du marché suivant le code des marchés publics, en vue de rechercher et sélectionner les fournisseurs à même de satisfaire les services requis. Le choix s'effectue en tenant compte du meilleur rapport qualité, coût et prix.

C. Achat : est le processus de gestion des stocks en vue d'acquérir des médicaments. L'objectif des achats est d'obtenir le bon produit, la bonne quantité au bon prix au bon moment avec le meilleur fournisseur.

D. Stockage : Le magasinier gère une quantité importante de produits notamment des médicaments, des dispositifs médicaux, des consommables, les antiseptiques et les réactifs de laboratoire. Les opérations de stockage sont effectuées par le magasinier sous la supervision du pharmacien responsable du grand magasin.

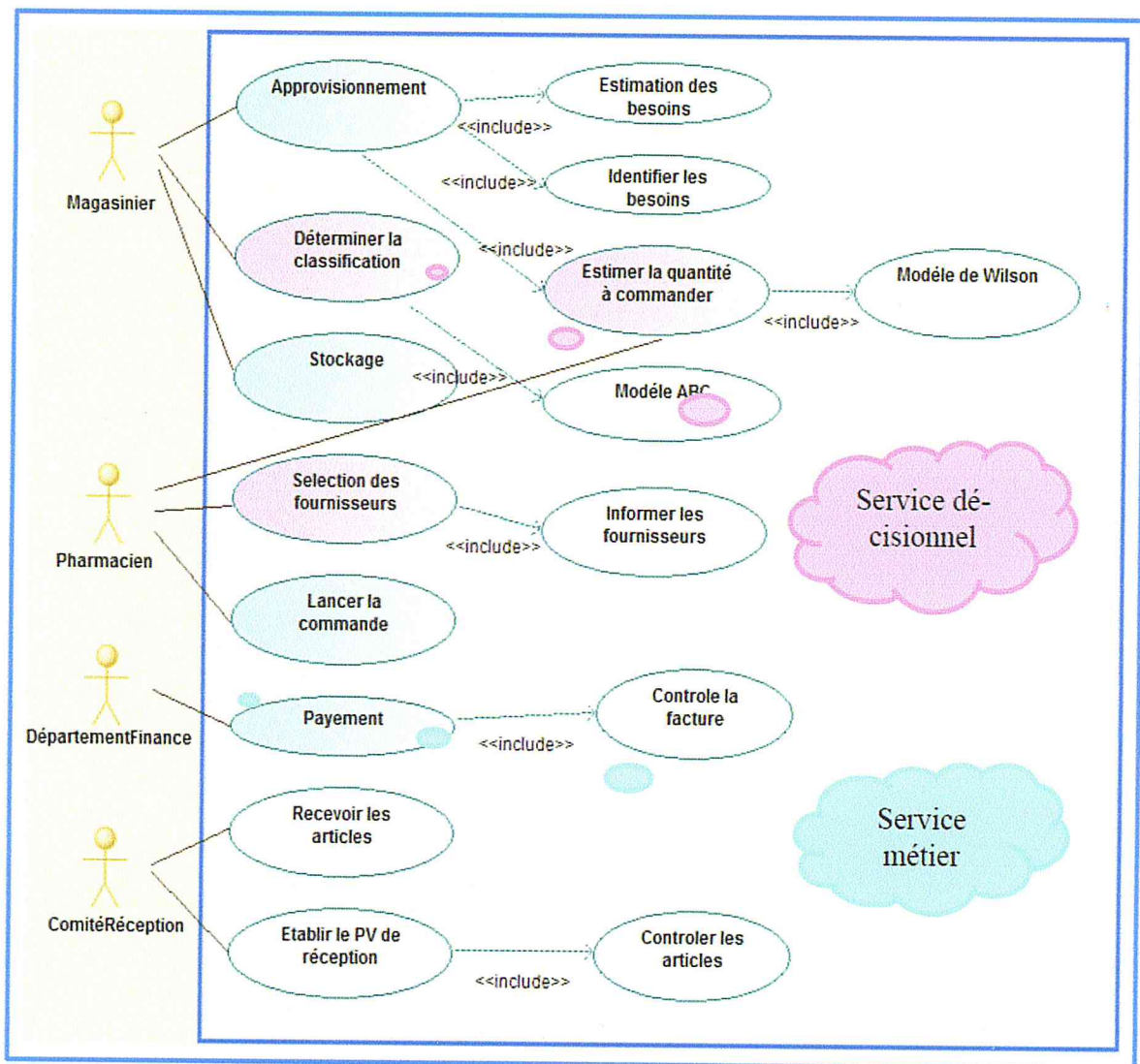


Figure 4:1 Diagramme de cas d'utilisation pour la gestion des stocks pharmaceutique.

4.2.1.2 Analyses métier

BPM est utilisé pour comprendre et décrire les processus opérationnels liés à l'environnement dans lequel sera utilisé le système à construire. Cette phase comprend deux sous-phases :

- ✦ Analyse des processus métiers existants et les objectifs.
- ✦ La modélisation du processus métier en adoptant le langage BPMN.

Cette phase est l'une des activités clés dans le développement des services du processus métier, car c'est l'entrée principale pour comprendre et décrire le processus métier dans l'organisation. Nous utilisons BPMN afin de le préciser les éléments graphique du BPMN qui sont illustrés dans la figure II-3, et la figure II-4 montre notre processus métier de la gestion de stock avec le langage BPMN.

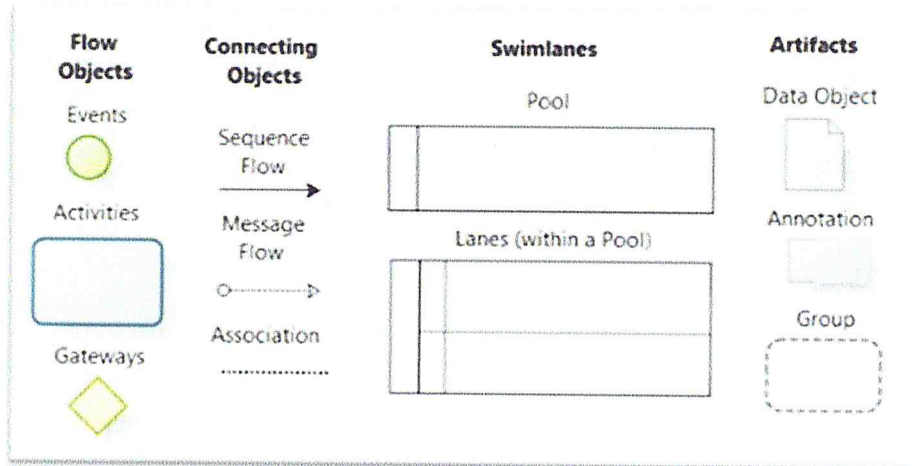


Figure 4:2 Les éléments graphiques de BPMN.

A. Le processus principal de gestion de stock

Le principe

Le processus se déclenche à la vérification de la fiche du stock, le magasinier vérifie la disponibilité des articles, deux cas se présentent :

Si les articles sont disponibles en quantité alors il met à jour l'état des stocks (Enregistrer le mouvement du stock), Sinon dans le cas contraire le magasinier va analyser l'état des stocks via le sous processus « Analyser l'état des stocks » pour voir le classement de chaque article dans la liste des articles en stock, i.e. Est ce qu'il s'agit d'un article important (Classification ABC), ou d'un article réalisant un CA important (Analyse PARETO), et le cas échéant accéder au tableau de bord et vérifier si un tel article n'a pas atteint son Point de commande, dans ce cas le magasinier passe la commande d'approvisionnement au pharmacien.

Ce dernier l'analyse et il estime la quantité à commander via (le sous processus « Estimer les commandes ») et enfin il lance les commandes Pour chaque catégorie d'articles via le sous processus (lancer la commande)

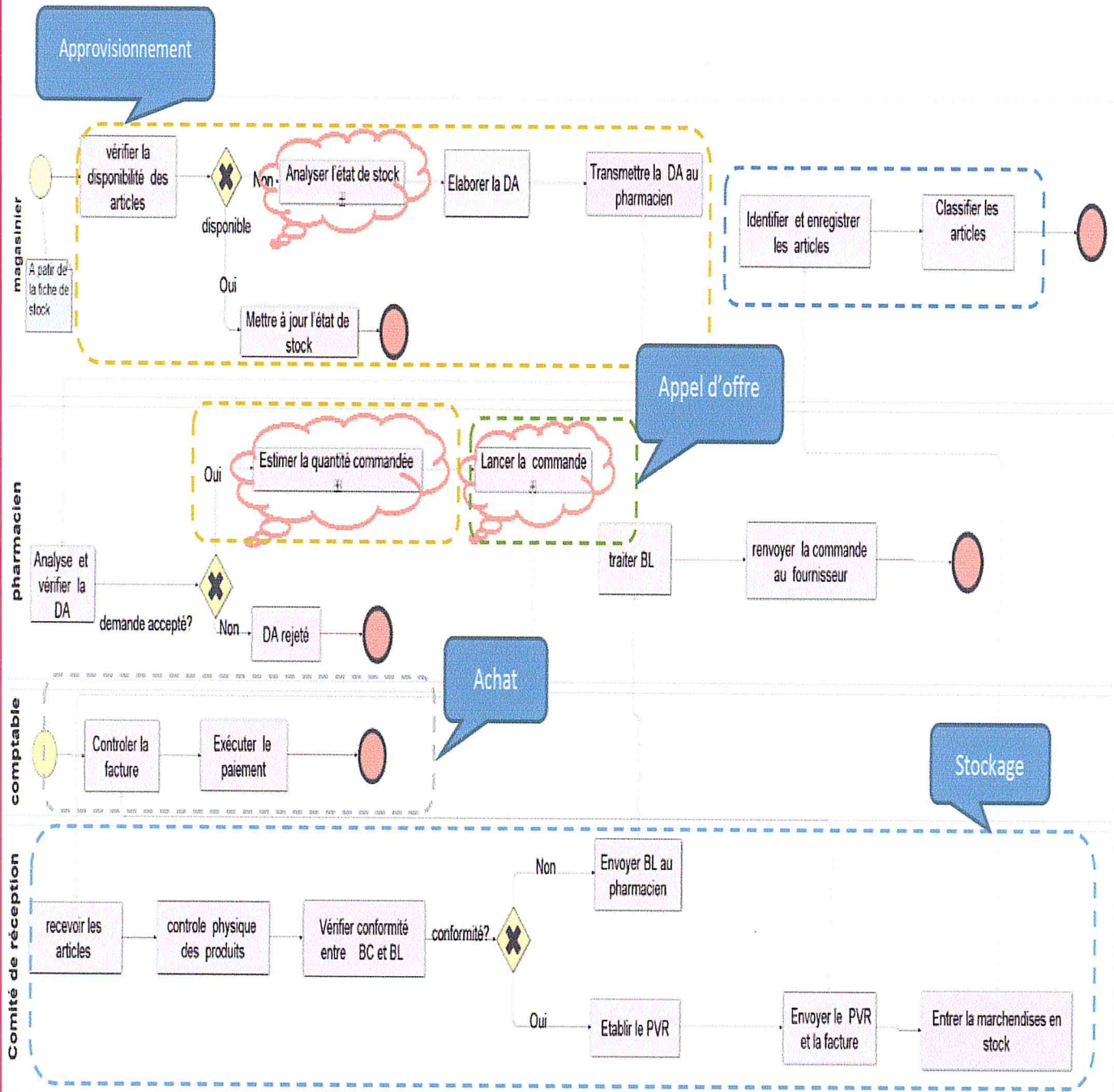


Figure 4:3 Le processus de gestion des stocks de la pharmacie.

B. Le sous processus Analyse de l'état des stocks :

Le Principe : comme cité précédemment, ce sous processus invoqué par le magasinier sert comme son nom l'indique à analyser l'état des stocks. Le processus se déclenche en choisissant un des deux cas :

Voir l'état actuel des stocks : Dans ce cas le magasinier accède au tableau de bord pour vérifier l'état des différents indicateurs, s'il y'a un qui signale il prend la décision qui convient à la situation. Dans notre cas on s'intéresse à l'indicateur « Stock d'alerte », qui est une quantité qui détermine le déclenchement de la commande, en fonction du délai habituel de livraison.

Analyser l'évolution des sorties de stock : Deux cas se présentent selon le rythme des sorties d'un article du stock ; si l'article est classé selon son importance alors il est dans la liste des articles classés en trois catégories (Classe A, B ou C). Si l'article est classé selon le CA qu'il réalise alors il est dans la liste des articles classés en deux catégories (1 : Les articles présents dans le stock en quantité de 20% et réalisant 80% du CA, 2 : Les articles présents dans le stock en quantité de 80% et réalisant 20% du CA).

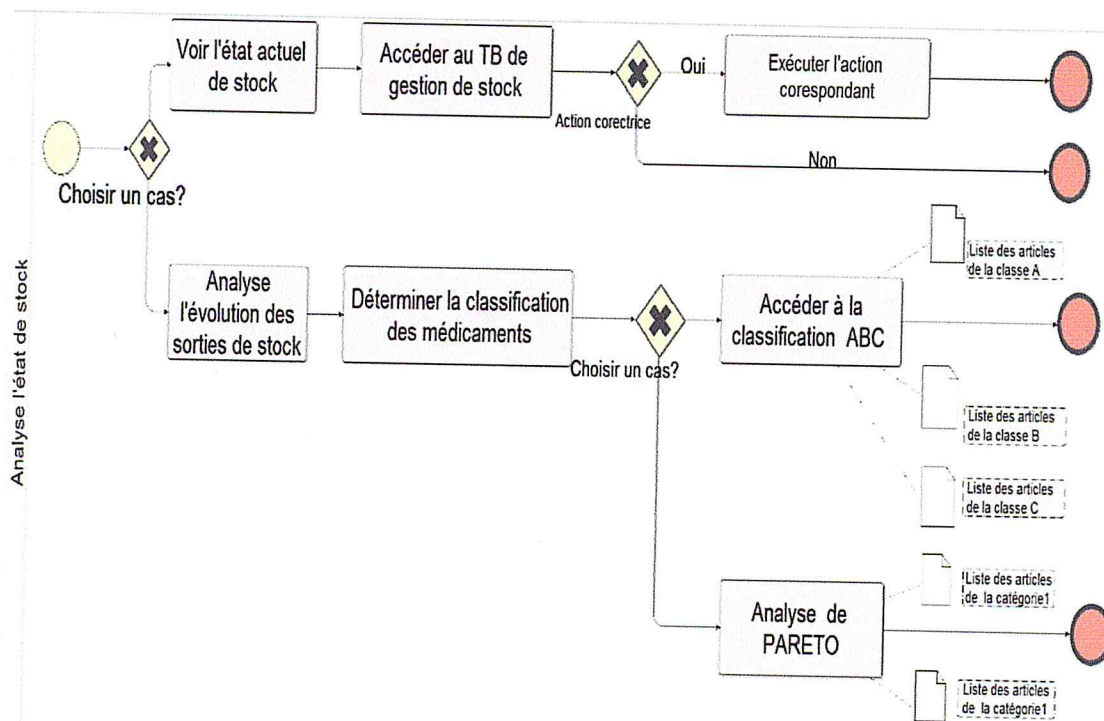


Figure 4:4 Le sous processus d'analyser l'état des stocks.

C. Le sous processus Estimer les commandes :

Le principe : Après établissement de la liste des produits à commander, le pharmacien se charge de l'étape suivante qui est le calcul de la quantité à commander pour chaque article, cette dernière est calculée en fonction de flux de sortie de l'article du stock.

Si le flux de sortie est régulier : Le magasinier applique le modèle de WILSON dont on a désigné auparavant le nombre optimale de commandes à passer dans la période. La quantité à commander dans ce cas dépend de la consommation (besoins) de l'entreprise dans la même période.

Sinon si le flux de sortie de l'article du stock est irrégulier alors la méthode de réapprovisionnement dépend de l'un des deux paramètres : le moment de la commande (la date de la commande) ou bien la quantité à commander.

