

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITÉ ABDERRAHMANE MIRA DE BEJAIA

FACULTE DES SCIENCES EXACTES
DÉPARTEMENT D'INFORMATIQUE
ÉCOLE DOCTORALE RÉSEAUX ET SYSTÈMES DISTRIBUÉS



Mémoire de Magistère En Informatique

Option : Réseaux et Systèmes Distribués

Thème

**Analyse et Modélisation Orientée Services d'un
Système de Maintenance Industrielle**
« Cas d'une unité de conditionnement – *Cevital* »

Présenté par :

TOULOUM Karim

Devant le jury composé de :

Président :	A. DAHMANI	Professeur, Université de Béjaïa.
Rapporteur :	M. KERKAR	Professeur, Université de Béjaïa.
Examineur :	B. BELATTAR	Maitre de Conférences, Université de Batna.
Examineur :	A. MELLIT	Maitre de Conférences, Université de Jijel.
Invité :	D. IDOUGHI	Docteur, Université de Béjaïa.

Promotion 2006 - 2007

Remerciements

Le travail présenté dans ce mémoire rentre dans le cadre d'un Mémoire de Magistère réalisée à l'Université A. Mira de Bejaia, Algérie. La partie pratique de ce travail a été effectuée au niveau de l'unité de conditionnement appartenant au groupe agro-alimentaire *Cevital* situé dans la ville de Bejaia.

Je tiens, tout d'abord à remercier le directeur de mémoire, le Professeur Moussa KERKAR de l'université de Bejaia, pour tout son soutien, ses orientations et ses encouragements tout au long de ce mémoire.

Mes remerciements les plus vifs vont aussi à mon promoteur Dr. IDOUGHI Djilali, enseignant-chercheur au département d'Informatique pour sa confiance exprimée envers mes capacités à réaliser ce travail, ses conseils, sa disponibilité et surtout ses remarques pertinentes qui m'ont beaucoup aider durant tout ce mémoire.

Je remercie également le personnel de l'organisme d'accueil de *Cevital*, qui a eu l'amabilité de m'accorder le stage pratique au sein de leur entreprise. En particulier, je cite le DRH, chef du bureau de méthodes ainsi que toute l'équipe des méthodistes au niveau de l'unité de conditionnement des huiles.

Enfin, je tiens à remercier les membres du jury, qui ont bien voulu accepter d'examiner et d'évaluer mon travail.

Dédicaces

Je dédie d'abord ce travail à mes chères parents que je remercie pour tout leurs patience et leurs soutient moral, sans oublier mes frères Samir, Yacine et ma sœur Samira et son charment fils Nacime.

Enfin, je ne saurais oublier tous ceux qui m'on aider de près ou de loin, à qui je dédie ce travail, particulièrement à mon chère ami FARAH Zoubeir, mes amis B. Bachir et M. Abdallah et à tous ceux qui ont contribué à la réussite de ce travail.

TABLE DES MATIERES

Remerciements

Dédicaces

Table des matières

Liste des figures

Liste des tableaux

INTRODUCTION GENERALE 1

CHAPITRE I : LA MAINTENANCE INDUSTRIELLE

I. Introduction 4

II. Aspects caractéristiques de la maintenance industrielle..... 5

II.1. Service de maintenance 5

II.2. Approches de la maintenance 6

II.3. Formes de la maintenance industrielles 9

III. Structure et architectures des systèmes de maintenance 11

III.1. Architecture d'un système de maintenance 12

III.2. Typologies des systèmes de maintenance 12

III.3. Evolutions majeures dans les systèmes de maintenance 15

IV. Synthèse et nouvelles motivations 18

IV.1. Vers une externalisation de la maintenance 18

IV.2. Problématique des silos applicatifs 19

IV.3. Interfaçage des applications 20

IV.4. Complexité liée aux informations échangées et le matériel de communication 21

IV.5. Les aspects liés aux relations entre les systèmes de maintenance..... 21

IV.6. Les aspects liés aux interactions Homme-Machine 22

V. Conclusion..... 23

CHAPITRE II : L'ARCHITECTURE ORIENTEE SERVICES (SOA) ET LES SERVICES WEB

I. Introduction 25

II. Eléments et concepts clés de SOA 26

II.1. Concept de service..... 26

II.2. Principes de bas des SOA 27

II.3. Caractéristiques générales des services 30

II.4. Infrastructure SOA et ses composants 32

III. Les services Web 35

III.1. Concept et caractéristiques des services Web 36

III.2. Standards des services Web	37
III.3. Composition et coordination des services Web.....	38
IV. Aspects méthodologiques liés à l'approche orientée services	41
IV.1. Analyse et modélisation orientées services	42
IV.2. Modélisation métier	42
V. Conclusion.....	43
 CHAPITRE III : VERS UNE APPROCHE D'ANALYSE ET DE MODELISATION ORIENTEE SERVICES POUR LES SYSTEMES DE MAINTENANCE	
I. Introduction	44
II. Méthode 1 : Approche de spécification orientée services d'IHM pour la supervision industrielle.....	45
II.1. Phase 1 : Etude de l'organisation globale de supervision et analyse métier.....	46
II.2. Phase 2 : Analyse et expression des besoins.....	48
II.3. Phase 3 : Spécification et conception de l'IHM.....	50
II.4. Conclusion sur la méthode 1.....	52
III. Méthode 2 : SOMA (<i>Services Oriented Modeling and Architecture</i>)	53
III.1. Phase 1 : Identification	54
III.2. Phase 2 : Spécification	56
III.3. Phase 3 : Réalisation.....	59
III.4. Conclusion sur la méthode 2	59
IV. Synthèse des approches présentées.....	60
IV.1. Les approches et l'analyse du domaine métier.....	60
IV.2. Les approches et le cycle de développement.....	62
IV.3. Identification des services	64
IV.4. Analyse de l'existant	63
V. Vers une approche d'analyse et de modélisation orientée services d'un système de maintenance industrielle	66
V.1. Phase 1 : Etude de l'organisation de maintenance et analyse métier	61
V.2. Phase 2 : Analyse et expression des besoins	68
V.3. Phase 3 : Spécification de l'architecture orientée services du système de maintenance .	69
VI. Conclusion	78
 CHAPITRE IV : APPLICATION ET MISE EN ŒUVRE DE L'APPROCHE PROPOSEE SUR UNE ETUDE DE CAS RELATIVE A LA MAINTENANCE INDUSTRIELLE (<i>Unité de conditionnement - Cevital</i>)	
I. Introduction et présentation du cas d'étude	80
II. Phase 1 : Etude de l'organisation de maintenance et analyse métier.....	80

II.1. Analyse métier	80
II.2. Analyse des utilisateurs et leurs tâches	87
II.3. Conclusion sur la phase 1	91
III. Phase 2 : Analyse et expression des besoins	91
III.1. Identification et catégorisation des services métier	91
III.2. Identification des besoins des acteurs métier	94
III.3. Analyse du système existant	101
III.4. Conclusion sur la phase 2	102
IV. Phase 3 : Spécification de l'architecture orientée services	103
IV.1. Spécification des processus métier	103
IV.2. Spécification de l'interface utilisateur	109
IV.3. Spécification des services et composants	116
IV.4. Conclusion sur la phase 3.....	127
V. Conclusion	127
 CHAPITRE V : CONCLUSION GENERALE ET PERSPECTIVES DE RECHERCHE	
I. Conclusion générale	129
II. Perspectives de recherche	130
REFERENCES BIBLIOGRAPHIQUES	132
 ANNEXES	
A : Description sommaire des couches de base des services Web	136
B : Description sommaire du langage BPEL	139
C : Description sommaire de la notation BPMN.....	141
D : Description sommaire des approches EA, OOAD et BPM.....	144
E : Description sommaire des formalismes CTT et SADT	146
F : Description sommaire de WSRP.....	148
G: Tableaux représentant les différents acteurs métier de la maintenance avec leurs catégories, besoins, tâches et leurs activités	150
H : Extraits des fichiers WSDL relatifs au service Web <i>Réservation des pièces de rechange</i>	152
I : Procédure globale pour l'ajout d'un nouveau service métier dédié à la maintenance industrielle dans l'architecture SOA	154
J : Ensemble de prototypes du portail Web dédié à la maintenance.....	156
K : Description sommaire des couches de base des services Web	159
GLOSSAIRE	160

Liste des figures

Figure N°	Intitulé	Page N°
Chapitre I		
1.1	Relations entre le service de maintenance avec les autres services d'une organisation complexe	5
1.2	Approche Centralisée de la maintenance	6
1.3	Approche Décentralisée de la maintenance	7

1.4	Approche Mixte de la maintenance	7
1.5	Approche de la maintenance sous-traitée	8
1.6	Approche Distribuée de la maintenance [Kaffel, 01]	9
1.7	Les Formes de la maintenance industrielle [Auberville, 04]	10
1.8	Structure d'un système de gestion de la maintenance (d'après [Kaffel, 01])	11
1.9	Architecture d'un système de maintenance (d'après [Rasovska <i>et al.</i> , 07])	12
1.10	Principales fonctionnalités d'un système de GMAO [Frédéric, 03]	13
1.11	Architecture globale d'un système GMAO (d'après [Frédéric, 03])	13
1.12	Un exemple d'architecture d'un système SCADA [Barre et Fonash, 04]	14
1.13	Architecture d'un système de télémaintenance [Rasovska <i>et al.</i> , 07]	15
1.14	Architecture d'un système de e-maintenance [Rasovska <i>et al.</i> , 07]	16
1.15	La e-Plateforme de Proteus [Rebeuf <i>et al.</i> , 05]	17
1.16	Fonctionnement en silo. <i>Inspiré de</i> [Fournier-Morel <i>et al.</i> , 06]	19
1.17	L'interfaçage des applications. <i>Inspiré de</i> [Frédéric, 03]	20
1.18	Outils EAI (<i>Enterprise Application Integration</i>) [Fournier-Morel <i>et al.</i> , 06]	20
1.19	Intensité de la relation entre les systèmes [Rasovska <i>et al.</i> , 07]	21
Chapitre 2		
2.1	Service : boucle d'interactions entre demandeur et fournisseur de service [Salatgé, 06]	26
2.2	La description d'un contrat de service	29
2.3	Principe d'Encapsulation de l'implémentation	30
2.4	Le triangle de collaborations dans SOA [Endrei <i>et al.</i> , 04]	31
2.5	Typologie et la granularité des Services [Bonnet, 05 b]	32
2.6	Représentation du système par composants de services (d'après [Raymond, 07])	33
2.7	Le principe des applications composites	34
2.8	Les interactions entre l'application composite et les services	34
2.9	Exemple d'exécution d'un processus métier par le moteur BPM	35
2.10	Les composants d'une plate-forme SOA (d'après [Fournier-Morel <i>et al.</i> , 06])	36
2.11	La pile des Spécification des services Web [Fournier-Morel <i>et al.</i> , 06]	37
2.12	L'orchestration de Services [Porter-Roth, 05]	39
2.13	La chorégraphie de Services [Porter-Roth, 05]	40
2.14	L'Approche SOA : liaison entre vue métier et vue technique [Fournier-Morel <i>et al.</i> , 06]	42
Chapitre 3		
3.1	Les trois phases de l'approche [Idoughi, 08]	45
3.2	Le processus d'identification des services métier [Idoughi, 08]	46
3.3	Description globale d'un scénario d'un processus métier [Idoughi, 08]	46
3.4	Exemple d'usage métier des services Web [Idoughi, 08]	47
3.5	Vue métier de la supervision dans le cadre de SOA [Idoughi, 08]	47
3.6	Décomposition hiérarchique d'un système homme-machine global [Idoughi, 08]	48
3.7	Modèle de cas d'utilisation de services métier [Idoughi, 08]	49
3.8	Différents services Web d'un processus métier [Idoughi, 08]	49
3.9	Processus d'expression des besoins à l'aide de services Web [Idoughi, 08]	50
3.10	Les niveaux de spécification de services Web orientés présentation [Idoughi, 08]	51
3.11	Le processus de maquettage et prototypage de l'interface utilisateur [Idoughi, 08]	51
3.12	SOMA par rapport aux approches OOAD, EA et BPM [Zimmerman <i>et al.</i> , 04]	53
3.13	Démarche SOMA pour l'analyse et modélisation orientée services [Arsanjani, 04]	54
3.14	Décomposition du Domaine (d'après [Ganci <i>et al.</i> , 06])	55
3.15	Modèle de services-objectifs (d'après [Endrei <i>et al.</i> , 04])	56
3.16	Les composants d'un sous-système	57
3.17	Les niveaux SOA selon IBM [Arsanjani, 04]	59