Si la date de la commande est fixe et la quantité à commander est variable alors le magasinier applique la gestion calendaire sinon si on fixe la quantité à commander dans des périodes différentes alors il applique le système de point de commande

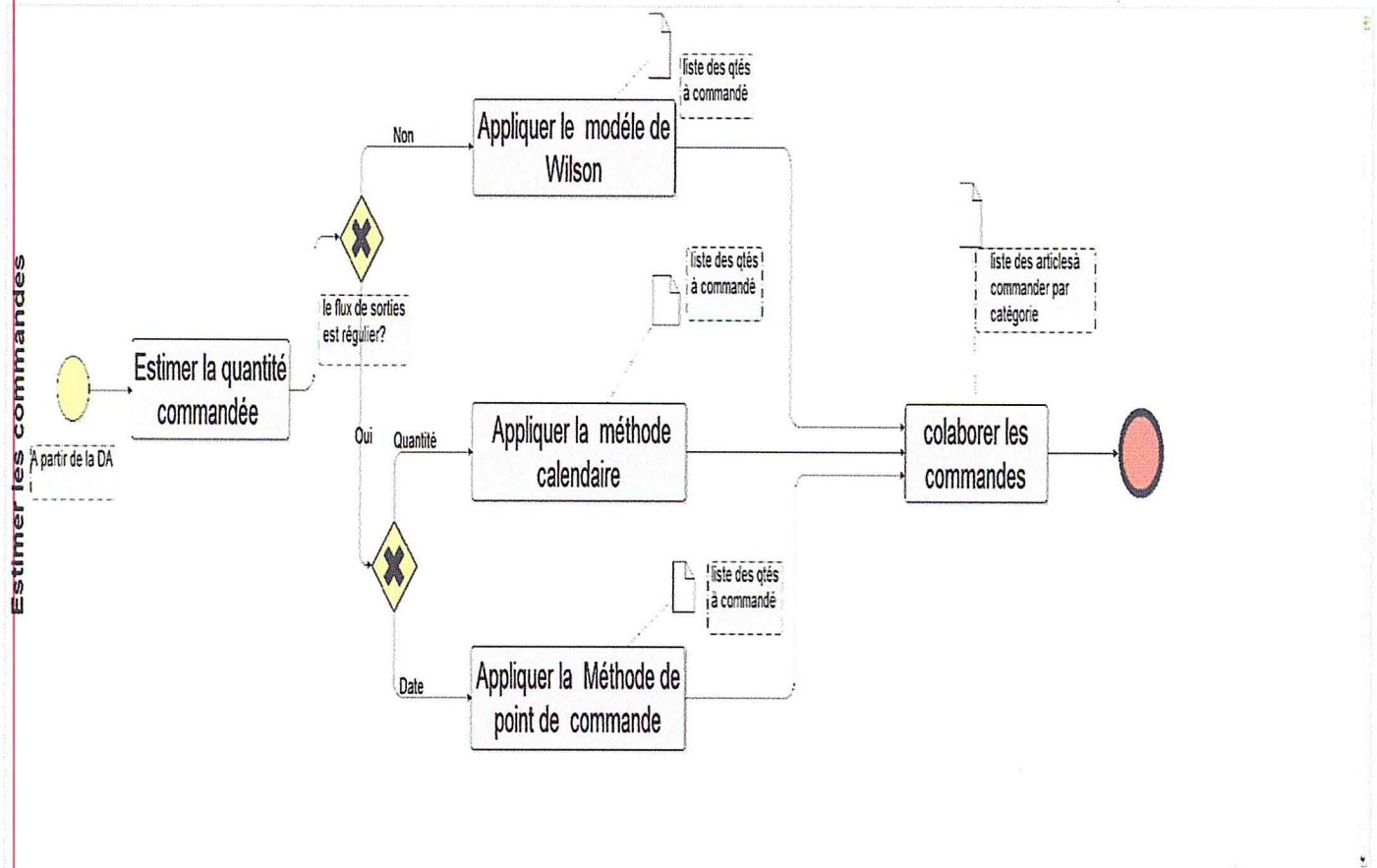


Figure 4:5 Le sous processus Estimer les commandes.

D. Le sous processus Lancer les commandes :

Le principe : Après consolidation des commandes, le pharmacien passe à la dernière étape qui est le lancement des commandes. Le processus débute par la rédaction d'un cahier de charge dans lequel le pharmacien spécifie ses besoins essentiels et nécessaires, une fois terminé il publie l'appel d'offre. A la réception des offres, le magasinier analyse et choisit des offres selon des critères qu'il spécifie puis il lance les commandes.

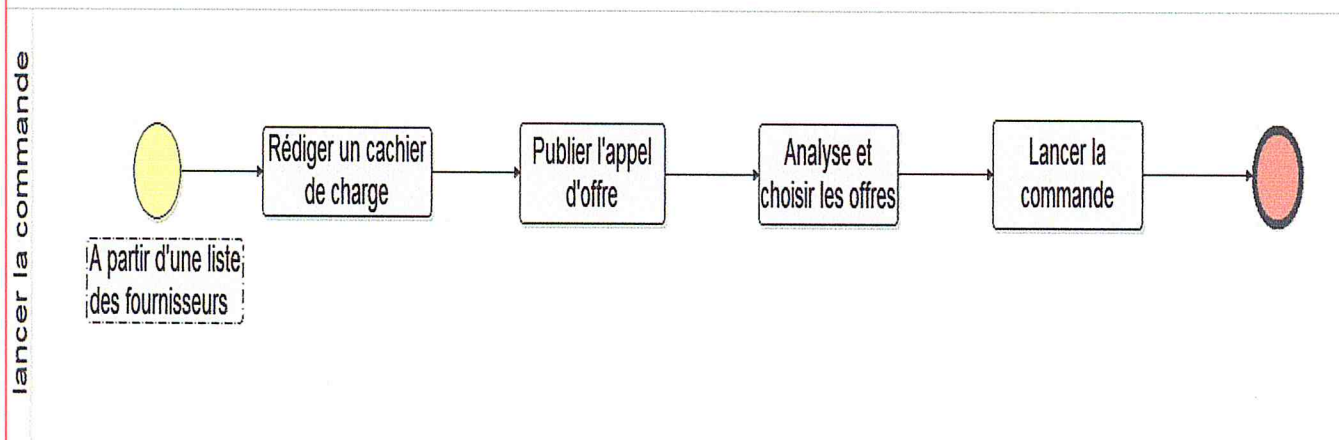


Figure 4:6 Le sous processus Lancer les commandes.

4.2.1.3 Analyse de la décision

Nous définissons trois étapes d'analyse de la décision :

4.2.1.3.1 Identifier les décisions

SOA + d propose d'identifier les décisions, de la manière suivante : à partir de diagramme de cas d'utilisation et le BPMN on détermine les décisions à partir des termes tels que : calculer, choisir, prévision, déterminer, valider, évaluer, sélection et, bien sûr, décider qui nous donne un indice sur l'aspect décisionnel. Cette tâche ou sous-processus sont considérés comme un service de décisions désigné. Ainsi à partir De notre BPMN et le cas d'utilisation de la gestion des stocks, nous avons trouvé les décisions suivant : sélection des fournisseurs, la prévision de la Quantité d'ordre, déterminer la classification des médicaments.

4.2.1.3 .2 Préciser les exigences décisionnel

Aller au-delà des descriptions simples des décisions pour spécifier les besoins détaillée de la décision. Spécifiez les informations et les connaissances nécessaires pour prendre les décisions et les combiner dans un diagramme des besoins de décision DRD. Le diagramme des besoins décision DMN est une notation qui illustre les éléments importants de la prise de décisions et de leurs dépendances. Les éléments importants sont les décisions, connaissance de l'entreprise, source de connaissances d'entreprise et les données d'entrée. La **Figure 4:7**, **Figure 4:8**et **Figure 4:9** illustrer un diagramme d'exigences en matière de décision DMN.

4.2.1.3.3Le niveau logique de décision

Le niveau logique de décision DMN et le milieu ou ce spécifie une expression logique finie, permettant éventuellement l'automatisation. Chaque décision dans le

DMN diagramme des besoins de décision peut avoir une expression de valeur, les affaires ou les modèle de connaissance de ses détails. Un modèle de connaissance de l'entreprise peut être des règles de gestion, table de spécification des besoins de décision, des modèles d'analyse d'un algorithme. La Fig. 4, Fig. 5 et Fig. 6 marquer la limite entre le niveau des besoins de décision et le niveau de logique de décision.

En appliquant les mesures proposées pour résoudre les décisions du la gestion des pharmacies hospitalière, on obtient les résultats suivants :

Décision 1 : sélection du fournisseur :

La sélection du fournisseur est un problème complexe impliquant des critères qualitatifs et quantitatifs des articles. La première étape de toute procédure d'évaluation des fournisseurs est de trouver les critères appropriés à utilisés pour évaluer le fournisseur. Pour se conformer aux critères de sélection des fournisseurs et de leur importance, les données nécessaires ont été recueillies, sur la base des études de (Dickson W. , 1966) Afin de sélectionner le bon fournisseur, l'approche l'analyse hiérarchique du processus (AHP)

La théorie de (Silver, Pyke, & Petersson , 1998) a été adoptée. Le AHP est une théorie de la mesure à travers des comparaisons par paires et s'appuie sur les jugements d'experts pour établir les échelles de priorité.

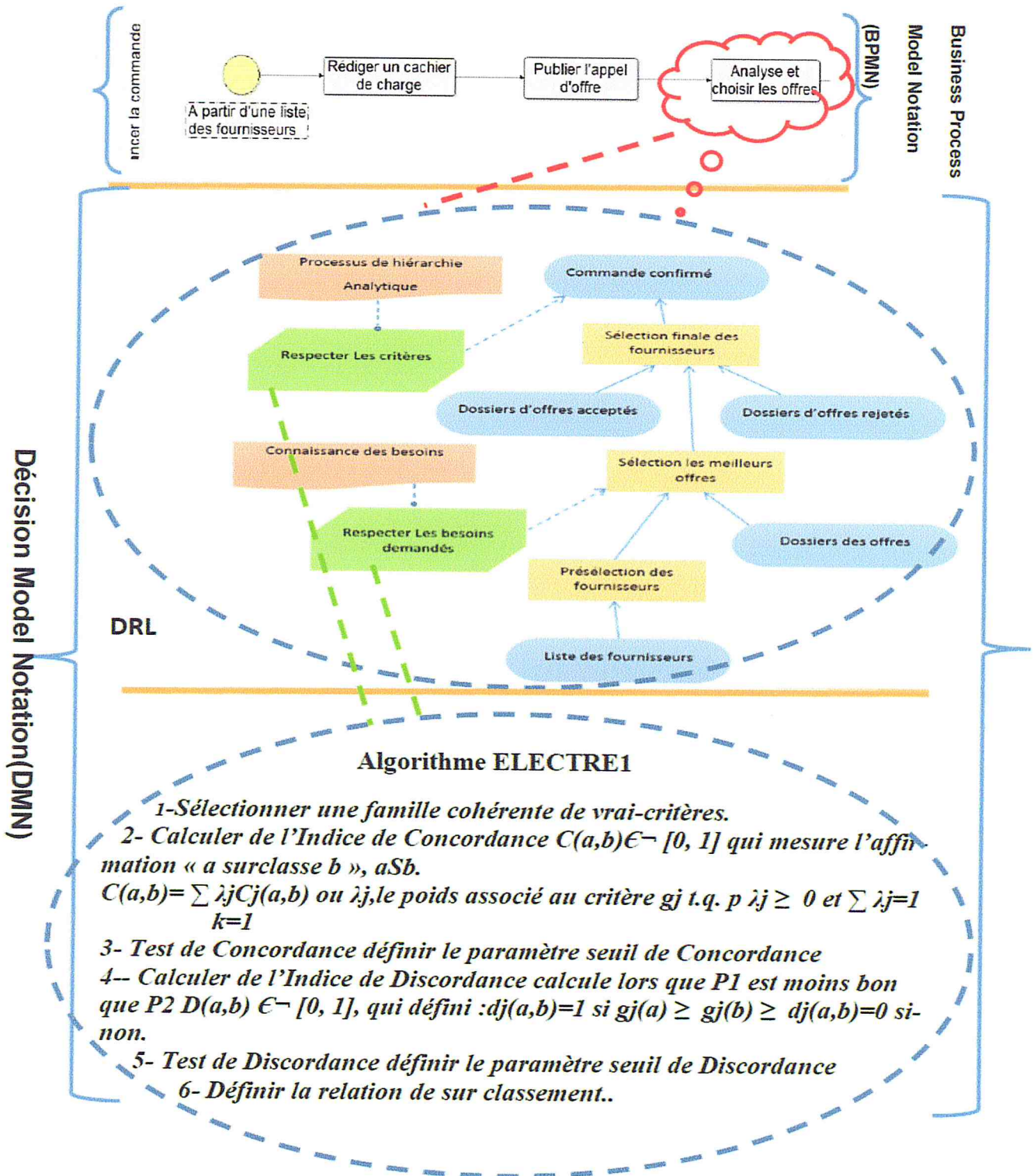


Figure 4:7 choix du fournisseur DMN.

- ✚ **Décision 2 : prévoir la quantité d'ordre :** déterminer la quantité économique (quantité d'ordre économique : combien d'ordre et quand) Prévion est l'activité d'estimer la quantité d'un produit ou un service que les consommateurs achèteront. Nous utilisons le modèle de Wilson pour résoudre ce problème.

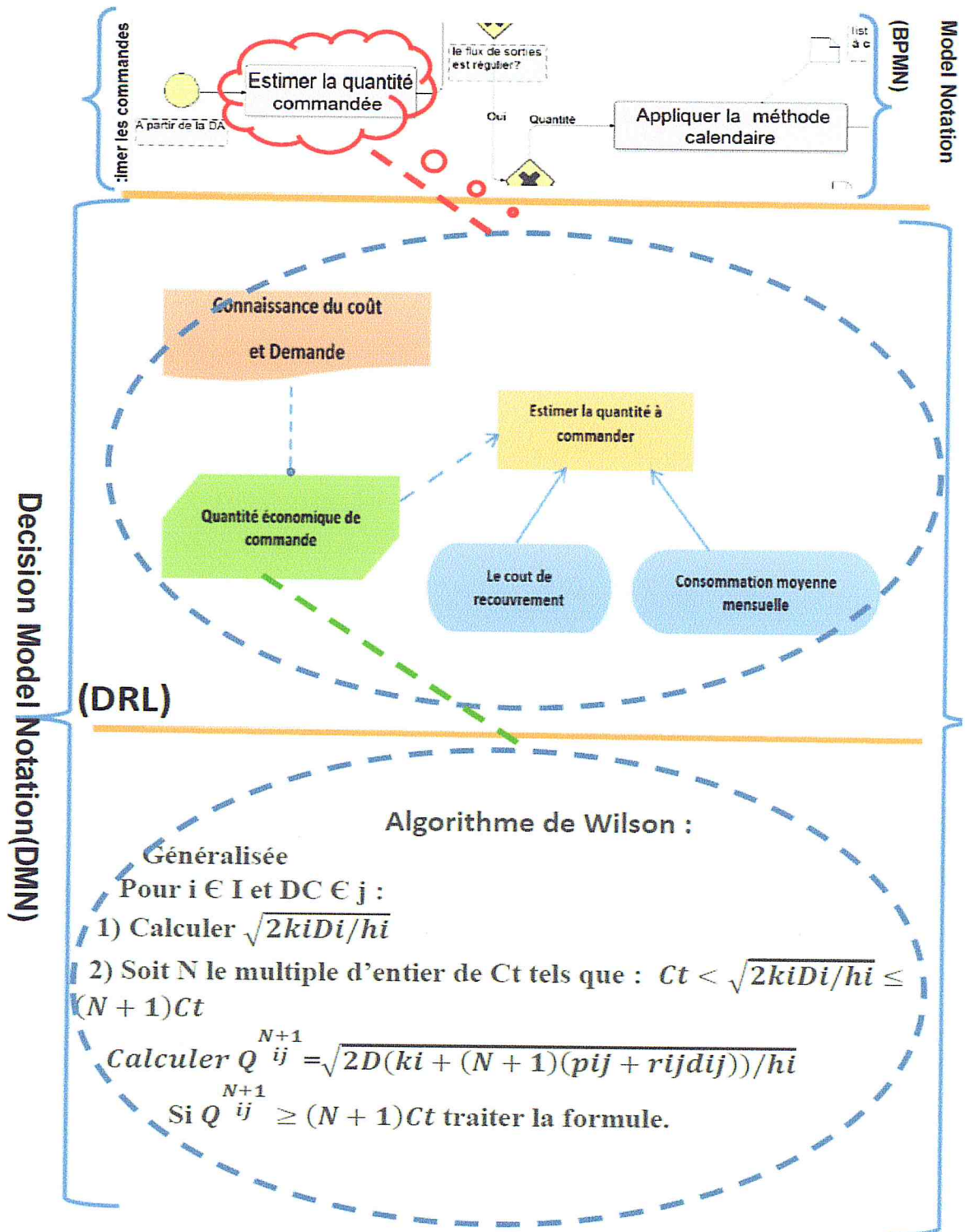


Figure 4:8 Estimer la quantité a commandé DMN.

- ✚ **Décision 3 : Déterminer la classification des médicaments :** Déterminer la classification des médicaments : classification des articles doit s'effectuer lors de la gestion d'une quantité importante de types de produits. Plusieurs modèles de classification peuvent être utilisés pour regrouper les articles. Une des approches les plus courantes et les plus efficaces est le modèle ABC (meilleur contrôle) qui classe les produits fondé sur leur valeur. Quand il s'agit de produits médicamenteux, non seulement que si la valeur qui importe, mais aussi la criticité. Ainsi, l'approche ABC est sélectionnée pour la classification des médicaments.