3.18	Synthèse des approches vis-à-vis de l'analyse et de la conception	63
3.19	Le filtrage des services dans SOMA (d'après [Dennerly, 07])	64
3.20	Les deux approches et l'analyse de l'existant	65
3.21	Vue globale de la démarche proposée	66
3.22	Les étapes de la première phase	67
3.23	Analyse des utilisateurs et leurs tâches	69
3.24	Analyse et expression des besoins	69
3.25	Identification et catégorisation des services métier	70
3.26	Identification des besoins métier	70
3.27	Conception de l'architecture orientée services	71
3.28	Les éléments impliqués dans un processus métier. Inspiré de [Idoughi, 08]	72
3.29	Exemple de spécification d'interactions dans un processus métier [Idoughi, 08]	73
3.30	Conception d'un processus métier	74
3.31	Position services Web orientés présentation dans SOA. Inspiré de [Idoughi, 08]	75
3.32	Maquettage/prototypage de l'interface utilisateur	75
3.33	Spécification de services	77
Chapitre 4		
4.1	Vue globale du complexe <i>Cevital</i>	81
4.2	Structure générale de l'unité de conditionnement	81
4.3	Système de Gestion de maintenance existant	82
4.4	Relations entre le service de maintenance et les autres services	82
4.5	Modèle de décomposition du domaine métier « <i>maintenance industrielle</i> »	83
4.6	Modèle d'interactions du scénario métier « <i>gestion de la maintenance corrective</i> »	86
4.7	Identification des cas d'utilisation métier	87
4.8	Typologie des acteurs intervenant dans le processus métier de la maintenance	87
4.9	Modèle des cas d'utilisation métier relatif au domaine « <i>opérations de maintenance</i> »	88
4.10	Modèle des cas d'utilisation métier relatif au scénario « <i>maintenance conditionnelle</i> »	89
4.11	Modèle d'activités SADT correspondant au sous-système « <i>maintenance conditionnelle</i> »	90
4.12	Extrait du modèle CTT du sous-système « <i>maintenance conditionnelle</i> »	90
4.13	Catégorisation des services métier	93
4.14	Identification des services CRUD (<i>Create, Read, Update, Delete</i>)	93
4.15	Exemple de composition des services à base de la catégorisation	94
4.16	Description du cas d'utilisation « <i>préparation des gammes</i> »	95
4.17	Expression en terme de services Web du cas d'utilisation : « <i>préparation des gammes</i> »	95
4.18	Description du cas d'utilisation « <i>réservation des ressources matérielles</i> »	96
4.19	Expression en terme de services Web du cas d'utilisation : « <i>réservation des ressources matérielles</i> »	96
4.20	Extrait du Catalogue des Services Métier identifiés	97
4.21	Scénario de maintenance relatif à la mobilité des acteurs humains	98
4.22	Expression en terme de services Web du scénario de mobilité : « <i>gestion des interventions</i> »	99
4.23	Expression d'une tâche coopérative « <i>réservation des ressources matérielles</i> » à l'aide de services	100
4.24	Expression en terme de services Web composants la tâche coopérative relative au scénario métier :	101
4.25	Décomposition des deux systèmes : ERP et SCADA	102
4.26	Expression des interactions entre les partenaires humains et automatiques avec le processus métier	104
4.27	Extrait du fichier relatif à la spécification des liens partenaires dans le processus métier « <i>traitement</i> »	105
4.28	Exemple de spécification des interactions lors de la génération de la DI à travers des variables et	106
4.29	Modélisation du processus métier « <i>traitement de la DI</i> » par la notation BPMN	107
4.30	Extrait du code BPEL relatif au processus métier « <i>traitement de la DI</i> »	109
4.31	Architecture simplifiée de la couche présentation basée sur les services WSRP	111
4.32	Illustration du scénario <i>réservation des ressources</i> sous l'angle d'IHM	112

4.33	Maquette d'authentification et accès de l'opérateur aux différents services dédiés à la maintenance	113
4.34	Maquette relative à la réservation des ressources matérielles	114
4.35	Maquettes relatives à la réservation des ressources matérielles et l'édition de l'OT	114
4.36	Exemples de prototypes d'interface relatifs au scénario de réservation de ressources	115
4.37	Modèle de composants relatif au sous-système <i>Ordre de travail</i>	117
4.38	Modèle de classes d'objets métier relatif au système de maintenance	119
4.39	Modélisation du « <i>TDE Alarme</i> »	120
4.40	Spécification du <i>TDE_Alarne</i> à l'aide de <i>XML Schema (XSD)</i>	121
4.41	Extrait du modèle d'interfaces des services métier	122
4.42	Message SOAP définissant les contraintes QoS dans un contrat de service	123
4.43	Modèle conceptuel de classes des composants	124
4.44	Extrait du modèle de services	125
4.45	Architecture Orientée Services pour un système de gestion de la Maintenance Industrielle	126
Annexes		
A.1	Structure générale d'un message SOAP [Kadima et Monfort, 03]	136
A.2	Structure d'un document WSDL [Kadima et Monfort, 03]	137
A.3	Principe de fonctionnement de l'annuaire UDDI [Chauvet, 02]	138
B.1	Exemple de la balise <process>	139
B.2	Exemple de la balise <import>	139
B.3	Exemple de la balise <partnerLinks>	139
B.4	Exemple de la balise <variables>	140
B.5	Exemple de la balise <sequence>	140
B.6	Exemple de la balise <receive>	140
B.7	Exemple de la balise <reply>	140
B.8	Exemple de la balise <invoke>	140
B.9	Exemple de la balise <forEach>	140
C.1	Types de Branchement	143
C.2	Exemple de modélisation du processus « Achat de produits »	143
D.1	Vues de l'Architecture d'Entreprise (EA)	144
D.2	Position des approches OOAD, EA et BPM	145
E.1	Les Actigrammes et les Datagrammes	147
F.1	Couche WSRP par rapport aux portails et les services Web	148
K.1	Vue synthétique de la démarche orientée services proposée	159

Liste des tableaux

Tableau N°	Intitulé	Page N°
3.1	Les approches et l'analyse du domaine métier	62
4.1	Description des processus métier identifiés	84
4.2	Liste représentative des services métier (candidats) identifiés	92
4.3	Extrait de la description des services métier dédiés à la mobilité des acteurs humains	99
4.4	Translation de quelques objets BPMN vers les éléments du langage BPEL	108
4.5	Exemple de contraintes non-fonctionnelles pour le contrat de service	123
C.1	Les objets de la norme BPMN	141
C.2	Types de Tâches	142
C.3	Les objets de Connexion	142
C.4	Types d'évènements	142
E.1	Les graphiques du formalisme CTT	146

INTRODUCTION GENERALE

Dans un contexte économique en constante évolution, la concurrence oblige les entreprises à être et demeurer compétitives. De ce fait, les industriels doivent améliorer le rendement de leurs installations de production pour répondre mieux aux besoins de leurs clients. L'une des voies permettant d'atteindre cet objectif, consiste à augmenter la productivité tout en garantissant la bonne qualité des produits. Par conséquent, la maintenance industrielle s'avère nécessaire pour rentabiliser toute la chaîne de production, en prolongeant la période d'exploitation et en minimisant le temps d'arrêts dû aux défaillances des équipements industriels.

La fonction de maintenance est devenue aujourd'hui un centre de préoccupation des entreprises pour deux raisons essentielles : (1) le coût de la maintenance représente une partie importante du coût total de la production, (2) un défaut d'un équipement peut avoir un impact considérable sur la qualité du produit et sur la disponibilité de l'équipement.

La maintenance représente un défi industriel impliquant la remise en cause des structures figées et la promotion de méthodes nouvelles adaptées à la nature moderne des matériels. Les avancées technologiques dans le domaine de communication ont favorisé l'apparition de nouveaux types et formes de maintenance : télémaintenance distribuée, E-Maintenance, etc.

Par ailleurs, la maintenance industrielle reste une fonction très complexe. Par conséquent, plusieurs systèmes informatiques d'aide, d'assistance et autres outils de gestion de ressources sont devenus indispensables et doivent cohabiter. Néanmoins, tous ces systèmes restent plus ou moins complémentaires, non interopérables, souvent hétérogènes, redondant et parfois incohérents.

D'autres parts, les systèmes d'information (SI) ont connus plusieurs évolutions notamment au niveau de leurs architectures. Les premiers SI connus sous le nom « mainframe » étaient basés sur un serveur unique et central appelé mainframe, où les clients, dotés de machines à faible puissance, communiquent avec ce serveur pour recevoir des informations ou effectuer les traitements appropriés.

Ensuite, les architectures Client/serveur ont vu le jour vers le début des années 90. Ces architectures introduisent des applications plus légères, et les clients ont la possibilité d'interagir avec plusieurs Serveurs.

A la fin des années 90, les applications Web ont commencé à s'émerger dans les architectures des SI d'Entreprise. Ces nouvelles architectures sont venues pour palier les problèmes de déploiement et de gestion des grands parcs de PC connus dans les applications client/serveur. Ainsi, on peut accéder à ce type d'applications à l'aide d'un simple navigateur Web. Cela a permis d'intégrer facilement les clients et partenaires de l'entreprise dans les processus métier et ainsi participer au développement du commerce électronique (e-business) entre autres.

Malgré toutes ces évolutions, les entreprises restent très exigeantes au niveau du développement et de l'actualisation de leur SI global dont le système de maintenance industrielle fait partie, notamment sur les points suivants : (1) Demande croissante en termes d'agilité⁽¹⁾: en effet, les SI basés sur des blocs applicatifs opaques deviennent trop rigides pour les modifications ou encore pour leur modernisation. Ainsi, les coûts et la durée induits par les mises-à-jour deviennent plus importants. De ce constat, l'architecture du SI doit être construite sur la base de composants plus faciles à faire évoluer et à réutiliser ; (2) Intégration des acteurs externes : le SI devant s'ouvrir aux consommateurs et partenaires de l'entreprise, exige des niveaux très élevés en termes de sécurité et de performances.

Face à cette situation, les acteurs du domaine Génie Logiciel ont pensé à une refonte complète au niveau de la modélisation et la conception des nouveaux SI, en tenant compte non seulement des exigences citée ci-dessus, mais surtout de l'arrivée des Services Web qui sont devenus les technologies de facto pour le développement des applications E-business.

Dans cette optique, les Architectures Orientées Services ou SOA (Services Oriented Architecture) sont nées. SOA propose une nouvelle façon de concevoir les systèmes d'information. Sa vocation principale est de décomposer les traitements sous forme de services rattachés à des catégories, et représentant chacune un objet métier. Ces services communiquent avec les clients (consommateurs de services) ou avec d'autres services via un bus d'intégration pouvant être orchestrés par un moteur d'orchestration de services.

Les SOA ont l'avantage non seulement d'utiliser les technologies basées sur les Services Web, mais aussi peuvent être basées sur des technologies déjà approuvées notamment les EJB (*Entreprise Java Beans*), CORBA (*Common Object Request Broker Architecture*) ou encore les composants DCOM (*Distributed Component Object Model*).

Notre projet s'articule essentiellement sur cette nouvelle démarche d'analyse et de modélisation basée sur la notion de services, qui sera appliquée dans le domaine de maintenance industrielle.

Notre cas d'étude est consacré aux systèmes dédiés à la gestion de la maintenance industrielle où peu de travaux sont réalisés dans ce secteur si important pour la gestion efficace de l'ensemble des activités de l'entreprise.

Compte tenu de la complexité de la fonction maintenance industrielle due aux évolutions des technologies liées à la maintenance et à la supervision des processus industriels d'une part, et d'autre part l'aspect d'hétérogénéité des sous systèmes qui interagissent dans la maintenance industrielle, il est donc nécessaire d'introduire une nouvelle approche d'analyse et de modélisation basée sur SOA pour les systèmes dédiés à la gestion de la maintenance industrielle.

⁽¹⁾ **Agilité** : terme qui désigne la capacité du SI à supporter les évolutions rapides du domaine métier de l'entreprise [Fournier-Morel et al, 06].

Ce mémoire est organisé suivant cinq chapitres :

Dans le chapitre 1, nous présentons une introduction sommaire au domaine de la maintenance industrielle à travers la définition des différents aspects et méthodes impliquées dans la fonction de la maintenance industrielle. En outre, nous dressons un état de l'art sur les architectures des systèmes de gestion de la maintenance et les nouvelles motivations relatives à la structure et aux évolutions en termes de technologies d'information et de communication de ces systèmes.

Le chapitre 2 est consacré dans sa première partie à la présentation de l'approche orientée services à travers les concepts, les éléments et les principes fondamentales de l'approche. La seconde partie du chapitre, introduit les éléments clé liés aux services Web, notamment en termes de protocoles de communication, langages de spécification, etc.

Dans le chapitre 3, nous exposons deux approches représentatives qui s'inscrivent dans un cadre méthodologique de l'approche orientée services. L'étude de ces approches servira de base pour la proposition d'une nouvelle démarche destinée à l'analyse et la modélisation orientée services d'un système de maintenance industrielle. Par conséquent, nous décrivons les phases et les étapes essentielles impliquées dans la démarche proposée.

Le chapitre 4 montre un exemple de mise en œuvre et de validation de la démarche proposée lors du chapitre 3 sur une étude de cas réelle relative à la maintenance industrielle dans une entreprise de production des huiles végétales. Lors de cette étude, nous soulignons l'apport de l'approche orientée services dans le développement des systèmes de gestion de la maintenance industrielle à travers des scénarios typiques issus du domaine de la maintenance.

Enfin, dans le chapitre 5, nous concluons notre travail par la présentation de quelques points et principes essentiels mis en avant dans l'approche proposée. Par ailleurs, nous dressons à la fin un certain nombre de perspectives de recherche visant à améliorer ou à étendre la démarche proposée sur d'autres domaines métier.

CHAPITRE I : LA MAINTENANCE INDUSTRIELLE

I. Introduction

Dans une entreprise, l'objectif principal de la fonction maintenance est de maintenir un bien dans un état lui permettant de répondre de façon optimale à sa fonction. Si l'entretien vise dans sa forme traditionnelle, à corriger les dérives de fonctionnement après leur apparition, la maintenance quand à-elle, vise à suivre au plus près le comportement de l'équipement et donc anticiper d'éventuelles défaillances. Ainsi, on peut conclure que « entretenir c'est subir alors que maintenir, c'est prévoir et anticiper ».

D'après la norme NF X 60-010 (ANFOR décembre 1994), la maintenance est définie comme étant « un ensemble des activités destinées à maintenir ou à rétablir un bien dans un état spécifique ou dans des conditions données de sûreté de fonctionnement, pour accomplir une fonction requise. Ces activités sont une combinaison d'activités techniques, administratives et de management ». L'agence AFNOR⁽²⁾ [AFNOR, 01] ajoute aussi que « bien maintenir, c'est assurer ces opérations au coût optimal ». Cette définition met en avant la notion du maintien par le suivi et la surveillance, la notion de rétablissement par les opérations de correction, état spécifique qui dépend des objectifs attendus et enfin, la notion du coût optimal qui suggère la nécessité de contrôler et de minimiser les coûts engendrés par les interventions.

Les évolutions permanentes dans les technologies liées à l'industrie (capteurs, outillage de mesure, etc.) et la généralisation des nouvelles technologies d'information et de communication, ont permis de développer dans la maintenance industrielle, de nouvelles techniques de surveillance notamment la thermographie et l'analyse optique. De nouvelles méthodes commencent à s'émerger telles que la maintenance participative⁽³⁾ et la maintenance prévisionnelle [Auberville, 04].

Ce chapitre est composé de trois parties : la première traite les aspects ainsi que les formes de la maintenance. Cette partie aborde aussi les différentes approches existantes dans la structure de la maintenance industrielle. La seconde partie aborde et décrit les systèmes de maintenance ainsi que leurs évolutions. La troisième partie sera consacrée à présentation de la problématique. Enfin, nous terminons ce chapitre par une conclusion des thèmes abordés.

⁽²⁾ **AFNOR** (Agence Française de NORmalisation) groupe spécialisé dans normalisation de tous les aspects liés à la maintenance industrielle (<http://www.afnor.org>).