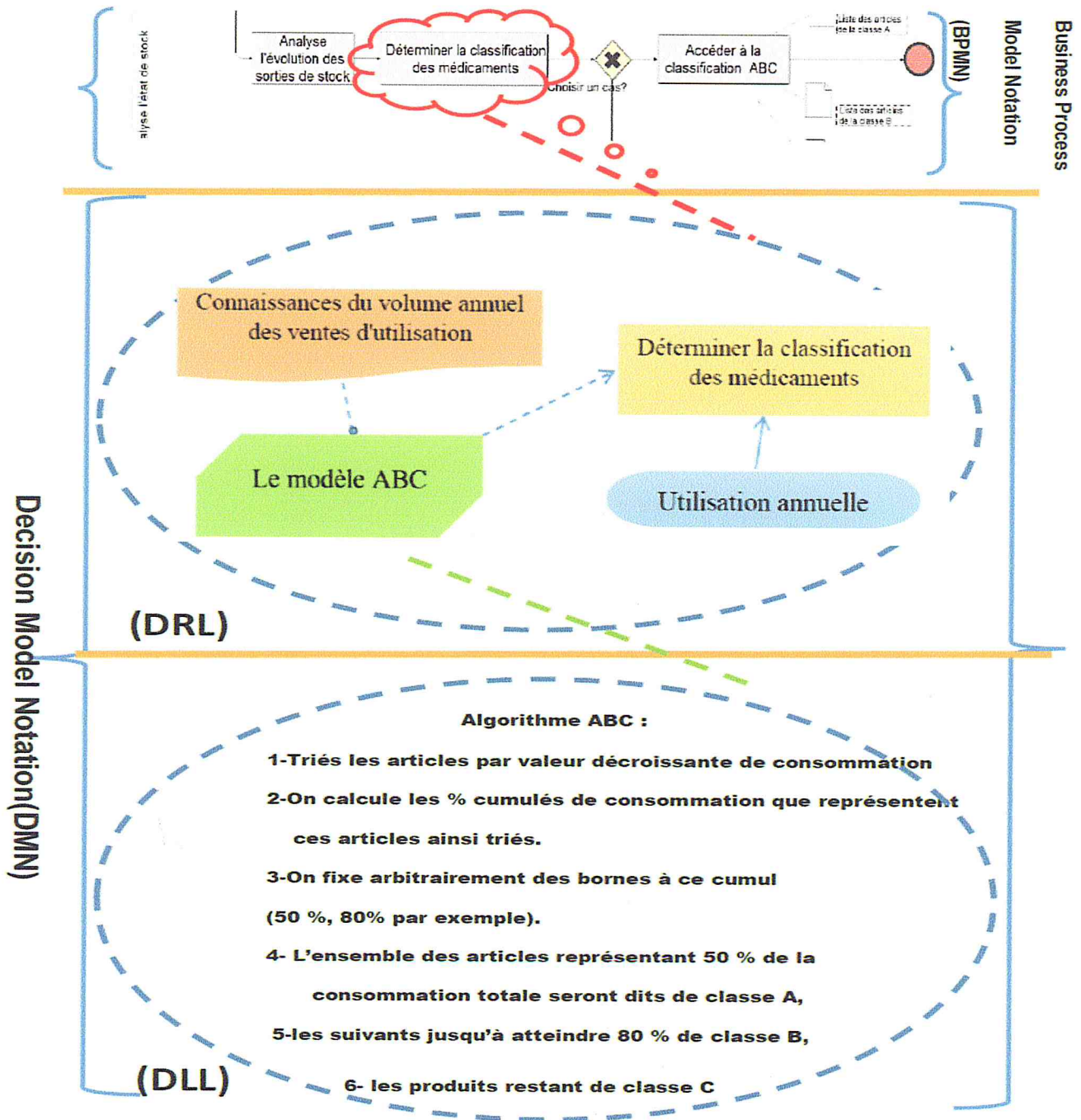


Figure 4:9 classification des médicaments DMN

4.2.2 Phase 2 : Identifier et classer les services

Cette étape est basée sur les cartographies déjà étudiées pour identifier les services (service métier, d'information et de décision). Comme nous l'avons déjà souligné, nous identifions trois types de services : services qui existent au niveau de l'entreprise, système d'information et les services décisionnels. Cette activité vise à identifier les services nécessaires à l'exécution du processus métier en cours de développement et il est une question clé dans notre approche. L'un des principaux apports de cette activité est le BP MN, le cas de l'utilisation et de la DMN précisé précédemment.

La Découverte de service vise à fournir un appui dans l'identification des services devant être utilisés pour les processus métier. Afin d'identifier et décrire les services de l'information qui est fournie en cas d'utilisation, BP MN et le DMN, nous proposons un processus comme guide pour identifier les services comme suit

- ✚ Chaque cas d'utilisation global (UC) correspond au service métier
- ✚ Le diagramme de cas d'utilisation et le BP MN purifient les décisions apparentes (voir mot de la décision). Cette tâche ou sous-processus sont considérés comme service de décisions.
- ✚ Chaque Business Knowledge dans le DRD correspond à la conception du service.
- ✚ Chaque Knowledge Source dans le DRD correspond au service de conception.
- ✚ Chaque Source de connaissance dans le diagramme d'exigences de décision correspond au service intelligent.
- ✚ Dans le cas d'utilisation, si l'objectif de l'association Ex tend et de créer, ajouter et supprimer, alors le service sera présenté comme service CRUD.
- ✚ Chaque activité identifiée dans les processus métier BP MN peut donner lieu à une fonction.
- ✚ Toutes les tâches identifiées dans le BPM peuvent donner lieu à un service intelligent ou technique.
- ✚ Un acteur en BP MN et le cas d'utilisation, correspond au consommateur des services. Après l'utilisation des règles définies dans le processus de gestion des stocks de la pharmacie.

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

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

Règles de passage de BPMN ou SOAML :

Pour passer d'un modèle BPMN à un modèle soaml, il faut respecter un ensemble de règles que Brian Elvæsæter, [Elvæsæter et al. 2010], a défini dans son article intitulé « Aligning Business and IT Models in Service-Oriented Architectures using BP MN and soaml » dont le but est de présenter la nouvelle version du langage de modélisation de l'architecture SOA, soaml.



Règle 0 :

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

Règle 1 : La Tâche à l'action UML :

Une tâche décrit une activité qui être consommé par les acteurs du processus. Elle représente une 'interface abstraite pour le travail accompli sans donner plus de précisions sur le flux de travail exécuté. Le tableau 1 illustre la correspondance des notations.

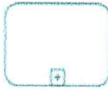

Table 1. Task to UML Action

| | BPMN | SoaML |
|-----------|---|---|
| Construct | Task | Action |
| Notation |  |  |

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

Un sous-processus représente une fonction plus complexe que la tâche. Il est à mentionner que l'architecture des services correspondant au sous-processus n'est pas de faible granularité. Le tableau 2 illustre la correspondance des notations.

Table 2. Sub-Process to Services Architecture

| | BPMN | SoaML |
|-----------|---|--|
| Construct | Sub-process | Services Architecture |
| Notation |  |  |

Règle 3 : Les groupements (Pool) au participant :

Un Pool représente un participant important dans un processus en BPMN . Il peut également être structuré à l'égard des autres participants du processus, créant ainsi un des participants hiérarchiques. On représente le Pool dans le modèle soaml par un rôle dans une architecture de services qui dispose d'un participant de type Pool. Le tableau 3 illustre la correspondance des notations.

Table 3. Pool to Participant (Community-level)



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

Règle 4 : La Ligne (Lane) au participant :

Une ligne représente un participant ou un département en BPMN et est situé dans un Pool, montrant ainsi la hiérarchie à deux niveaux. Afin de montrer la possibilité pour une nouvelle subdivision, la ligne est représentée par un rôle dans un participant au niveau de l'architecture des services de type ligne. L'architecture des participants du type ligne doit adhérer à l'architecture des services

Dont les participants du type Pool (les lignes sont contenues dans ces Pool). Le tableau 4 illustre la correspondance des notations.

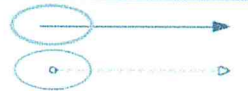
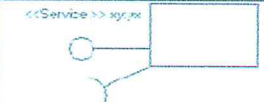
Table 4. Lane to Participant

| | BPMN | SoaML |
|------------------|---|--|
| Construct | Lane | Participant |
| Notation |  |  |

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

Le point de départ de chaque message dans BP MN, il représente un canal de transmission de données entre deux participants ou deux Pools. Dans le soaml ça représente un port d'un service. Le tableau 5 illustre la correspondance des notations.


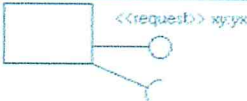
Table 5. Message "Begin" to Service

| | BPMN | SoaML |
|------------------|---|--|
| Construct | Message "Begin" | Service |
| Notation |  |  |

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

Le point de fin de chaque message dans BPMN, il représente un canal de transmission de données entre deux participants ou deux Pools. Dans le soaml ça représente un port de demande d'un service. Le tableau 6 illustre la correspondance des notations.

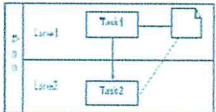
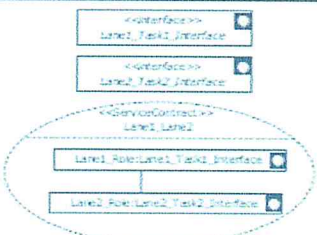
Table 6. Message "End" to Request

| | BPMN | SoaML |
|-----------|---|--|
| Construct | Message "End" | Request |
| Notation |  |  |

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

Il n'Ya pas de construction unique en BPMN qui ressemble à un contrat de service. Vous avez besoin d'analyser les processus BPMN et d'identifier les fragments de processus qui peuvent être mappées à des contrats de service. Un contrat de service définit un cahier des charges de service qui définit le rôle joué par chaque participant dans le service, et les interfaces qu'ils mettent en œuvre pour compléter ce rôle. On peut cependant, définir un modèle BPMN qui peut être associé à un contrat de service. Le tableau 7 illustre la correspondance des notations.

Table 7. Process fragment (pattern) to Service Contract

| | BPMN | SoaML |
|-----------|---|--|
| Construct | Lane1 → Lane2 | Service Contract |
| Notation |  |  |

5.5.2 Application des règles de correspondance sur le processus de gestion des stocks :

On représente les interfaces par des classes qui ont des méthodes.

Enfin on représente le début de flux de message et la fin de flux de message par des ports, le premier (le début de flux qui va du fournisseur au consommateur) est représenté par un "Service Point" et le deuxième (la fin du flux qui va du consommateur au fournisseur) est représenté par un "requestpoint". Les schémas ci-dessous représentent l'architecture des services ainsi que les contrats des services :

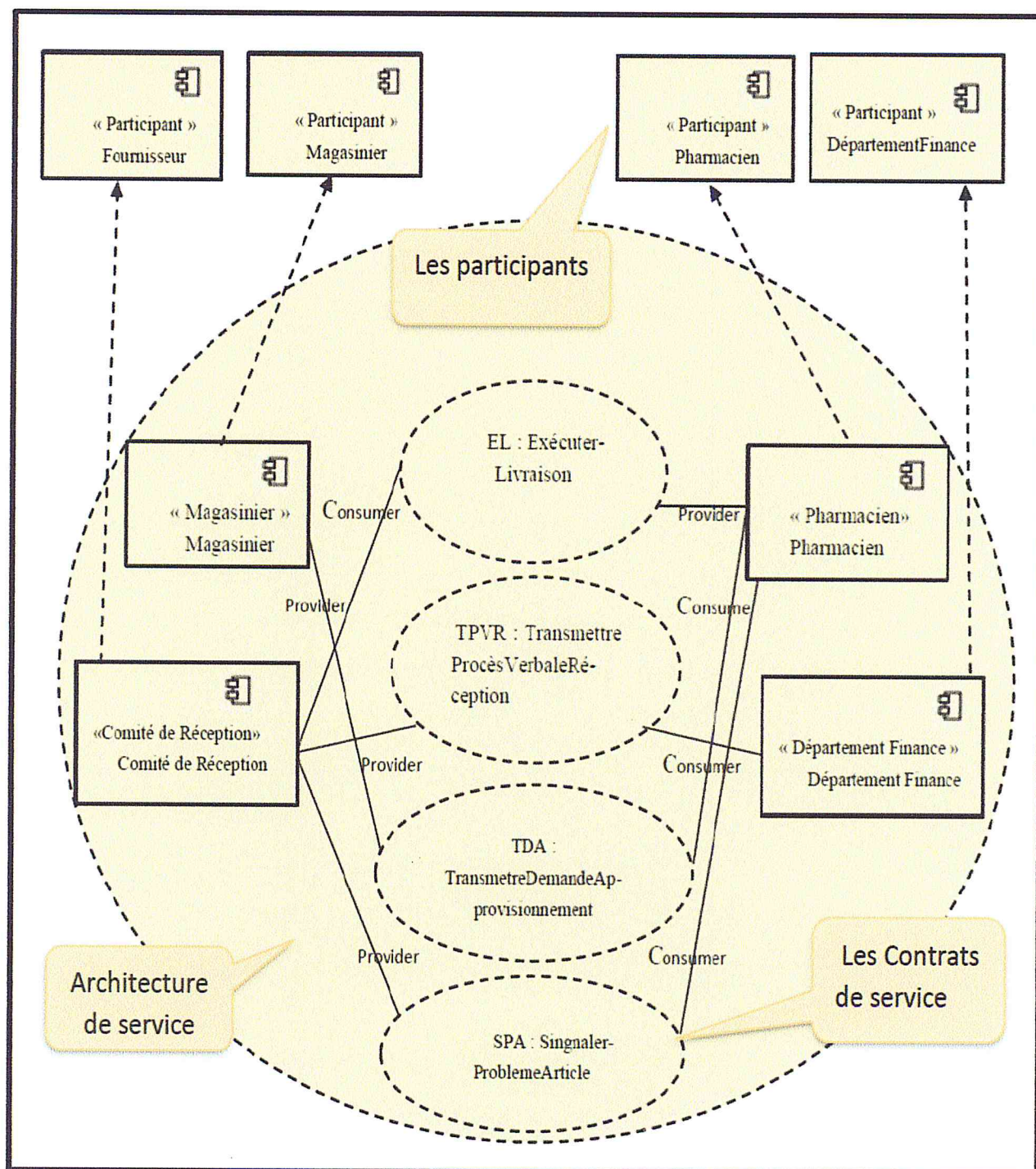


Figure 4:11 Architecture générale.

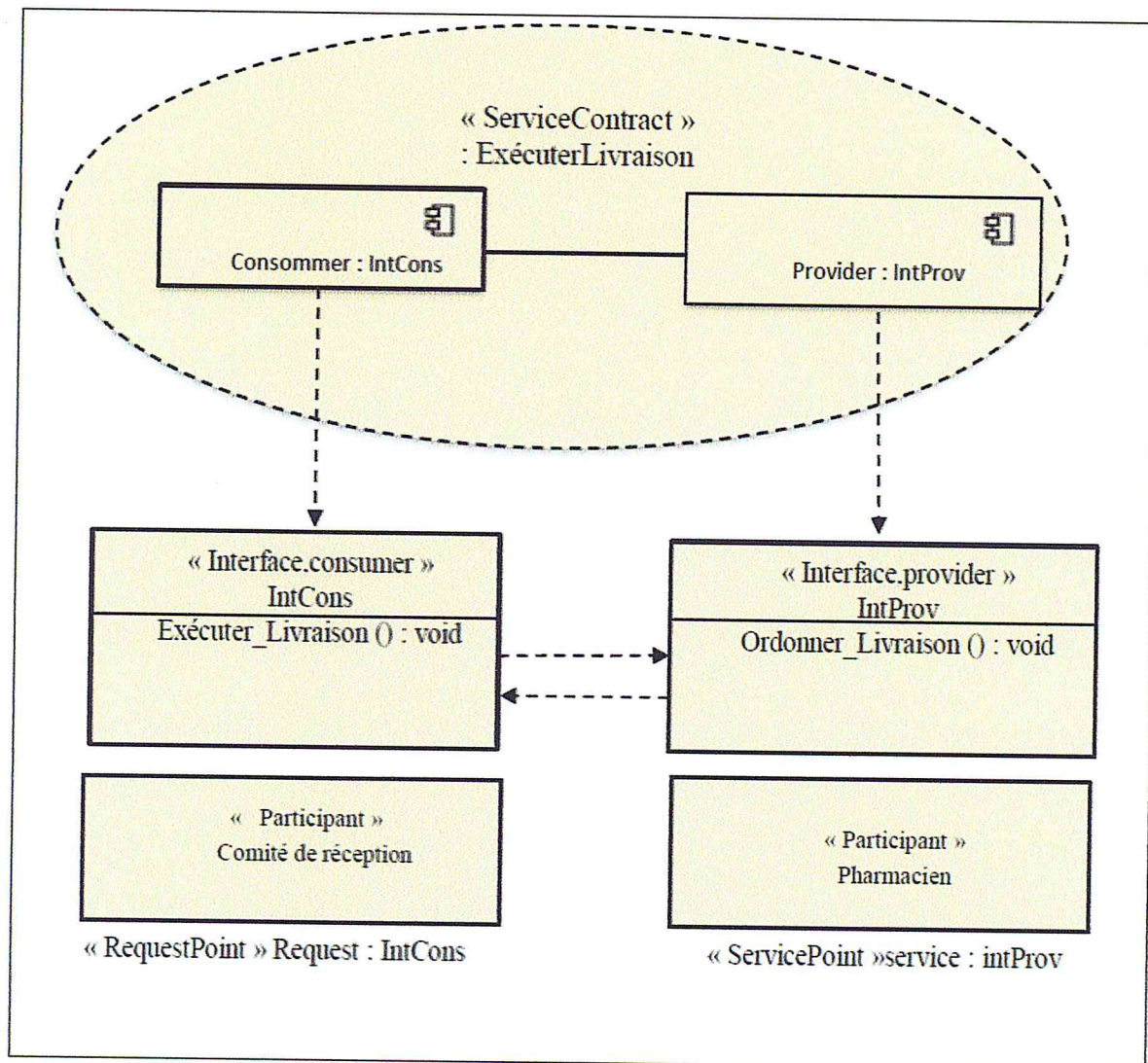


Figure 4:12 Le contrat de service « Exécuter Livraison ».

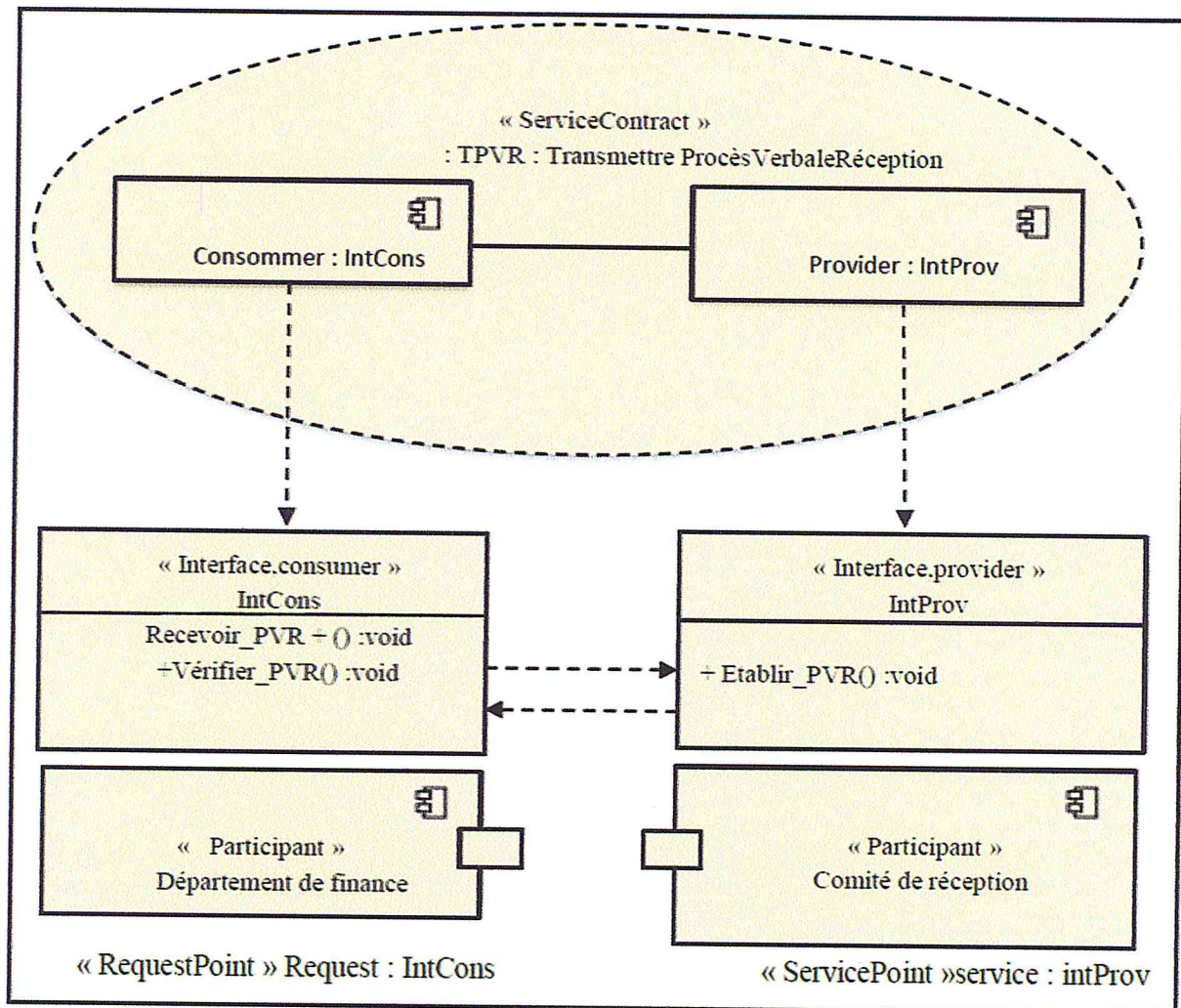


Figure 4:13 Le contrat de service « Transmettre Procès Verbale Réception ».

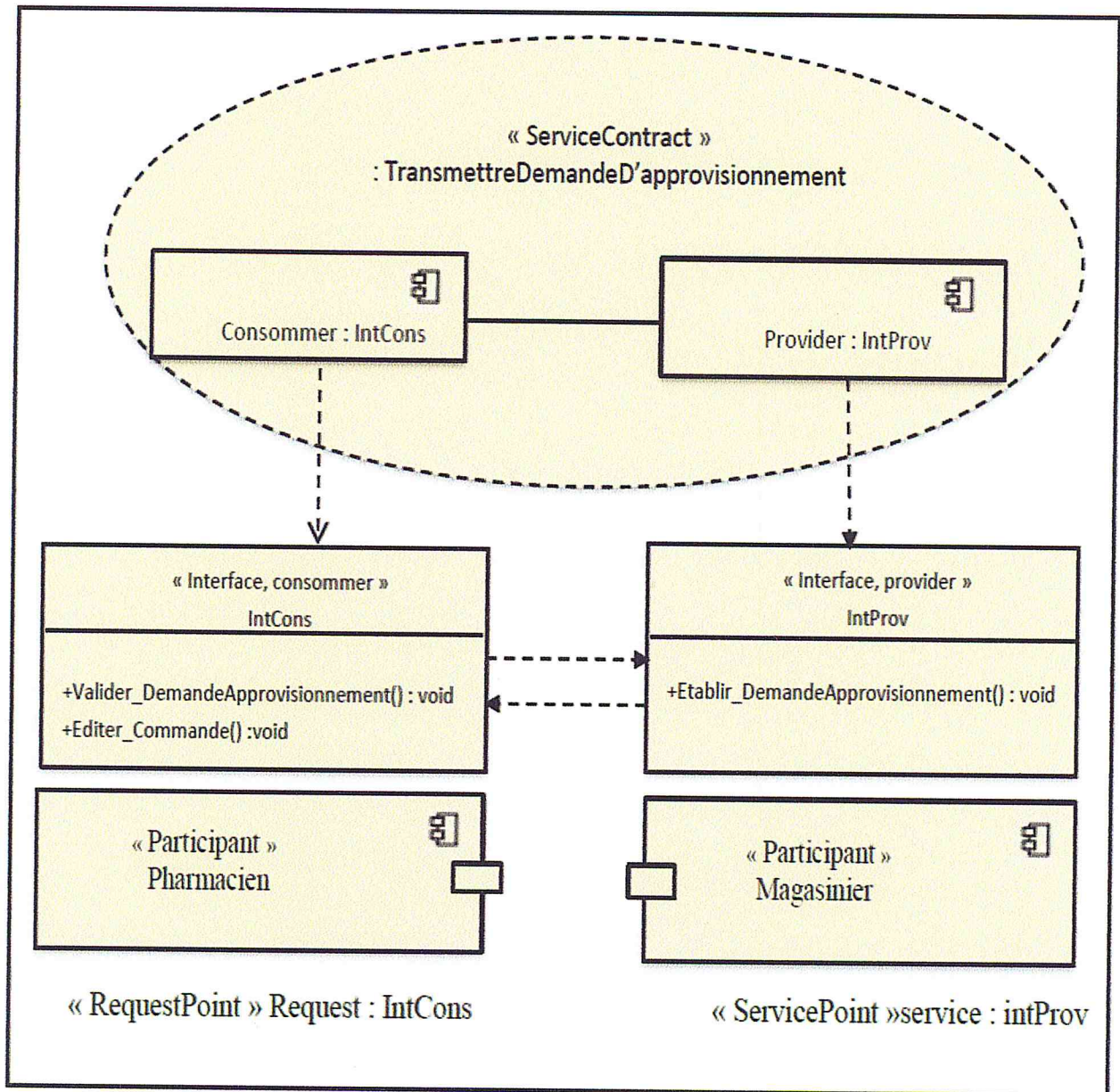


Figure 4:14 Le contrat de service «Transmettre Demande d'approvisionnement».

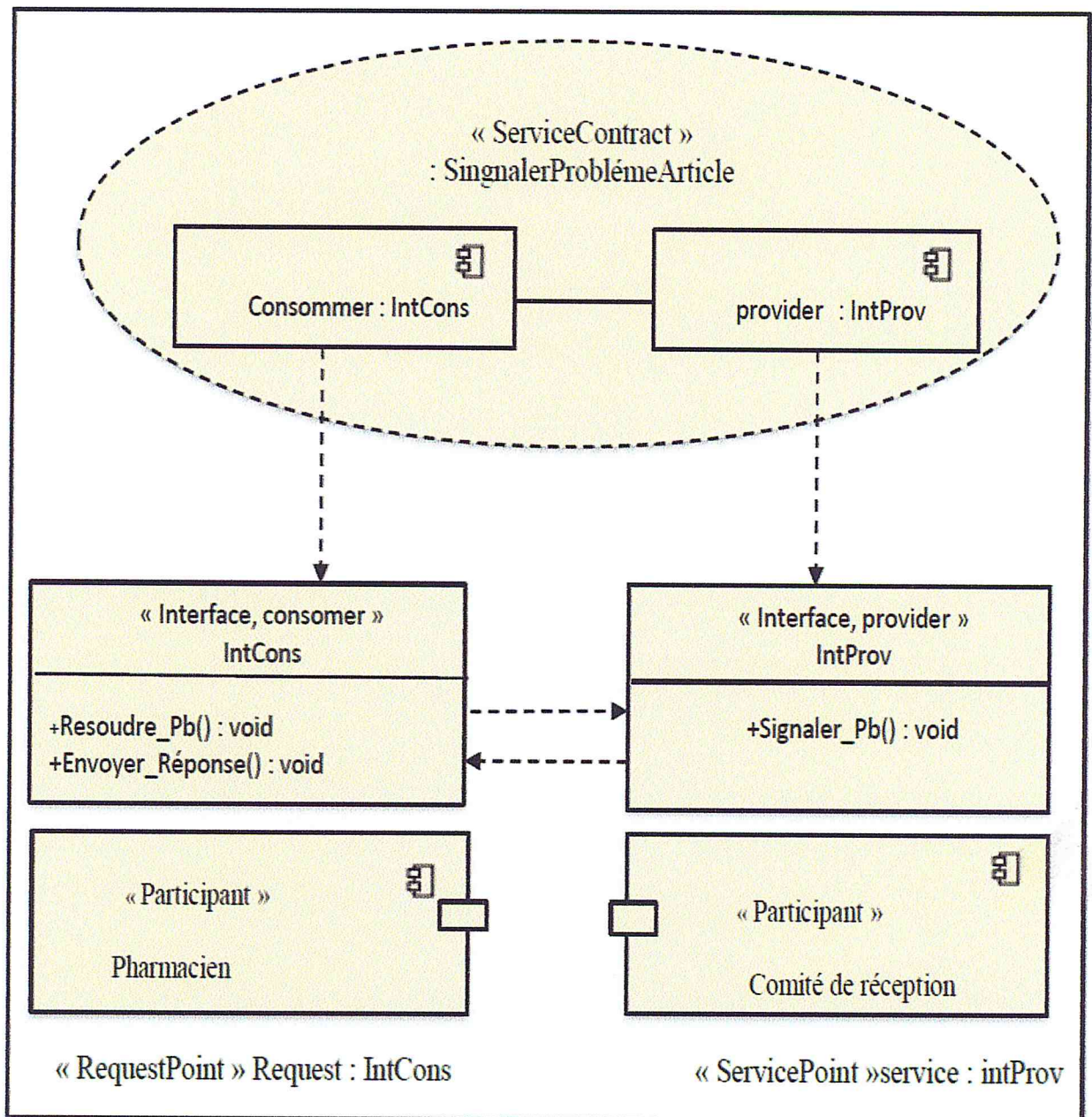


Figure 4:15 Le contrat de service « Signaler Problème Article ».

Synthèse : Après avoir modélisé le processus de gestion des stocks par soaml, nous avons extrait que des services métiers. Donc soaml ne modélise que l'aspect métier alors que notre objectif principal est de modéliser l'aspect décisionnel. Ainsi, nous allons proposer une architecture globale pour le système de gestion des stocks qui inclut l'aspect décisionnel.

4.2.4 Phase 4 : La réalisation

Nous proposons de développer les services et les déployer pour qu'il soit appelé. Le choix technique se focalise sur (système de gestion de base de données, environnement de développement, serveur d'applications, processus système de gestion de l'entreprise ... Etc.). La figure résume les différents choix, et donne une vue globale de la plate-forme de destination pour la gestion des stocks.

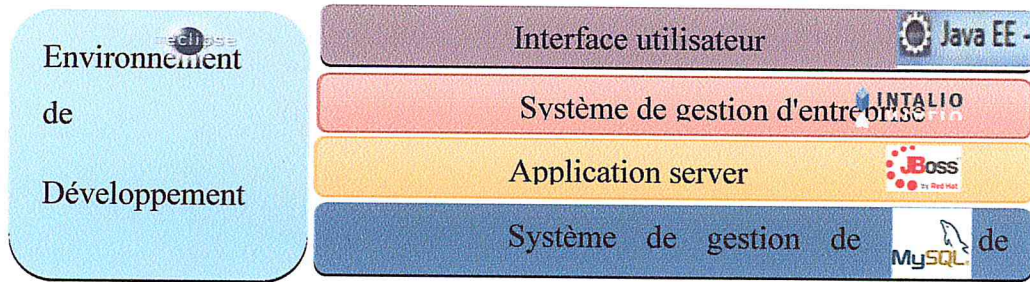


Figure 4 :16 L'architecture des composants techniques.

4.2.6.1 L'architecture des composants technique

✦ *Système de gestion de base de données*



- **MySQL Workbench** est un outil de conception et de modélisation visuelle des bases de données, proposé par mysql/Sun. Il permet aux développeurs et aux DBA de créer de nouveaux modèles de données physiques pour les bases de données mysql, mais aussi de modifier les bases de données physiques mysql existantes avec une rétro/pro-ingénierie et des fonctions de gestion des changements. MySQL Workbench a été conçu pour optimiser la productivité et

la réussite de l'utilisateur dans la conceptualisation, la communication, la création et la gestion des principales métadonnées d'entreprise, ainsi que des bases de données et des entrepôts de données à haute performance.

✚ **Serveur de Gestion des services web**

Jboss Web Service (jboss WS) est un framework de services web développés dans le cadre du serveur d'applications jboss.

Jbossws implémente les spécifications JAX-ws pour la définition des services web. Cette norme définit un ensemble d'annotations qui permettent la transformation d'un POJO (ou EJB3) en service web. Les annotations sont interprétées par le serveur d'application afin de transformer les classes Java en service web, il offre aussi d'autres services.

JAX-WS (Java API for XML based Web Services) est une nouvelle API, mieux architecturée, qui remplace l'api JAX-RPC, mais n'est pas compatible avec elle. Elle propose un modèle de programmation pour produire (côté serveur) ou consommer (côté client) des services web qui communiquent via des messages XML de type SOAP.

✚ **Moteur d'orchestration**

Le BPM est devenu un intermédiaire incontournable entre l'it et le métier. **Intalio|BPMS** et **jboss JBPM** sont les ténors dans le monde Open Source. Les deux outils ont les mêmes objectifs, malgré leurs différences idéologiques.

- **Intalio|BPMS** se base sur le standard BPMN pour la modélisation des processus métiers. A partir du BPMN il génère du BPEL pour l'orchestration des processus. Il est conçu pour les utilisateurs ayant peu d'expérience technique.
- **Jboss JBPM** ne supporte pas le standard BPMN, il se base sur le JPDL et est conçu pour les développeurs.

Par conséquent, nous avons choisi **Intalio|BPMS** pour la gestion des processus métiers.

✚ **Présentation de la solution INTALIO|BPMS**



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

- **Intalio|Designer** est l'outil de modélisation et de conception des processus métiers fondé sur la plateforme Eclipse. Permet de créer les diagrammes de processus à l'aide de la notation BPMN et de compléter ces modèles en intégrant les Services Web et autres détails nécessaires

à l'exécution des processus métiers. Cet outil traduit automatiquement les diagrammes BPMN en code source BPEL d'exécution des processus.

- **Intalio|Server** comporte le moteur d'exécution des processus métiers en interprétant le langage BPEL. Il propose également l'ensemble des fonctionnalités d'administration.
 - **Intalio|Workflow** est basé sur la nouvelle extension BPEL4People il supporte l'exécution des activités humaines par le biais d'interfaces utilisateur AJAX générées automatiquement par le moteur xforms.
- ⇒ Intalio|Workflow utilise Intalio|Server pour l'exécution des processus BPMN.

✚ *Serveur d'application*



Jboss Application Server (jboss AS) est un serveur d'applications entièrement écrit en Java, publié sous licence GNU LGPL. Parce que le logiciel est écrit en Java, jboss AS peut être utilisé sur tout système d'exploitation fournissant une machine virtuelle Java(JVM).

✚ *Environnement de développement*



Notre choix s'est porté vers l'edi, qui permet de créer des applications Java EE et Web, incluant des outils pour javaee, JPA, JSF, ejbet d'autres.

Par conséquent en a installé eclipse Luna.

La figure suivante résume les divers choix effectués, et donne une vue globale de la plateforme de destination.

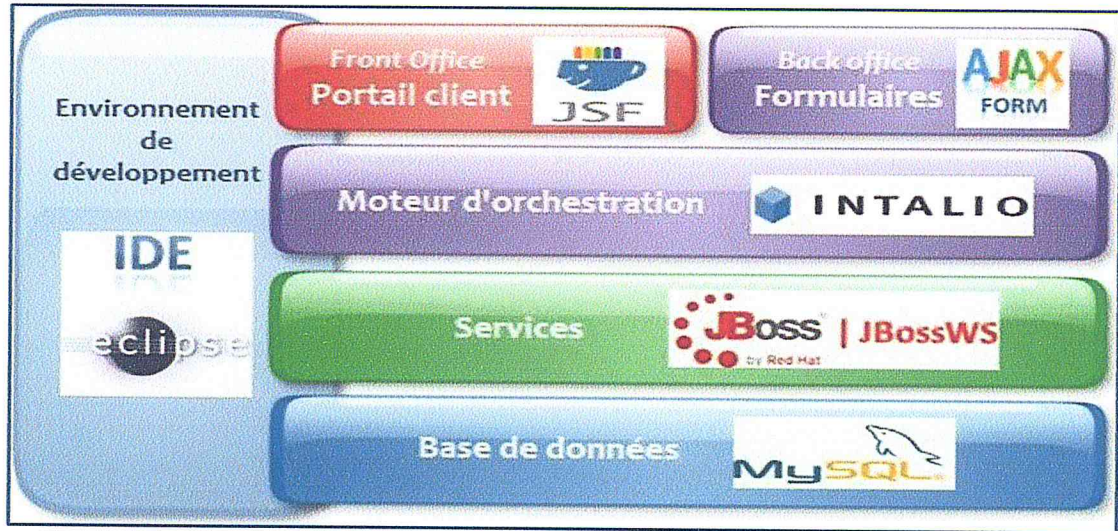


Figure 4:17 Les choix techniques de notre solution.