⁽³⁾ **Maintenance participative** : désigne une série d'activités de maintenance qui nécessitent une collaboration entre le service maintenance de l'entreprise avec des partenaires externes notamment, les experts en maintenance ou encore les constructeurs des équipements industriels.

II. Aspects caractéristiques de la maintenance industrielle

Dans cette section, nous décrivons des aspects importants relatifs à la maintenance, notamment le rôle et les modèles organisationnels (ou approches) du service de maintenance dans une entreprise, leurs caractéristiques et leurs différentes politiques (formes).

II.1. Service de maintenance

Le rôle principal d'un service de maintenance est de mettre en œuvre une politique définie dans l'entreprise en vue d'atteindre le rendement optimal des équipements industriels. Le service de maintenance doit définir les stratégies les mieux adaptées selon le type de machine permettant d'établir des prévisions à long terme concernant les investissements, les planifications de maintenance selon la charge de production, ou encore des prévisions à court terme dans le cas d'intervention nécessitant un minimum de préparation [Auberville, 04].

Le service de maintenance peut avoir des relations avec plusieurs autres services au sein de l'entreprise tel que le service de production. Le service de maintenance peut être sollicité dans les cas suivant : en cas d'une ou plusieurs dérives observées ou en cas de concertation entre les deux services sur les plannings d'interventions, la gestion des ressources humaines et la gestion des stocks pour effectuer les réservations des ressources nécessaires aux opérations de la maintenance.

Par ailleurs, le service de maintenance peut avoir des relations qui s'étendent à l'extérieur de l'entreprise, notamment avec les services des sous-traitants pour les réparations externes, fournisseurs de matériels (pièces de rechange et outillage) et les constructeurs d'équipements industriels pour l'assistance technique comme illustré par la figure 1.1.

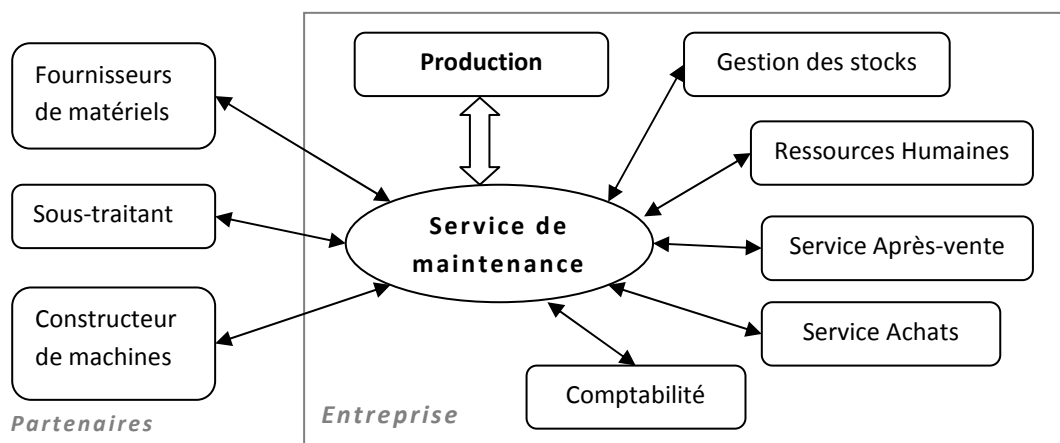


Figure 1.1 : Relations entre le service de maintenance avec les autres services d'une organisation complexe [Auberville, 04].

II.2. Approches de la maintenance

Selon la spécificité et la taille de l'entreprise, on distingue plusieurs types d'approches possibles dans la fonction de maintenance notamment : approche centralisée et l'approche décentralisée [Abbou, 03] ou encore les approches mixte, sous-traitée et décentralisée [Tararykine, 05]. Nous décrivons succinctement ces approches ci-après.

II.2.1. Approche de maintenance centralisée

Dans cette approche, toutes les activités de maintenances sont gérées par une structure unique qui est le service central de maintenance. Un atelier de maintenance est attaché à ce service permet d'assurer toutes les opérations de maintenance à travers toutes les unités de production présentes au sein de l'entreprise (figure 1.2). Le service possède aussi un organisme, appelé « bureau de méthodes », chargé de l'ordonnancement des travaux, de leurs préparation, réservation des ressources, etc.

Ce type de maintenance possède plusieurs avantages, notamment dans la gestion efficace des coûts induits en maintenance, l'unification des méthodes et les procédures utilisées et le mode de communication qui devient plus simplifié avec les autres services de l'entreprise grâce à son caractère centralisé [Abbou, 03].

Cependant, ce type d'approche peut être mal adapté pour les entreprises de grande taille où les unités sont éloignées géographiquement. Le service de maintenance devient donc plus difficile à gérer.

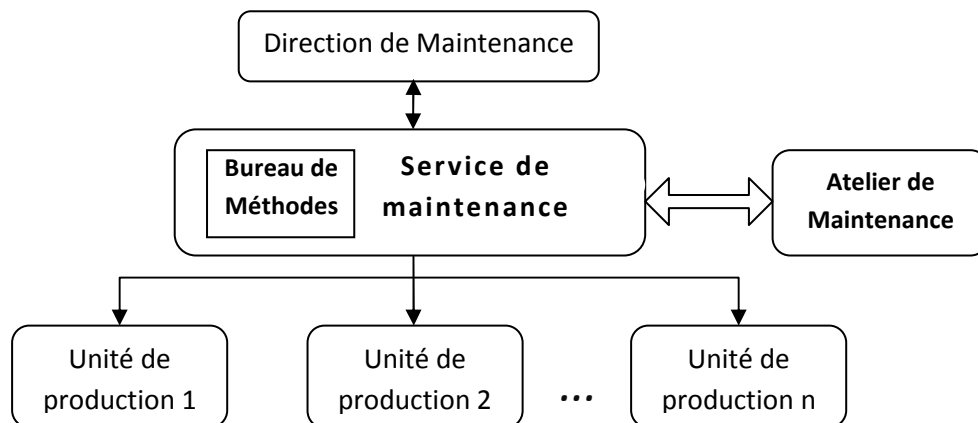


Figure 1.2 : Approche Centralisée de la maintenance

II.2.2. Approche de maintenance décentralisée

Dans cette approche, le service de maintenance n'a pas de direction commune, mais plutôt décomposé en plusieurs services. Chaque service est attaché à une structure de production bien spécifique (figure 1.3). Ce type de maintenance permet une meilleure communication entre les opérateurs humains sur le site et le service maintenance affilié à la même unité de production. D'autre part, la gestion administrative devient plus allégée en tenant compte du nombre réduit d'effectif et aussi des machines à gérer.

Cependant, munir chaque unité d'un service de maintenance présente les inconvénients suivant : (1) devient vite très coûteux lors de l'investissement dans les ressources (matérielles et humaines), (2) le manque de visibilité sur l'état du parc des équipements à gérer, (3) l'absence d'une politique commune au sein de l'entreprise sur les méthodes de maintenance et les procédures à adopter dans les activités liées à la maintenance [Abbou, 03].