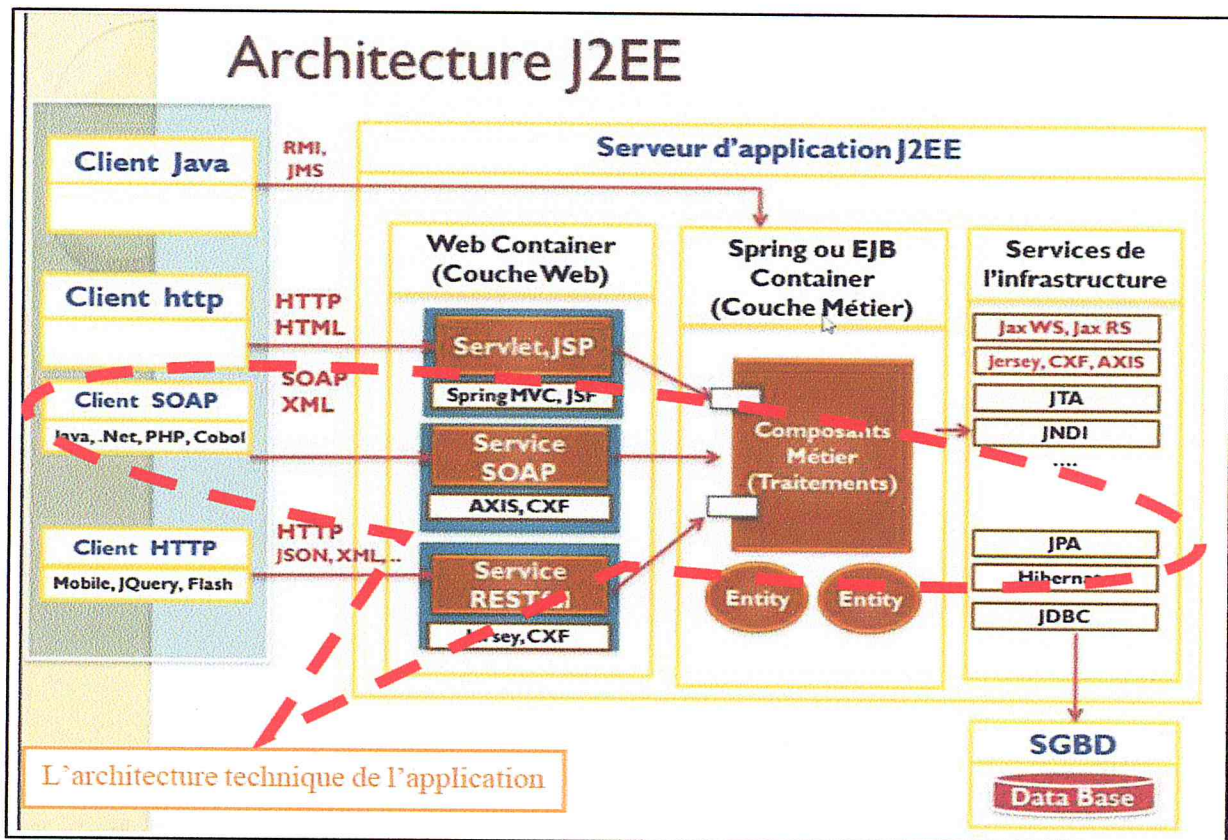


Figure 4:18 Architecture des composants technique de notre solution.

4.2.4.1 La couche métier

Pour chaque machine logique (MLM) de la couche métier (Figure V - 2) correspond un service CRUD³, qui permet de créer, rechercher, mettre à jour et supprimer l'information dans les référentiels du système d'information.

```
J FactureMB.java ☒
 2
 3+ import javax.annotation.PostConstruct;
11
12
13 @ManagedBean
14 @ViewScoped
15 public class FactureMB {
16     @Inject
17     private FactureFacade factureFacade;
18
19     private Facture facture;
20
21     @PostConstruct
22     public void init() {
23         setFacture(new Facture());
24         int id = (int) FacesContext.getCurrentInstance().getExternalContext()
25             .getFlash().get("id");
26
27         this.facture = factureFacade.find(id);
28     }
29
30     public Facture getFacture() {
31         return facture;
32     }
33
34     public void setFacture(Facture facture) {
35
36         this.facture = facture;
37     }
38
39
40     public String update() {
41         factureFacade.edit(facture);
42         return "liste?faces-redirect=true";
43     }
44 }
45
```

Figure 4:19 Service CRUD.

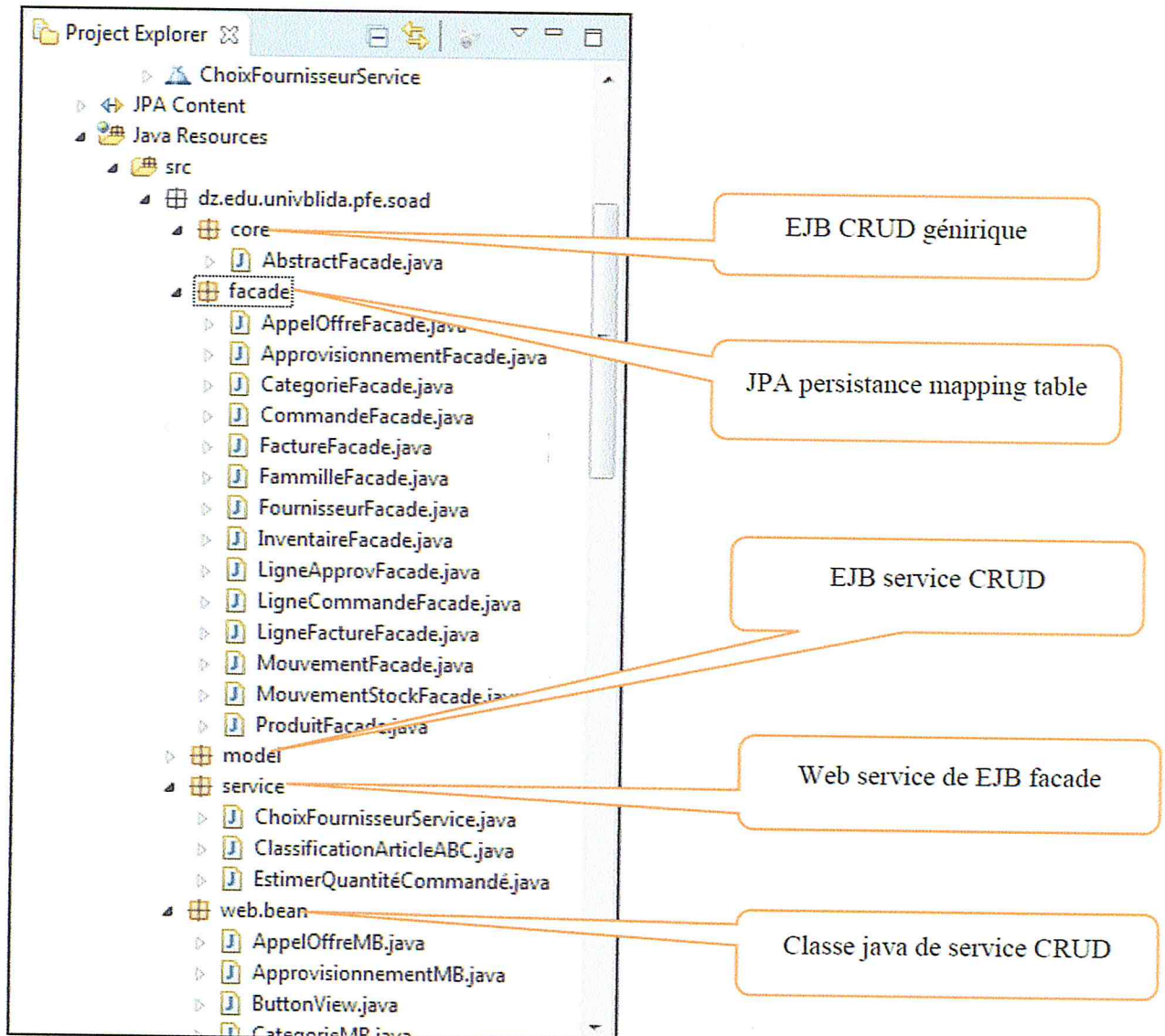


Figure 4:20 l'architecture des couches.

4.2.4.2 La couche organisation

Les machines logiques (MLO) de la couche organisation (Figure V - 2) sont réalisés par des services web, ces derniers seront par la suite orchestrés par le processus BPM, ou appelés directement par le portail client. Le processus BPM à son tour devient un WS.

4.2.4.2.1 Les services web :

Deux approches peuvent être utilisées pour la création de Web Service :

➤ **Bottom-Up**

Création d'un POJO (Plain Old Java Object) un langage donné, puis génération du WSDL et des classes nécessaires. Cette approche masque la partie définition WSDL, elle permet surtout de réutiliser du code existant.

➤ **Top-Bottom**

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

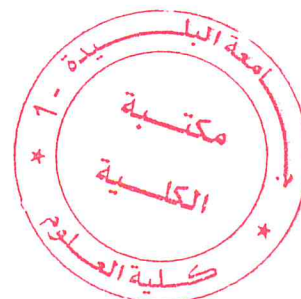
Les Services Web de notre application sont créés suivant l'approche Bottom-Up à l'aide du Framework Axis et déployés sur le serveur Tomcat.

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

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



Figure 4:21 Exemple d'un fichier WSDL .



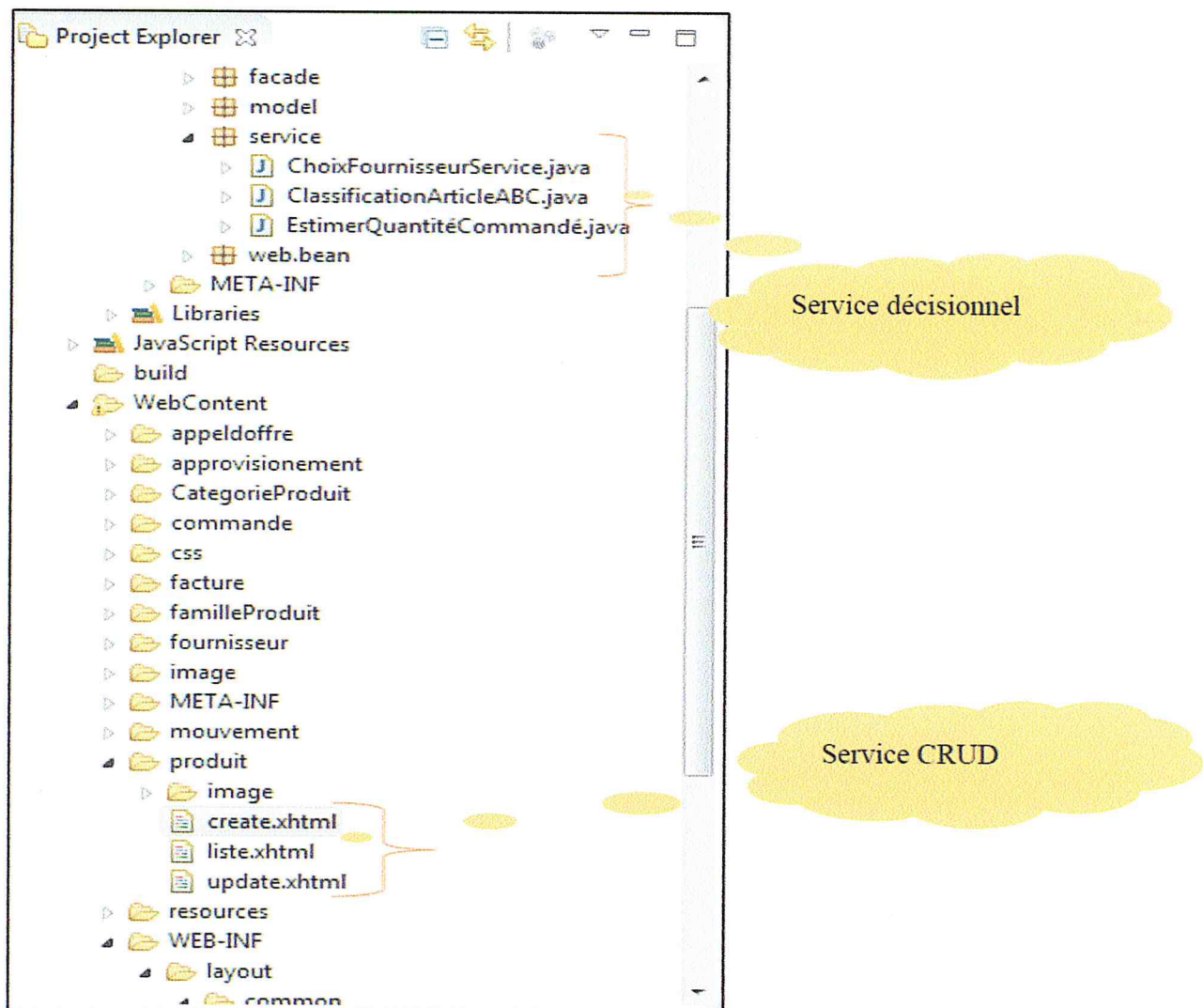


Figure 4:21 Les services CRUD et décisionnels

4.2.4.2.2 Le processus métier

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

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

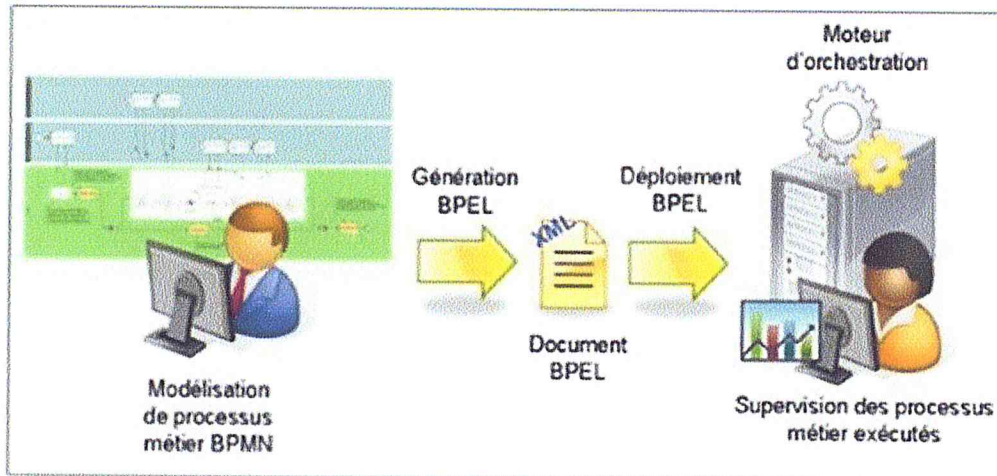


Figure 4:22 Les étapes de transposition des processus métiers.

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

A. La définition des rôles des intervenants du Workflow

Se traduits en pool dans le diagramme BPD. La Figure suivante illustre la représentation des intervenants des processus dans sa définition graphique.

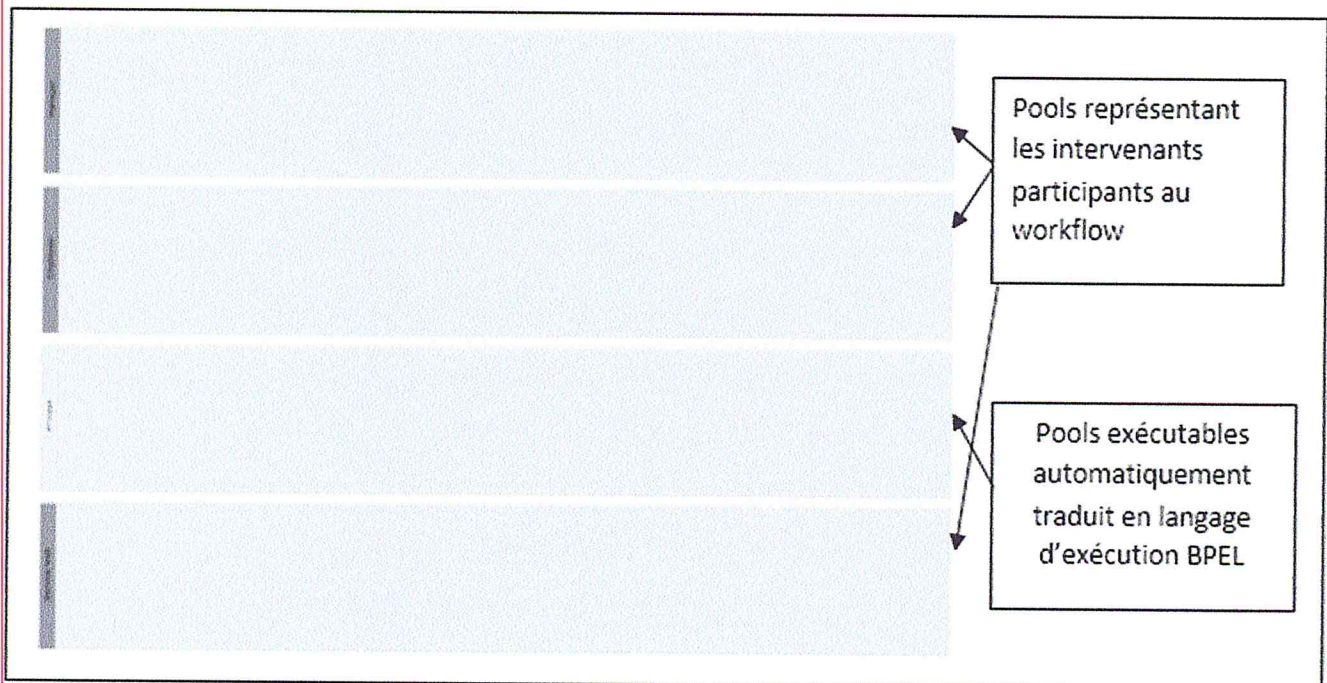


Figure 4:23 Définitions des rôles.