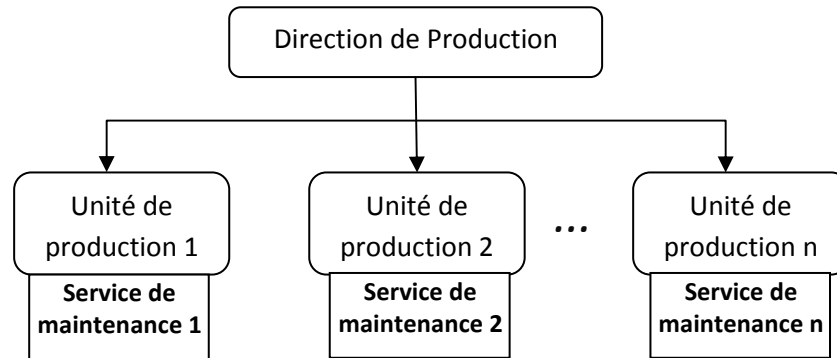


Figure 1.3 : Approche Décentralisée de la maintenance

II.2.3. Approche mixte

Dans cette approche, on a un service de maintenance central qui gère les activités de maintenance dans toutes les unités de production mais en déléguant certaines tâches (visites, interventions, etc.) aux sous-services présents dans chaque unité de production (figure 1.4). Cette approche a l'avantage d'être plus souple que l'approche centralisée car elle permet d'avoir une meilleure prévention et une bonne maîtrise technique grâce aux équipes présentes sur les sites de production tout en gardant la visibilité sur l'état global du parc des machines ainsi que les stratégies de maintenance adoptées.

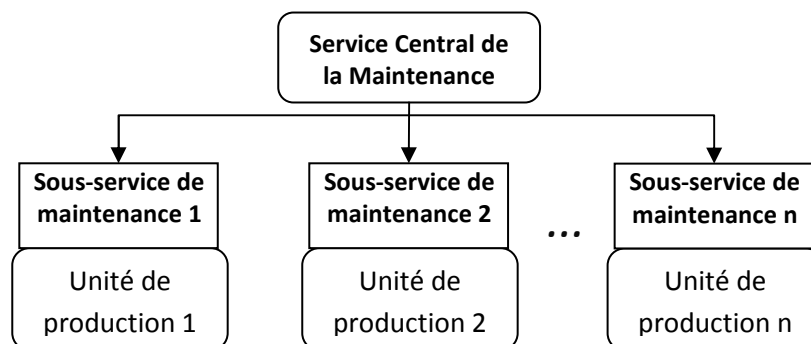


Figure 1.4 : Approche Mixte de la maintenance

II.2.4. Approche de maintenance sous-traitée

Dans certaines entreprises, la fonction de la maintenance est totalement externalisée et sous-traitée en vue de se concentrer plus sur les activités fondamentales de l'entreprise, c'est-à-dire le métier de production [Tararykine, 05]. Les avantages majeurs de cette approche se résument essentiellement par : (1) le gain obtenu en termes du nombre d'effectif intervenant sur le site de production, (2) décharger l'entreprise des tâches liées à la préparation des interventions, notamment la réservation des ressources matérielles et

humaines, et (3) bénéficier d'une bonne qualité des travaux effectués par un personnel hautement qualifié et spécialisé.

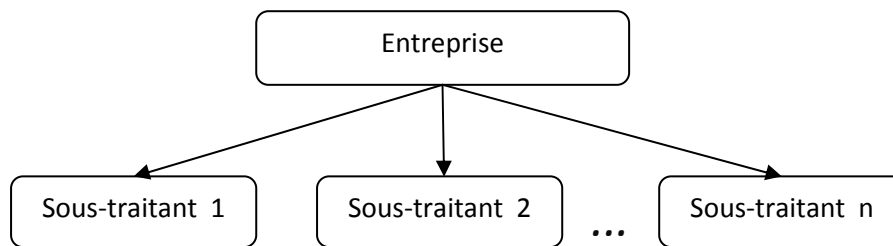


Figure 1.5 : Approche de la maintenance sous-traitée

Cependant, l'externalisation de la maintenance peut s'avérer désavantageuse pour l'entreprise. En effet, avec l'évolution des technologies, l'entreprise risque de perdre des champs de compétences et de savoir-faire. D'autre part, l'entreprise risque de devenir incapable d'évaluer la qualité des interventions ainsi que la pertinence des coûts facturés [Abbou, 03].

II.2.5. Approche distribuée

La maintenance distribuée est une nouvelle approche basée sur l'analyse des activités et les ressources selon une approche réseau [Kaffel, 01]. Dans ce type de maintenance, l'architecture du système comprend un ensemble de processeurs (humains, matériels et informationnels), internes ou externes à l'entreprise. Ces processeurs sont qualifiés pour réaliser un ensemble de processus. Un processus de maintenance est défini comme une séquence, partiellement ordonnée, d'étapes (sous-processus ou activités), reliées selon certaines contraintes de précédence, déclenchée par un évènement, réalisé par des ressources (ou processeurs) afin d'améliorer la compétitivité de l'entreprise [Kaffel, 01].

Il existe deux types de solutions pour cette approche [Tararykine, 05]. La première est la solution partagée où l'entreprise fait partie d'un « réseau d'entreprises qui se partagent une même Unité de Maintenance (UM) leader ». Dans ce cas, si pour un processus de maintenance déterminé, l'UM juge qu'une ressource humaine existante dans une entreprise X est inexploitée et qualifiée pour participer à la réalisation d'un processus de maintenance dans une entreprise Y, alors l'UM leader peut l'affecter à cette tâche.

La seconde solution est basée sur le principe de « partage et collaboration ». Dans ce type d'architecture, le réseau de maintenance est dynamique où les entreprises peuvent s'interagir directement entre elles, et en même temps, avec l'unité de maintenance partagée [Tararykine, 05]. Ceci crée une interaction entre les membres du réseau sans avoir besoin d'un leader, ainsi on obtient un réseau de maintenance dynamique [Kaffel, 01].

La figure 1.6 montre ces deux solutions ainsi que l'intensité existante du lien entre les entreprises (Ent_i) d'une part, et d'une autre part entre l'entreprise et l'unité de maintenance leader (UM).

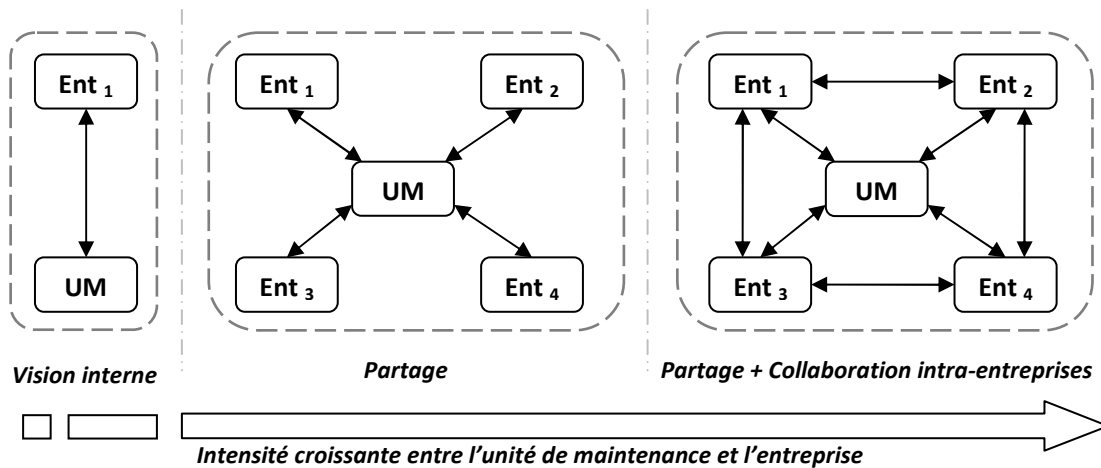


Figure 1.6 : Approche Distribuée de la maintenance [Kaffel, 01].

Plusieurs avantages peuvent être tirés en appliquant ce type approche dans les architectures des systèmes de maintenance industriels. En effet, l'entreprise peut bénéficier d'une gestion plus efficace des ressources (matérielles et humaines) et une réalisation plus rapide des processus de maintenance. En outre, en profitant d'un système d'information adapté selon les besoins et d'un suivi plus rigoureux des opérations, l'architecture du système de maintenance devient plus cohérente et plus flexible.

Enfin, l'entreprise bénéficiera, en collaborant avec d'autres unités de maintenances présentes sur le réseau, d'une maîtrise de la chaîne de logistique ainsi que de la diminution des arrêts aléatoires de la production et aussi d'une expertise externe importante pour gestion de la maintenance.

En conclusion, on insiste sur le fait qu'aucune approche n'exclue l'autre. Ainsi, les avantages de l'une peuvent présenter les inconvénients pour l'autre [Auberville, 04]. Par conséquent, la prise en compte d'une approche doit se faire non seulement par rapport à la structure et la taille de l'entreprise, mais aussi par rapport aux stratégies globales de l'entreprise visant à améliorer la productivité et aussi réduire le coût et les prix de reviens de ses produits.

II.3. Formes de la maintenance industrielle

Dans le domaine de la maintenance industrielle, on distingue deux grandes classes de stratégies (formes): (1) la maintenance préventive et (2) la maintenance corrective (figure 1.7). Nous décrivons succinctement chacune de ces formes ci-après.

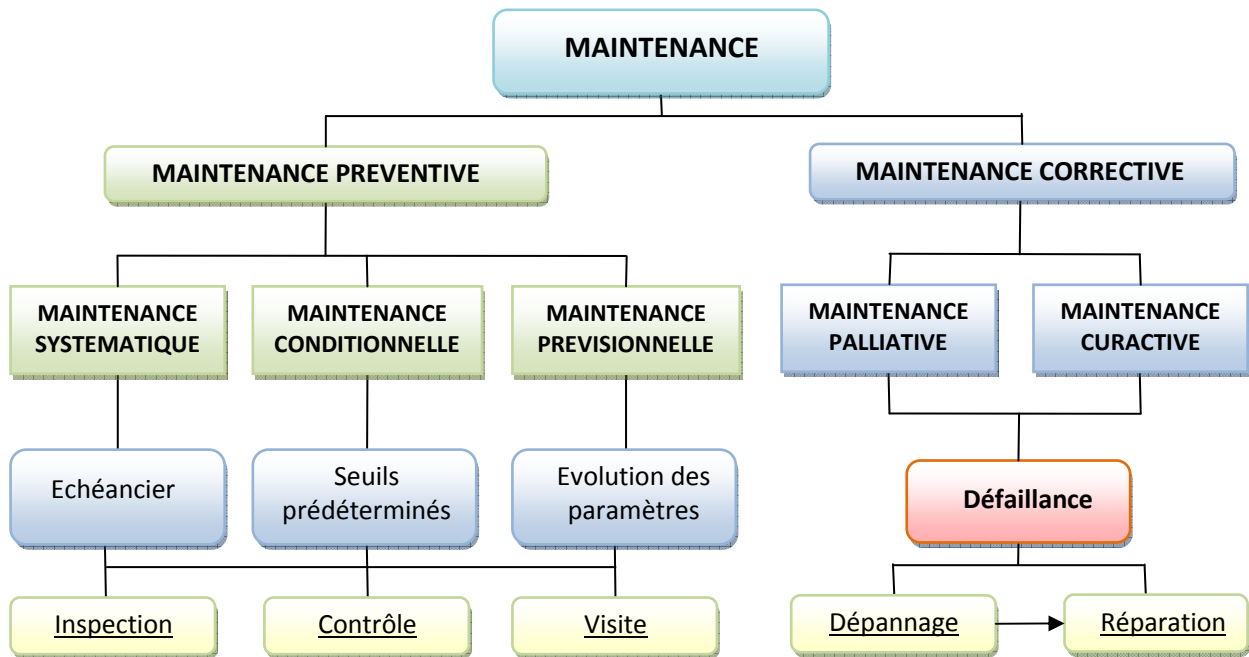


Figure 1.7 : Les Formes de la maintenance industrielle [Auberville, 04]

II.3.1. Maintenance préventive

La maintenance préventive a pour objet de réduire la probabilité de défaillance ou de dégradation d'un bien ou d'un service rendu. Les activités correspondantes sont déclenchées selon un échéancier (planning) établi à partir d'un nombre prédéterminé d'unités d'usage. Ce type de maintenance est connu sous le nom « **maintenance systématique** ».

Par ailleurs, ces activités peuvent être déclenchées en se basant sur des critères prédéterminés significatifs de l'état de dégradation du bien ou du service, c'est ce qu'on appelle « **maintenance conditionnelle** » [St-Marseille et Lapointe, 97].

Une autre forme plus récente de la maintenance préventive, appelée « **maintenance prévisionnelle** », commence à s'émerger dans le domaine industriel. Cette nouvelle forme se base, comme la maintenance conditionnelle, sur les technologies des capteurs mais en tenant compte des évolutions d'informations qu'ils délivrent. En effet, dans cette maintenance, on analyse l'évolution de certains paramètres significatifs liés à la dégradation du bien. Ainsi, par cette surveillance, il est possible de mieux planifier ou de retarder les interventions.

Dans la maintenance préventive, les opérations peuvent être regroupées en trois familles : les inspections, les contrôles et les visites [Frédéric, 03]. Les inspections sont des activités de surveillance s'exerçant dans le cadre d'une mission définie, tandis que les contrôles représentent un moyen de vérification de conformité du bien par rapport à des données préétablies. Enfin, les visites qui se traduisent par des examens détaillés et prédéterminés d'une partie ou de tout l'équipement industriel.

II.3.2. Maintenance corrective

Cette forme de la maintenance constitue l'ensemble des activités réalisées après la défaillance d'un bien, ou la dégradation de sa fonction pour lui permettre d'accomplir une fonction requise, au moins provisoirement : ces activités comportent notamment la localisation de la défaillance et son diagnostic, le remise en état avec ou sans modification, le contrôle du bon fonctionnement [Auberville, 04].

On distingue deux types de « maintenance corrective » : la maintenance palliative et la maintenance corrective. La maintenance palliative comprend toutes les activités de la maintenance corrective destinées à permettre à un bien d'accomplir provisoirement tout ou partie d'une fonction requise [AFNOR, 01]. Ce type de maintenance est connu sous le nom de **dépannage**. La maintenance palliative est principalement constituée d'actions à caractère *provisoire* qui devront être suivies d'actions curatives.