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

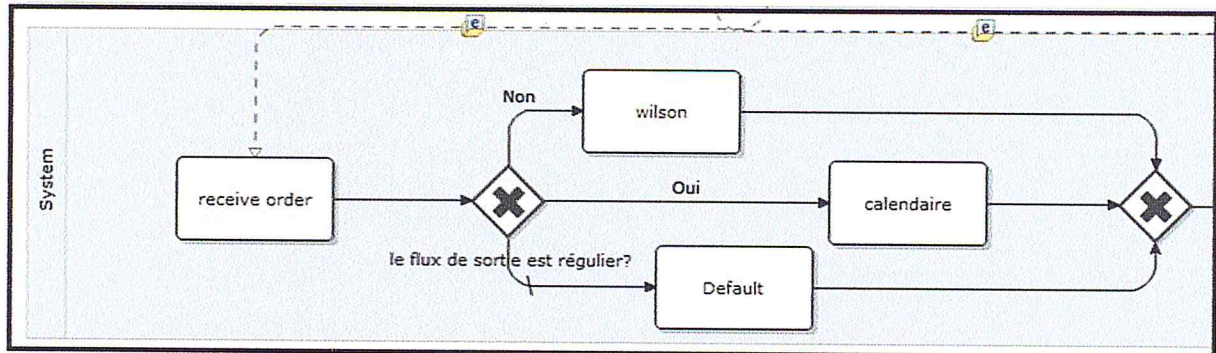


Figure 4:24 Partie du sous processus Analyse Stock.

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

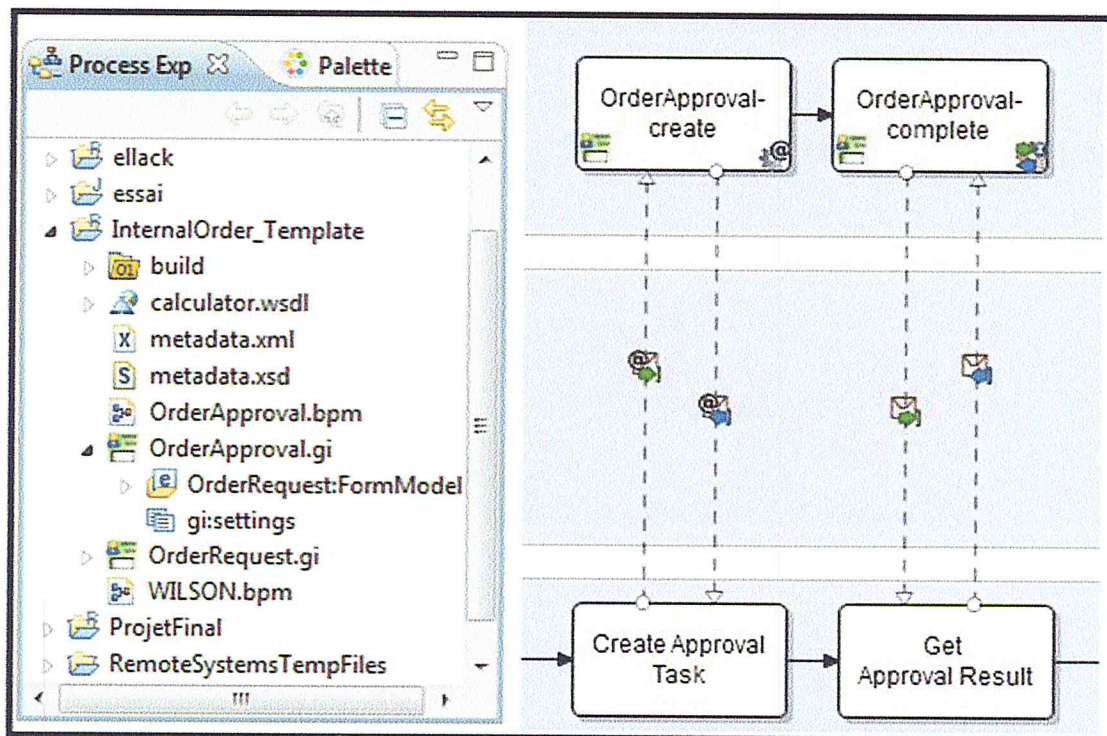


Figure 4:25 Intégration des formulaires.

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

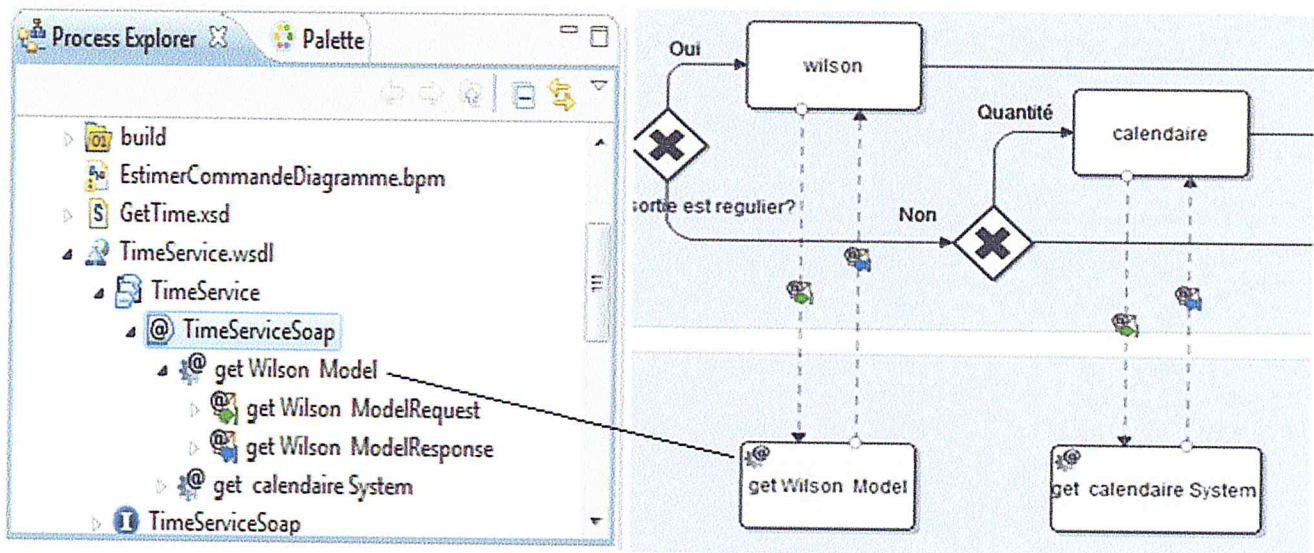


Figure 4:26Intégration des Services Web.

E. Le déploiement des diagrammes BPD sur Intalio|Server :

Un diagramme BPD déployé sur le serveur devient lui-même un service Web. Il est éventuellement appelé depuis Intalio|Workflow via un navigateur web.

4.2.4.3 La couche Présentation

Ces formulaires représentent les tâches à réaliser ou les notifications provenant dans l'exécution du processus. Les données saisies sont véhiculées tout au long de l'exécution des tâches du processus.

4.2.4.3.1 Les interfaces :

Dans ce qui suit nous allons décrire les fonctionnalités des différentes interfaces qui constituent notre application.

En effet notre système est décomposé de quatre acteurs :

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



Figure 4:27 La fenêtre Accueil.

Pour commencer l'exploitant doit se connecter à son espace réservé par une saisie de ses coordonnées, une fois le système a vérifié les coordonnées, l'exploitant se dirige vers son espace de travail.

✚ Espace Pharmacien :

Dans notre application nous avons réservé un espace pour l'administrateur pharmacien. Via cet espace il peut gérer les services décisionnels intégrés ainsi que la base de données en la sauvegardant ou en l'instaurant ou bien en l'alimentant. Il peut gérer également les comptes utilisateurs et accéder à la console de l'exécution des processus 'Intalio|Console'.

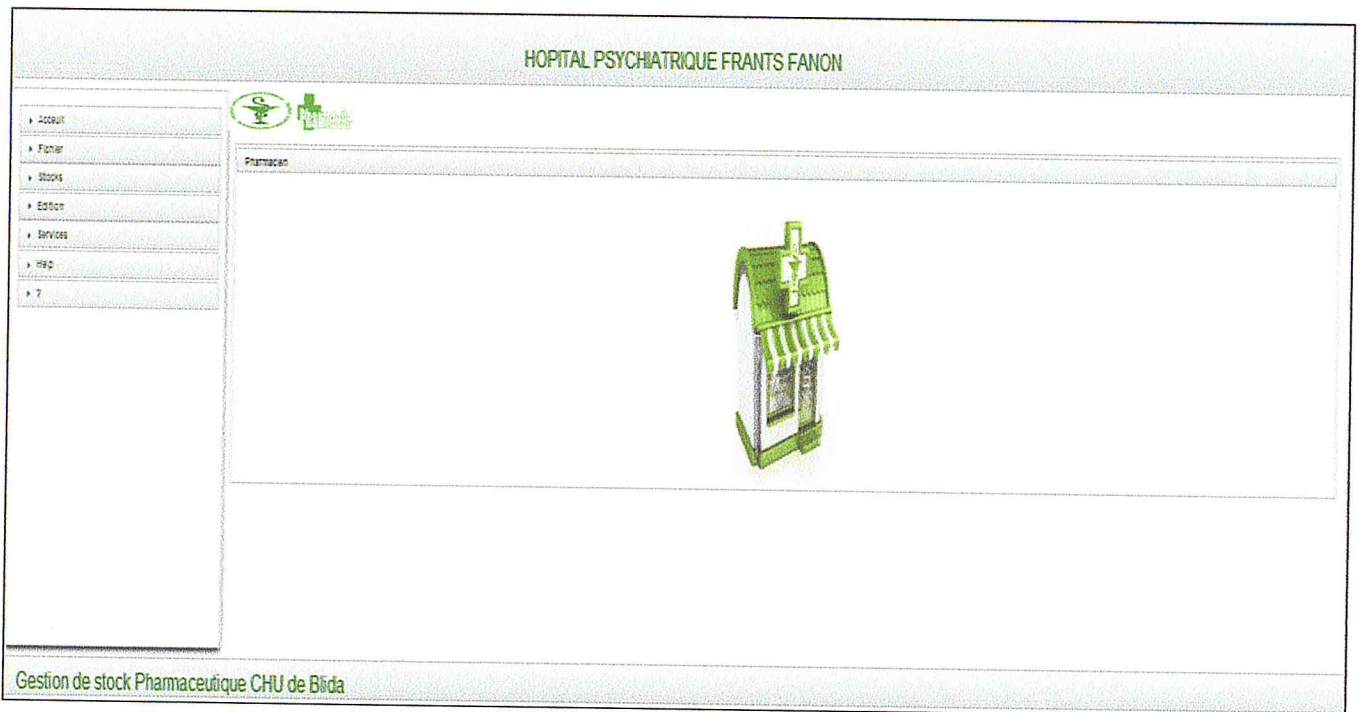


Figure 4:28 Espace pharmacien.

✚ Espace magasinier :

Cet espace est réservé au magasinier qui se charge de la gestion physique des stocks, elle lui permet d'effectuer des opérations telles que : l'établissement des inventaires physiques, l'édition des bons d'entrée et sortie stock. Il peut également consulter la liste des produits en stock et effectuer des bons de livraison client.



Figure 4:29 Espace magasinier.

✚ Espace department finance :

Cet espace est réservé au département de finance qui se charge de vérifier les factures et effectuer le paiement après l'avoir avisé par le pharmacien .




Figure 4:30 Espace département finance.

✚ Service CRUD

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

HOPITAL PSYCHIATRIQUE FRANTS FANON

- ▶ Accueil
- ▶ Fichier
- ▶ Stocks
- ▶ Edition
- ▶ Help
- ▶ ?



 * Ajouter * Modifier * Supprimer

Liste des Produits de la pharmacie

| codeProd | Libellé | Prix-Achat | Stock Max | Stock Min | Stock Alerte | Famille | Catégorie |
|----------|---------|------------|-----------|-----------|--------------|---------|-----------|
| 18 | 18 | 12.00 | 120 | 10 | 45 | Famille | Cart |
| 19 | 22 | 12.0000 | 200 | 100 | 100 | | |
| 20 | 23 | 12.00 | 2000 | 220 | 220 | | |
| 21 | 200 | 200.0 | 200 | 500 | 10 | | |
| 22 | 200 | 40.000 | 240 | 500 | 10 | | |

montrer la liste des produit

Gestion de stock Pharmaceutique CHU de Blida

Figure 4:31 services CRUD

HOPITAL PSYCHIATRIQUE FRANTS FANON

Accueil
Fichier
Stocks
Edition
Help
?

Ajouter Modifier Supprimer

Tableau d'estimation de la quantité optimale a commander

| Nombres de commandes | SI | SF | Stock moyen en quantité | Stock moyen en valeur | Coût de détention du stock | Coût de passation des commandes | Coût total | Quantité économique à commander |
|----------------------|----|-------|-------------------------|-----------------------|----------------------------|---------------------------------|------------|---------------------------------|
| 1 | 0 | 300 | 150.00 | 330.00 € | 33.00 € | 2 € | 35.00 € | 300 |
| 2 | 0 | 150 | 75.00 | 165.00 € | 16.50 € | 4 € | 20.50 € | 150 |
| 3 | 0 | 100 | 50.00 | 110.00 € | 11.00 € | 6 € | 17.00 € | 100 |
| 4 | 0 | 75 | 37.50 | 82.50 € | 8.25 € | 8 € | 16.25 € | 75 |
| 5 | 0 | 60 | 30.00 | 66.00 € | 6.60 € | 10 € | 16.60 € | 60 |
| 6 | 0 | 50 | 25.00 | 55.00 € | 5.50 € | 12 € | 17.50 € | 50 |
| 7 | 0 | 42.86 | 21.43 | 47.14 € | 4.71 € | 14 € | 18.71 € | 43 |
| 8 | 0 | 37.5 | 18.75 | 41.25 € | 4.13 € | 16 € | 20.13 € | 38 |
| 9 | 0 | 33.33 | 16.67 | 36.67 € | 3.67 € | 18 € | 21.67 € | 33 |
| 10 | 0 | 30 | 15.00 | 33.00 € | 3.30 € | 20 € | 23.30 € | 30 |
| 11 | 0 | 27.27 | 13.64 | 30.00 € | 3.00 € | 22 € | 25.00 € | 27 |
| 12 | 0 | 25 | 12.50 | 27.50 € | 2.75 € | 24 € | 26.75 € | 25 |
| 13 | 0 | 23.08 | 11.54 | 25.38 € | 2.54 € | 26 € | 28.54 € | 23 |

Gestion de stock Pharmaceutique CHU de Blida

Figure 4:32 Tableau d'estimation de quantité à commander.

4.2.4.3.2 Les formulaires :

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

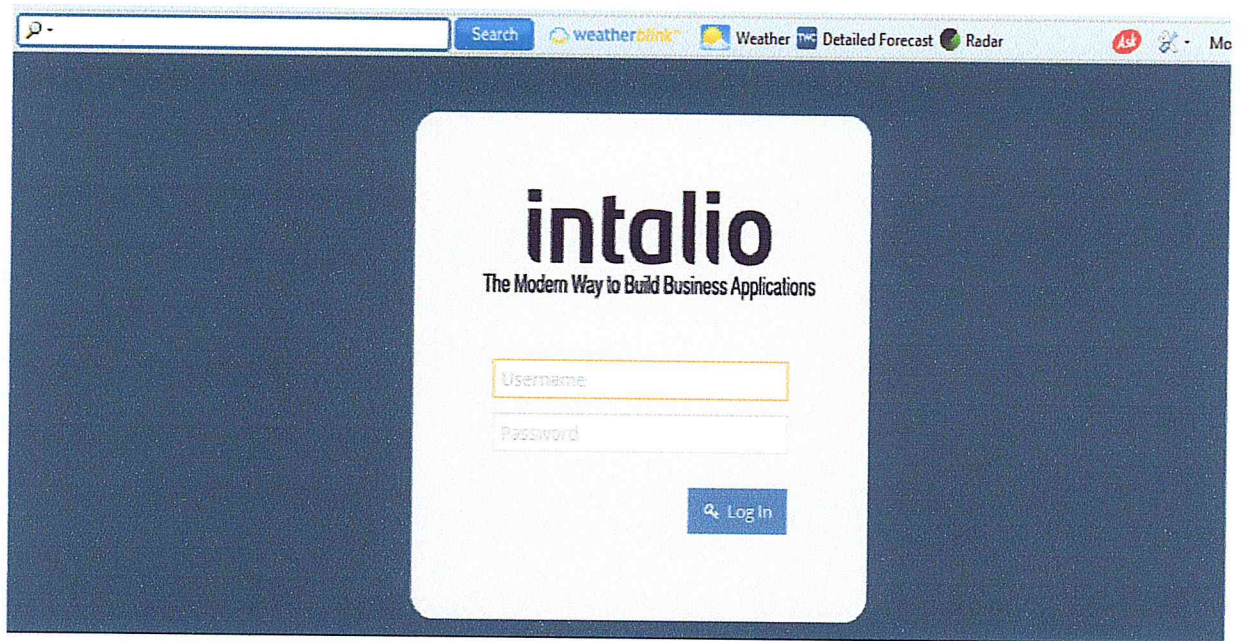


Figure 4:34 Intalio|Workflow «Authentification».

Après l'authentification, la fenêtre suivante apparaît à l'administrateur des stocks

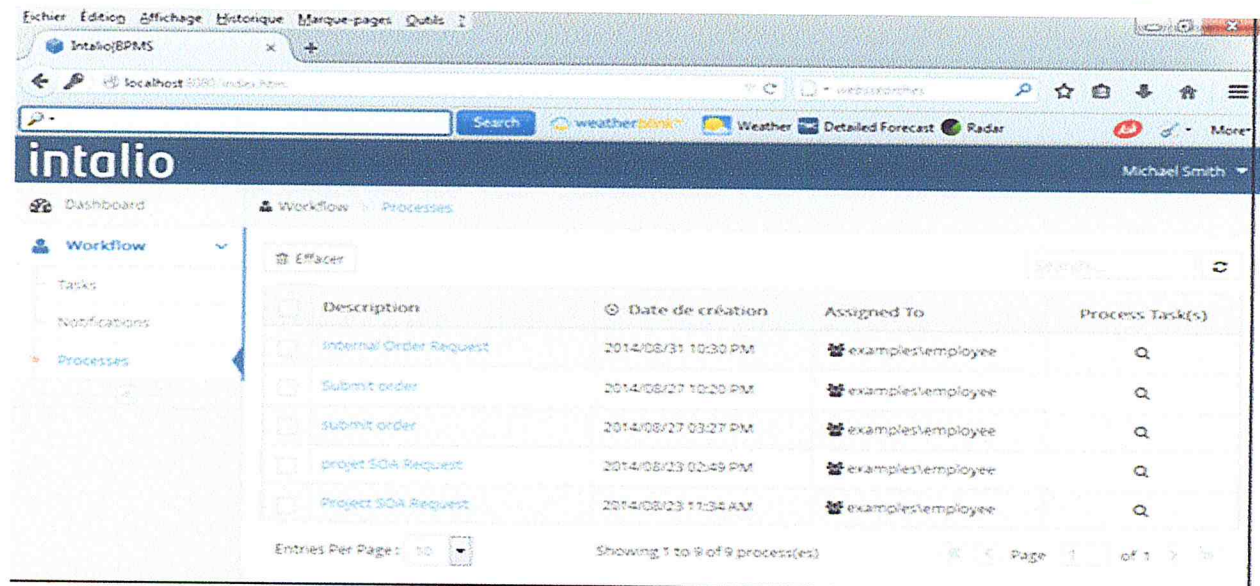


Figure 4:35 Intalio|Workflow «Liste des tâches».

Cette fenêtre présente les tâches associées à cet utilisateur, les messages qu'il a reçus ainsi que les processus qu'il peut déclencher.

Un formulaire apparaît à l'utilisateur dès qu'il a choisi la tâche à réaliser. Il peut ainsi:

The screenshot shows a web form titled 'Workflow' with a sidebar containing 'Tasks', 'Notifications', and 'Processes'. The main content area contains the following fields:

- Product:** Bus (dropdown menu)
- Price:** 234 (text input)
- Quantity:** 34 (text input)
- Approval:**
 - Decision:** Radio buttons for 'approved' and 'rejected'.
 - Comment:** Text input field.

Figure 4:36 Formulaire de commande.

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

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

The screenshot shows the Intalio console interface. The main content area displays a table of process instances with the following data:

| Process | Life Cycle | Instance Count | | | | | |
|---|------------|----------------|-----------|---------|--------|-----------|-----------|
| | | In Progress | Completed | Failure | Failed | Suspended | Terminate |
| InternalOrder_Template [v18] Deployed On: 2014/08/31 10:30:24 PM. Deployed By: admin | | | | | | | |
| ...approvalProcess | ... | 2 | - | - | - | - | - |
| WILSONSystem | ... | - | - | - | - | - | - |

At the bottom of the table, it indicates 'Showing 1 to 10 of 43 process(es) in 5 package(s)' and 'Page 1 of 5'.

Figure 4:37 Intalio console.

Conclusion

Les entreprises y compris les hôpitaux, ne peuvent vivre ou survivre si elles ne développent pas des mécanismes de gestion efficaces. Or l'efficacité impose la rationalisation de la gestion. La gestion des stocks étant une partie très importante de la gestion, on comprend dès lors comment les entreprises pour assurer un service régulier et satisfaisant à la clientèle doivent disposer d'un niveau de stock optimum.

Conclusion générale

Conclusion générale

Conclusion générale

Le travail de PFE qui nous a été confié consistait à contribuer à une étude qui vise à intégrer les aspects décisionnels dans une architecture technique de type SOA pour le système de gestion des stocks de la pharmacie centrale de Blida. Pour cela, nous avons effectué un travail indispensable de recherche et d'analyse sur l'architecture orientée services et l'automatisation de processus métier.

Après l'étude du domaine de la gestion des stocks, nous avons retenu les méthodes d'analyse des stocks (ABC et PARETO) ainsi que les méthodes de réapprovisionnement (le modèle WILSON, la gestion calendaire et le point de commande). Ces méthodes ont un intérêt certain pour la gestion des stocks.

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

La conception été a abordée avec la méthode soa+d avec les quatre phase illustré dans le chapitre 1, ainsi lié au concept de l'approche soaml.

En ce qui concerne la partie réalisation nous avons utilisé le SGBD mysql workbenchI, l'environnement de développement Eclipse Luna et le conteneur de servlet jboss ws.

Nous avons également utilisé l'outil « Intalio » pour la modélisation et l'exécution de processus métier lié à la gestion des stocks.

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

- ✚ L'application de notre architecture sur différentes applications d'aide à la décision et pourquoi pas sur un SIAD,
- ✚ La mise en œuvre d'un ESB (donner accès aux services web pour différents utilisateurs),
- ✚ Implémenter d'autres méthodes de gestion des stocks.

Annexes

➔ **Annexe 1 : Aperçu du standard BPMN.**

Annexe 1

Aperçu du standard BPMN

1. Business Process Modeling Notation (BPMN)

Cette partie présente le formalisme de modélisation de processus **BPMN** en se basant sur le document de travail de **BPMI** (*Business Process Management Initiative*) : « Business Process Modeling Notation » du 25 août 2003. Nous allons pour cela articuler cette description autour de :

- Cadre d'application de BPMN
- Éléments de modélisation du formalisme
- Principe de modélisation (Exemple)

Pour donner une image et se faire une première idée, on peut très schématiquement positionner cette notation comme héritière à la fois des Réseaux de Petri et des Statecharts (même s'il ne s'agit absolument pas d'une description d'états mais bien d'un diagramme d'activité). Elle présente en effet le *Business Process* comme un *enchaînement d'activités, processus et tâches* (activés par des *Token*) qu'on peut relier par le biais de *connexions* décrivant le flow (conditionnelles ou non) et caractérisés grâce aux *attributs* qui définissent chaque élément de modélisation et grâce au principe de *décomposition* d'une activité en sousprocessus.

2. Cadre d'application de BPMN

BPMN est destiné à la modélisation de processus vus en tant que workflow. BPMN n'est pas adapté aux modélisations organisationnelles ou structurelles. Le formalisme s'adapte au niveau d'abstraction auquel on souhaite se placer et permet (principe de

décomposition) de raffiner à volonté une description ou une partie d'une description de processus.

BPMN ne permet qu'une description dynamique du processus sans définir les composants de l'organisation qui seront responsables des tâches identifiées. Il s'agit d'une description « *fonctionnelle* » (identification d'activités) sans se préoccuper des entités physiques réelles qui implémenteront et réaliseront les actions identifiées.

Les diagrammes BPMN peuvent être directement reliés à du code BPEL4WS. Le document étudié (« Business Process Modeling Notation » de BPMI du 25 août 2003) présente comment traduire chaque configuration graphique issue d'un schéma BPMN en code BPEL4WS.

Enfin, il est important de noter que comme tout formalisme de modélisation, BPMN est dépendant du point de vue adopté : la modélisation d'un processus ne sera pas la même selon qu'on se place à tel ou tel point de vue.

3. Eléments de modélisation de BPMN

3.1 Présentation sommaire

BPMN décrit les activités à l'aide de *composants* reliés par des *liens* et groupés dans des *conteneurs*.

Composants de modélisation :

- Les événements (*events*)
- Les activités (*activities*)
- Les connexions (*gateways*)

Connecteurs graphiques :

- Les séquences (*sequence flows*)
- Les messages (*message flows*)
- Les associations (*associations*)

Conteneurs d'entités :

- Conteneur graphique (*pools*)
- Sous-conteneur (*lanes*)

Un diagramme BPMN est également caractérisé par ses attributs :

- **ID** (identifiant du diagramme),
- **Name** (description du diagramme),
- **Version** (numéro de version),
- **Author** (auteur du diagramme),
- **Language** (langue de conception),
- **CreationDate** (date de création),
- **Process*** (ensemble des processus modélisés : 0..n),
- **Pool+** (les conteneurs composant le diagramme : 1..n),
- **Documentation?** (éventuelle documentation complémentaire)

3.2 Les éléments de modélisation

3.3 Les Composants

Les objets de modélisation (*événements*, *activités* et *passages*) disposent d'attributs communs :

- **ID** (identifiant du composant)
- **Name** (description du composant)
- **Assign*** (expressions évaluées à l'arrivée d'un token : 0..n)
- **Pool** (le conteneur dans lequel se trouve le composant)
- **Lane*** (la ou les lignes auxquelles appartient le composant)
- **Documentation ?** (éventuelle documentation complémentaire : 0..1)

3.3.1 Evènements (Events)

Les évènements peuvent être de trois types différents : *start* (événement démarrant un processus), *intermediate* (événement en cours de processus) et *end* (événement terminant un processus).

Les règles de construction sont les suivantes :

- ➡ Il peut y avoir zéro ou plusieurs START EVENT et END EVENT.

- S'il y a (au moins) un START EVENT, il doit y avoir (au moins) un END EVENT.
- S'il y a (au moins) un END EVENT, il doit y avoir (au moins) un START EVENT.
- S'il n'y a pas de START EVENT, alors, les événements n'ayant pas de flow entrant constituent les points d'entrée du processus (ils doivent tous être « activés » lors de l'initialisation du processus).
- S'il n'y a pas d'END EVENT, alors, les événements n'ayant pas de flow sortant constituent les points de sortie du processus (on doit attendre qu'ils soient tous « validés » pour terminer le processus).




| Type d'évènement | Symbole | Attributs spécifiques (en plus des attributs communs) |
|--------------------|---|--|
| START EVENT |  | Trigger (le type du déclencheur de l'évènement parmi les types : <i>none, message, timer, rule, link...</i>) Message/Timer/Rule... (le contenu du Trigger) |
| INTERMEDIATE EVENT |  | Trigger (le type du déclencheur de l'évènement parmi les types : <i>none, message, timer, rule, link ...</i>) Message/Timer/Rule... (le contenu du Trigger) |
| END EVENT |  | Result (le Type de la conséquence de l'évènement parmi les types : <i>none, message, exception, compensation, link...</i>) Message/Exception/Compensation... (le contenu du Trigger) |

Tableau1 : Les evenement de bpmn .

3.3.2 Activités (Activities)

Les activités peuvent être élémentaires (*Task*), décomposables (*Sub-Process*) ou globales (*Process*). La hiérarchisation des *Business Process, Process, Sub-Process* et *Task* peut être schématisée comme suit :

Les *Process* ne disposent pas de symbole et l'on peut considérer qu'ils sont représentés graphiquement à l'aide des *Pools* (cf. § conteneurs). Les *Sub-Process* et *Task* sont représentés comme suit :

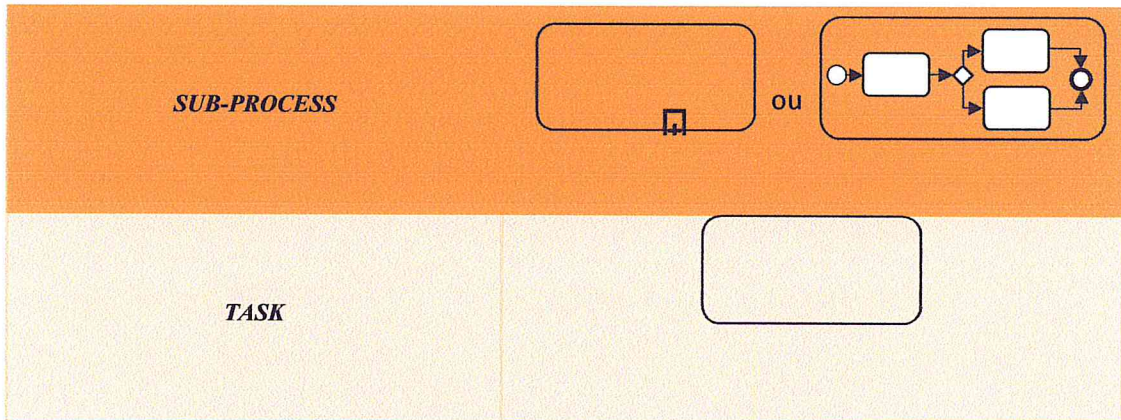
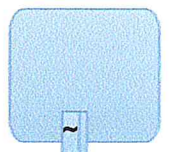


Tableau2 : Les sous processus.

Les activités (*Task* et *Sub-Process*) peuvent présenter des caractéristiques graphiques supplémentaires traduisant certains aspects complémentaires de leur nature ou de leur comportement :

- + **Loop** : il s'agit d'une activité qui pourra être exécutée en boucle (la condition d'arrêt sera un événement lié à l'activité de type *Timer* par exemple)
- + **Multiple Instance** : l'activité peut être instanciée plusieurs fois en parallèle (par exemple, chaque chapitre d'un livre peut être rédigé selon le même processus, mais tous les chapitres peuvent être écrits en parallèle)
- + **AdHoc** : les activités décrites dans ce *sub-process* peuvent ne pas être ordonnées (par exemple, le nettoyage d'une pièce correspond au nettoyage spécifique de différentes parties sans que le séquençage soit crucial)
- + **Compensation** : activité (faisant suite à un événement de compensation) déclenchée si le *sub-process* global a été annulé afin de compenser l'une des activités spécifiques du *sub-process*.



Ces caractéristiques des activités suivent les règles suivantes :

- Une activité peut comporter de 0 à 3 caractéristiques (indiquées en bas du symbole graphique)
- Une activité ne peut pas comporter simultanément les caractéristiques **Loop** et **Multiple Instance**
- Une tâche ne peut comporter la caractéristique **AdHoc** (réservé au seul *Sub-Process*)

Les activités de type *Process*, *Sub-Process* et *Task* présentent de nombreux attributs complémentaires précisant en particuliers les propriétés de l'activité ainsi que les éléments complémentaires liés aux caractéristiques **Loop**, **Multiple Instance**, **AdHoc** et **Compensation**.

3.3.3 Passages (Gateways)

Les « passages » ou « aiguillages » (*Gateways*) servent à contrôler l'évolutions des flows dans le processus (convergence, divergence...). Il existe plusieurs sortes de *Gateways* à sémantiques distinctes :



no



XOR Data-Based : les flows sortants de ce *gateway* correspondent à diverses alternatives. Les tests conditionnant les embranchements sont évalués dans l'ordre et dès que l'un d'eux est valable, le token est transmis par le connecteur qui lui est relié. Ainsi, un seul chemin peut être pris (exclusif). Il est souvent utile de placer une option finale « Defaut » permettant de garantir le passage (BPMN ne prévoit pas le cas ou aucune alternative n'est possible en cours d'exécution). Utilisé en tant que « merge » (point de jonction) de différents flows, ce *gateway* est **UNE** valve ouverte laissant passer successivement tous les tokens qui lui arrivent.



XOR Event-Based : Le principe de fonctionnement de ce *gateway* est identique au précédent à la différence près que les conditions déterminant quel chemin sera emprunté par le token dépendent d'un événement « attaché » au *gateway*





OR : Les flows sortant de ce *gateway* sont conditionnés par des tests. Cependant, le fait qu'un chemin soit validé n'exclue pas l'évaluation des autres (comme pour le XOR). Ainsi, chaque condition vérifiée donnera lieu à un token. Comme pour le XOR, il est nécessaire de s'assurer qu'au moins l'une des option sera valable (BPMN ne prévoit pas le cas où aucune alternative n'est validée). Utilisé en tant que « merge » de différents flows, ce *gateway* est un synchronisateur, il fusionne les tokens qui lui arrive en simultanément.



INCLUSIV : Ce *gateway* nécessite que tous les flows entrant soient actifs lorsqu'il est utilisé comme « répartiteur ». Lorsqu'il est utilisé comme « merge », ce *gateway* fournit un token à chaque chemin sortant.



COMPLEX : Ce *gateway* sert à réaliser les situations qui n'ont pas pu être modélisées

Les *gateways* présentent de un certain nombre d'attributs précisant par exemple le type du *gateway*, le nombre de connexions qu'il exprime, leurs caractéristiques...

3.3.4 Les Conteneurs

Les conteneurs graphiques sont de deux types correspondant à une « hiérarchisation de la contenance » : les *pools* qui se décomposent éventuellement en *lanes*. Ces partitionnements présentent en particulier un intérêt pour représenter les *Business process* dans un environnement collaboratif « B2B ».

3.3.5 Pool

Ce conteneur permet d'isoler un ensemble de tâches du reste du modèle lorsqu'on représente une situation « Business to Business ». La *pool* peut être utilisée en tant que « boîte noire » (i.e. on ne représente pas le ou les processus qu'elle contient) ou en tant que « boîte blanche »

(i.e. on détaille l'intérieur). Les seuls *flows* circulant entre les *pools* sont des *messages flows* (dans la description interne d'une *pool*, il peut bien entendu y avoir d'autres *flows* type *sequence flows*).

3.3.6 Lane

Il s'agit d'une sub-division de la *Pool*, destinée à organiser la *pool* (selon un critère de décomposition choisi par le modelleur). Les *lanes* peuvent elles-même être décomposées en *sublanes*. Les *pools* et les *lanes* se représentent comme suit :

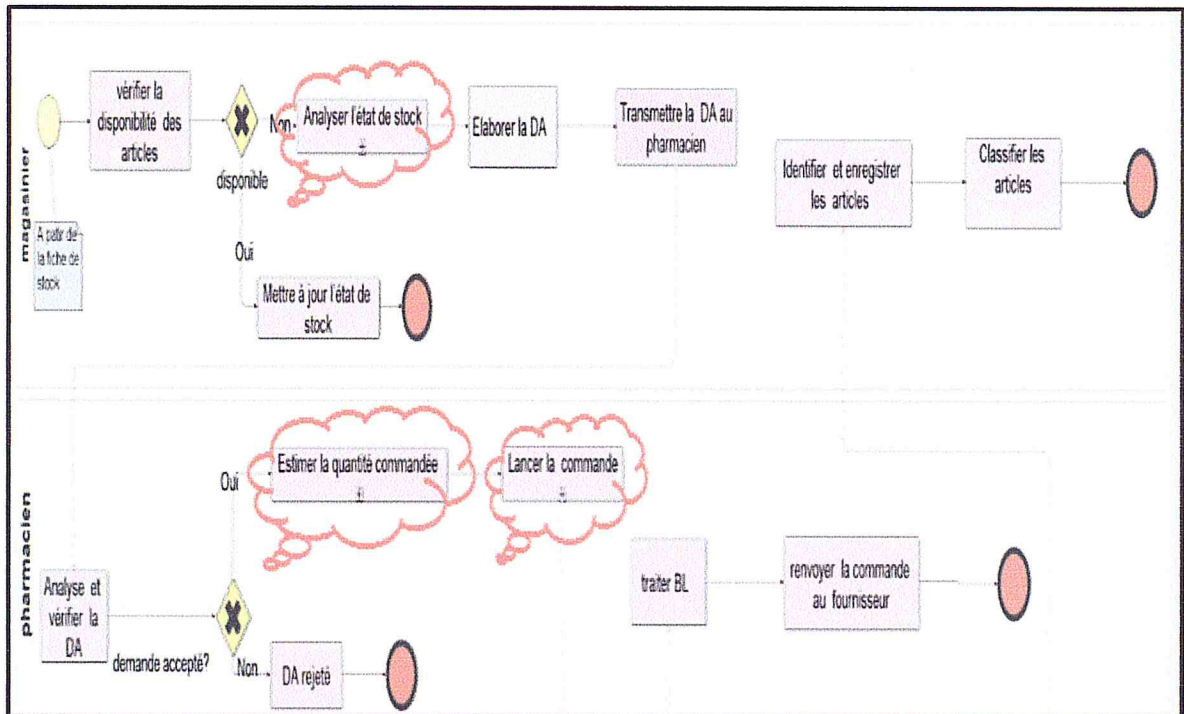


Figure1 : les lanes et les figure.

3.3.7 Les Connecteurs

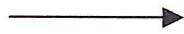
Les connecteurs graphiques se classent selon qu'ils sont liés à un flow (*séquence* et *messages*) ou à une association graphique (par exemple d'un commentaire à un élément de modélisation) par le biais d'une connexion de type *association*. Usuellement, les *séquences* sont placées horizontalement et les *messages* verticalement (mais ce n'est qu'une convention de clarté).

3.3.7.1 Séquence flow

Ce connecteur est utilisé pour représenter l'ordre dans lequel les activités seront exécutées. Ce flow n'a qu'une seule origine et qu'une seule destination (toute deux de type *event*, *activity* ou *gateway*, et seulement ceux-là). Le *sequence flow* sert de

passage pour le token d'un objet à l'autre (au sein d'une même *pool*). Un *sequence flow* peut franchir les limites d'une *lane* mais pas d'une *pool*.

Les différents symboles graphiques possibles pour un *sequence flow* sont les suivants :



Sequence flow basique



Séquence flow conditionné (dans la cas où ce *sequence flow* est relié à un *gateway*, le losange n'existe pas puisque le *gateway* stipulera lui-même l'aspect conditionnel)



Sequence flow correspondant à une option par défaut (par exemple relié à un *gateway* de type XOR)

3.3.7.2 Message flow






























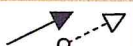





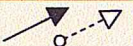

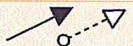









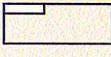





Ce connecteur est utilisé pour représenter l'échange de message entre deux entités susceptibles d'envoyer ou de recevoir un message. Une règle important régit l'utilisation de ce type de *flow* : Un *message flow* doit relier deux *pools* ou deux entités situées dans deux *pools* différentes. Un *message flow* ne peut pas connecter deux objets d'une même *pool*.

3.3.7.2 Association

Ce connecteur est utilisé dans divers cas :

- Pour associer un élément de documentation à une entité
- Pour représenter des données en tant qu'entrées ou sorties d'une activité ou comme élément du flow (ce que représenterait réellement le token).
- Pour associer une compensation à une activité

En outre, une association peut être directionnelle ou non. Les différentes connexions possibles entre les divers composants de modélisation (*sequence flows* et *message flows*) peuvent être représentées à l'aide du tableau récapitulatif suivant :

| Dest. Source |  |  |  |  |  |  |  |
|---|---|---|---|--|---|---|---|
|  | |  |  |  |  |  | |
|  | |  |  |  |  |  | |
|  |  |  | |  |  | |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  | |  |  |  |  |  | |
|  |  |  | |  |  | |  |

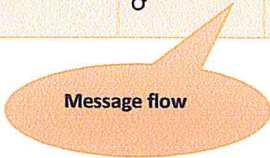
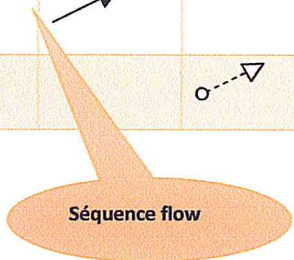

Message flow

Séquence flow

Tableau 3 : La récapitulatif des connexions


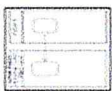
































| BPMN 2.0 | | SoaML beta2 | | BPMN 2.0 | | SoaML beta2 | | |
|--|---|---|---|---|--|---|---|--|
| Element | Icon | Element | Icon | Element | Icon | Element | Icon | |
| Definitions | (complete model) | Model |  <Model> | Collaboration |  | Services Architecture |  <Collaboration> | |
| Process |  | Participant |  <Class> | Collaboration Participant |  | Participant Part |  <Property> | |
| ServiceTask (provider) +MessageFlow (sourceRef=Task, targetRef=ServiceTask) +Task (consumer) |  | MessageType |  <Class> | MessageFlow (sourceRef=Task, targetRef=ServiceTask) |  | CollaborationUse Dependency consumer provider |  <Collaboration Use>  <Dependency> | |
| | | Interface |  <Interface> | ServiceTask (provider) |  | Service (of Participant) |  <Port> | |
| | | Operation |  <Operation> | Task (consumer) |  | Request (of Participant) |  <Port> | |
| | | Parameter In/Out |  <Parameter> | | | | | |
| | | ServiceContract Provider Consumer |  <Collaboration>  <Property> provider  <Property> consumer | | | | | |
| When Interfaces, Operations, Parameters and Messages are present in the BPMN2 model | | | | | | | | |
| MessageFlow (sourceRef=Task, targetRef=ServiceTask) |  | ServiceContract Provider Consumer |  <Collaboration>  <Property> provider  <Property> consumer | Interface |  Interface | Interface |  <Interface> | |
| | | | | Operation |  Operation | Operation |  <Operation> | |
| Message |  | MessageType |  <Class> | MessageRef In/Out |  | Parameter In/Out |  <Parameter> | |

Tableau 4 : Les composants globaux BPMN.

Bibliographie

Bibliographie



- (Allen, 2007) : Allen, P. (2007, February). The service oriented process. Cbdi Journal,,
Vo, February.
- (Arsanjani, Ghosh, Allam, & Abdollah, 2008) : Arsanjani, A., Ghosh, S., Allam , A., & Abdollah, T.
(2008).« SOMA: a method for developing service-oriented solutions ». IBM System
Journal, vol. 3, no. 47, 377-396.
- (Chappell , 2004) : Chappell , D. (2004). Enterprise service bus. Beijing;Cambridge. O'Reilly:
ISBN.
- (Dickson W. , 1966) : Dickson, W. (1966). An analysis of vendor selection systems and
decisions. Journal of Purchasing, 2, 5-20.
- (Jones & Morris, 2005) : Jones , S., & Morris, M. (2005). A methodology for service
architectures. Capgemini UK plc. Capgemini UK plc.
- (Silver, Pyke, & Petersson , 1998) : Silver, E., Pyke, D., & Petersson , D. (1998).
Inventorymanagement and Production Planning and Scheduling. New York: John Wiley
& Sons.
- (Vescoukis, Doulamis , &Karagiorgou, 2012) : Vescoukis, V., Doulamis , N., & Karagiorgou, S.
(2012). "A service oriented architecture for decision support systems in environmental
crisis management," *Future Generation Computer Systems*.
- (Adrien, 2008) : Adrien, L. (2008, Octobre). *Bus De Service -ESB-. Nouvelle Technologie Pour
l'intégration*. Consulté le octobre 2011, sur <http://petals.ow2.org/docs/Livre-Blanc-ESB-fr.pdf>
- (Amsden , 2010) : Amsden , J. (2010). Modeling with soaml, the Service-Oriented Architecture
modelinglanguage.*IBM*.Récupérésur

Bibliographie

- <http://www.ibm.com/developerworks/rational/library/09/modelingwithsoaml-1/index.html>
- Berre, A., Roman, D., Elvesaeter, B., & Carrez, C. (2010, 11 29). *Service oriented architecture Modeling Language (soaml)*. Récupéré sur Tutorial at MOPAS: http://www.iaria.org/conferences2010/filesmopas10/MOPAS_2010
- (Birol , 2008) : Birol , B. (2008). From The Business Motivation Model (BMM) To Service Oriented Architecture (SOA). *Journal of Object Technology*, Vol 7, no 8.
- (Börner & Goeken, 2009) : Börner, R., & Goeken, M. (2009). Methods for Service Identification: A Criteria-Based. Literature Review. *15th Americas Conference on Information Systems (AMCIS)*, (p. Paper 162). San Fran-cisco, California.
- Boukadi, & Khouloud. (2009). *Coopération Interentreprises à La Demande : Une Approche Flexible à Base De Services Adaptables*. France: SAIT-ETIENNE Ecole Nationale Supérieur des Mines. Consulté le Février 2012
- (Boukadi & Khouloud, 2009) : Boukadi, & Khouloud. (2009). *Coopération Interentreprises à La Demande : Une Approche Flexible à Base De Services Adaptables*. France: SAIT-ETIENNE Ecole Nationale Supérieur des Mines. Consulté le Février 2012
- (F. Boumahdi and R. Chalal, 2013) : Boumahdi, F., & Chalal , R. (2013). "soadsspm: A new Service-Oriented Architecture of the decision support system for the Project Management. *International Conference on Control, Engineering & Information Technology (CEIT'13)*. Sousse Tunisie.
- (Bourey, 2007) : Bourey, J.-P. (2007). *Model Driven Interoperability and Service-Oriented Architecture*. Séminaireimszurich.
- [Casanave , 2012] : Casanave , C. (2012). *Enterprise Service Oriented Architecture Using the OMG soaml Standard, a Model Driven Solutions*. Modeldriven.org.
- (CBDI-SAE, 2007) : CBDI-SAE. (2007). Meta Model for SOA Version 2. Consulté le juin 4, 2010, sur http://www.cbdiforum.com/public/meta_model_v2.php
- (Chaari, Biennier, Favrel , & Benamar, 2007) : Chaari, S., Biennier, F., Favrel , J., & Benamar, C. (2007). Towards a service-oriented enterprise based on business components identification. *Enterprise Interoperability*, 495–506.
- (Daehene, 2009) : Daehene, P. (2009). *Architecture SOA*. Haute école de gestion de Genève Informatique, degestion. Consulté le, Janvier 2012, sur <http://campus.hesge.ch/Daehne/courscti2009/Presentation/moduleiva.pdf>
- (De Castro, Marcos, & Vara , 2011) : De Castro, V., Marcos, E., & Vara , J. (2011). Applying CIM-to-PIM model transformations for the service-oriented development of information systems. *Information and Software Technology*, 53(1), 87–105.
- (Dickson W. , 1966) : Dickson, W. (1966). W. G. Dickson, An analysis of vendor selection systems and decisions. *Journal of Purchasing*, 5-20.

Bibliographie

- (Edmond & Ter Hofstede , 2002) : Edmond , D., & Ter Hofstede , A. (2002). What's in a service ?Distrib. Parallel Databases.
- (Elr T. , 2008) : Elr, T. (2008). *SOA : principles of service design*. Prentice Hall.
- (Elvesæter , Berre , & Sadovykh , 2011) : Elvesæter , B., Berre , A., & Sadovykh , A. (2011). Specifying services using the service oriented architecture modeling language (soaml). *The 1st International Conference on Cloud Computing and Services Science CLOSER*. The Netherlands.
- (Erl T. , 2004) : Erl, T. (2004). *Service Oriented architecture : A field Guide to Integrating XML and Web Services*. Prentice Hall.
- (Erl, 2010) : Erl, T. (2010). *Service-Oriented Architecture : Concepts, Technology, and Design : 10ème édition (The Prentice Hall Service-Oriented Computing Series from Thomas Erl)*. Prentice Hall.
- (Erl & Upper Saddle River, 2005) : Erl, T., & Upper Saddle River. (2005). *Service-oriented architecture : concepts, technology, and design*. ISBN: Prentice Hall PTR.
- (Erradi, Anand, & Kulkarni , 2006) : Erradi, A., Anand, S., & Kulkarni , N. (2006). *Erradi, A., Anand, S., & Kulkarni, N. SOAF: An Architectural Framework for Service Definition and Real ization. Services Computing, 2006. SCC '06*. IEEE International Conference on.
- (Godart , 2009) : Godart , C. (2009). *Les processus métiers : concepts, modèles et systèmes*. Paris: Hermès science.
- (Group,2010): Group, O. (2010,décembre 4).*Ontologies for SOA*. Récupéré sur <http://www.opengroup.org/projects/soa-ontology>
- (Ivanyukovich & Gangadharan , 2005) : Ivanyukovich, A., Gangadharan, G., & V. D'Andrea, M. (2005). Towards a service-oriented development methodology. *Journal of integrated design & process science*, 53-62.
- (Kim & Yun, 2006) :Kim, Y., & Yun, X. (2006). An Approach to Modeling Service-Oriented Development Process. *IEEE International Conference on Services Computing*, (pp. 273–276).
- (Mittal, 2006):Mittal, K. (2006). Build your SOA, Part 3: The Service-Oriented Unified Process. *IBM developer Works*, URL : www.ibm.com/developerworks/library/ws-soa-method1.html.
- (OMG, 2011) : OMG. (2011). Business Process Model and Notation (BPMN) Version 2.0. *OMG*, <http://www.omg.org/spec/BPMN/2.0>.
- (OMG, 2012) :OMG. (2012). *Service oriented architecture Modeling Version 1.0.1*.
- (OMG,2014) :OMG. (2014)*Decision Model and Notation Beta 1*. [Http://www.omg.org/spec/DMN/1.0/Beta1/PDF](http://www.omg.org/spec/DMN/1.0/Beta1/PDF).

Bibliographie

- (Papazoglou & Heuvel , 2006) : Papazoglou , M., & Van Den Heuvel , W. (2006). Service-oriented design and development. *International Journal of Web Engineering and Technology*, p. 412–442.
- (Pascal Bou Nassar, 2012) :Pascal Bou Nassar. (2012). *Thèse*. Récupéré sur : <http://theses.insa-lyon.fr/publication/2012ISAL0102/these.pdf>
- (Rahmani, Rafe, Sedighian, & Abbaspo, 2006) :Rahmani, A., Rafe, V., Sedighian, S., & Abbaspo. (2006). An MDA-Based Modeling and Design of Service Oriented Architecture. *Computational Science*, 578–585.
- (Silver, Pyke, & Petersson , 1998) :Silver, E., Pyke, D., & Petersson , R. (1998). *Inventorymanagement and Production Planning and Scheduling*.New York: John Wiley & Sons.
- (Sodki , 2008) :Sodki , C. (2008). *Interconnexion des processus Interentreprises : une approche orientée services*. Lyon: EDIIS.
- (Thai Tri, 2005) :Thai Tri, H. (2005). "*Architecture Des Systèmes d'information d'entreprise Et Architecture Orientée Service*". Institut de la Francophonie pou l'informatque. Consulté le Octobre 2011, sur www.ifi.auf.org/site_data/rapports/tpe.../tipe-thai_tri_hung.pdf
- (Touzi, 2007) :Touzi, J. (2007). *Aide à La Conception Des Systèmes d'information Collaboratif Support De l'interopérabilité Des Entreprises*.
- (Vescoukis, Doulamis , &Karagiorgou, 2012) :Vescoukis, V., Doulamis, N., & Karagiorgou, S. (2012). A service oriented architecture for decision support systems in environmental crisis management. *Future Generation Computer Systems*, 28(3), 593–604.
- (Yann , 2007) :Yann , L. (2007). *Urbanisation & Intégration de Systèmes «THINK SERVICE »Version 1.2*.
- (Zimmermann, 2009) :Zimmermann, O. (2009). An architectural decision modeling framework for service-oriented architecture design. *University of Stuttgart ISBN 978-3-86624-438-2*, 1-214.
- (Zimmermann, Pal , & Clive , 2004) :Zimmermann, O., Pal , K., & Clive , G. (2004). *Elements of Service-Oriented Analysis and Design : An interdisciplinary modeling approach for SOA projects*.Consulté le janvier 18, 2009, sur <http://www.ibm.com/developerworks/webservices/library/ws-soad1>