Tandis que la maintenance curative regroupe toutes les activités de maintenance corrective ayant pour objet de rétablir un bien dans un état spécifié ou de lui permettre d'accomplir une fonction requise. Les résultats des activités réalisées doivent présenter un caractère permanent [AFNOR, 01]. Ces activités peuvent être des **réparations**, des modifications ou aménagements ayant pour objet de supprimer le ou les défaillances.

III. Structure et architectures des systèmes de maintenance

Un système de gestion de la maintenance regroupe un ensemble d'acteurs (humains et matériels) qui utilisent un système d'information afin d'assurer la fonction de la maintenance. Le système d'information sous-jacent est constitué d'un ensemble d'informations d'échange, de processus et d'applications informatiques permettant de gérer tous ces processus liés à la maintenance.

Un système d'information dédié à la maintenance est organisé autour de trois sous-systèmes suivants :

- Le sous-système de pilotage et de décision, comprend toutes les fonctions liées aux suivis, décisions et ordonnancement des différentes tâches de la maintenance. Il définit aussi les objectifs et les stratégies à adopter dans la fonction de maintenance.
- Le sous-système opérant, comprend la réalisation des opérations de maintenance afin d'atteindre les objectifs fixés par l'entreprise. Il reçoit des informations sous différentes formes (ordres de travail, consignes de sûreté, etc.) pouvant être exploitées avec d'autres ressources (documents techniques, outils d'intervention, etc.) pour l'exécution des travaux et la gestion des différentes opérations de maintenance.
- Le sous-système d'information, sert à assurer les échanges entre les sous-systèmes de pilotage et opérant. Il doit être capable de relier d'une manière intelligente les différents intervenants, de leur acheminer une information complète et surtout, de les renseigner d'une façon précise sur l'état du système de production à tout instant.

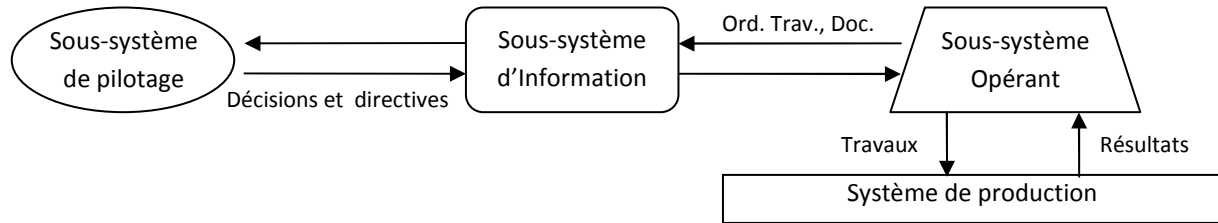


Figure 1.8 : Structure d'un système de gestion de la maintenance (d'après [Kaffel, 01])

La section suivante présente les différentes architectures d'un système de maintenance industrielle et leurs évolutions à l'ère des technologies de l'information et de la communication.

III.1. Architecture d'un système de maintenance

Un système de maintenance est composé d'une façon générale de deux principales classes de systèmes : le système physique et le système de gestion. Le système physique regroupe l'ensemble des outils matériels permettant d'alimenter le système de gestion de toutes les informations liées à l'état des installations présentes sur le site de production (figure 1.9). Ces informations seront utilisées par le système de gestion pour produire un ensemble de résultats ou de décisions liées à la maintenance [Rasovska *et al.*, 07].

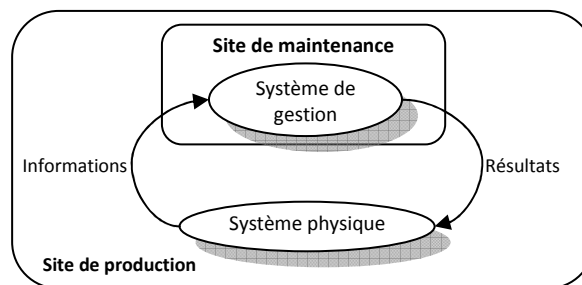


Figure 1.9 : Architecture d'un système de maintenance (d'après [Rasovska *et al.*, 07])

III.2. Typologies des systèmes de maintenance

Pour bien mener la fonction de la maintenance, on utilise donc plusieurs systèmes qui interagissent entre eux et aussi avec les utilisateurs (expert, techniciens, etc.) afin de réaliser les opérations de maintenance. Dans les paragraphes suivants, nous présentons quelques types de ces systèmes qui sont largement utilisés dans le domaine de la maintenance industrielle.

a. GMAO (*Gestion de Maintenance Assistée par Ordinateur*)

C'est un système informatique dédié à la gestion de la maintenance organisé autour d'une base de données permettant de programmer et de suivre sous trois aspects (technique, budgétaire, organisationnel) toutes les activités d'un service de maintenance et les objets de cette activité (services, lignes ateliers, machines, équipements, sous-ensembles, pièces, etc.) à partir de terminaux disséminés dans les bureaux techniques, ateliers, magasins et bureaux d'approvisionnement. La GMAO a pour principaux objectifs la réduction des coûts de

maintenance et l'amélioration de la disponibilité et de la fiabilité des équipements par l'utilisation de logiciels spécialisés.

Par ailleurs, l'ensemble des progiciels GMAO proposent presque les mêmes fonctionnalités similaires [Frédéric, 03] telles que : (1) l'ordre de travail (OT), qui constitue l'élément central dans la GMAO, où se concentrent diverses informations sur le ou les équipements concernés, le travail à réaliser, les ressources (matérielles et humaines) chargées de l'exécution, les coûts estimés des travaux, etc., (2) la base de données sur les équipements à maintenir, (3) la gestion des pièces de rechange et des stocks, (4) la gestion des achats (traitement des demande d'achat, transformation en commande, etc.), (5) la sous-traitance de la maintenance, (6) la maintenance programmée (ou préventive) où le module correspondant se charge de créer automatiquement des OT programmés en fonction du planning de tâches élaborés et (7) la gestion des ressources humaines où la GMAO gère les différents aspects sur les acteurs métier, notamment sur les qualifications des acteurs (superviseur, expert, etc.), la disponibilité des employés, les compétences du corps métier, etc.

La figure 1.10 récapitule l'ensemble des fonctionnalités génériques offertes par la GMAO.

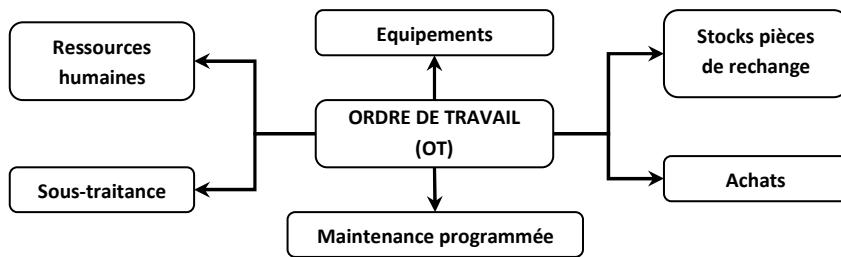


Figure 1.10 : Principales fonctionnalités d'un système de GMAO [Frédéric, 03]

L'architecture de la GMAO est basée essentiellement sur une base de données commune qui serve à maintenir toutes les informations sur le déroulement des tâches de maintenance (corrective, préventive, etc.). La figure 1.11 illustre un exemple d'architecture d'une GMAO.

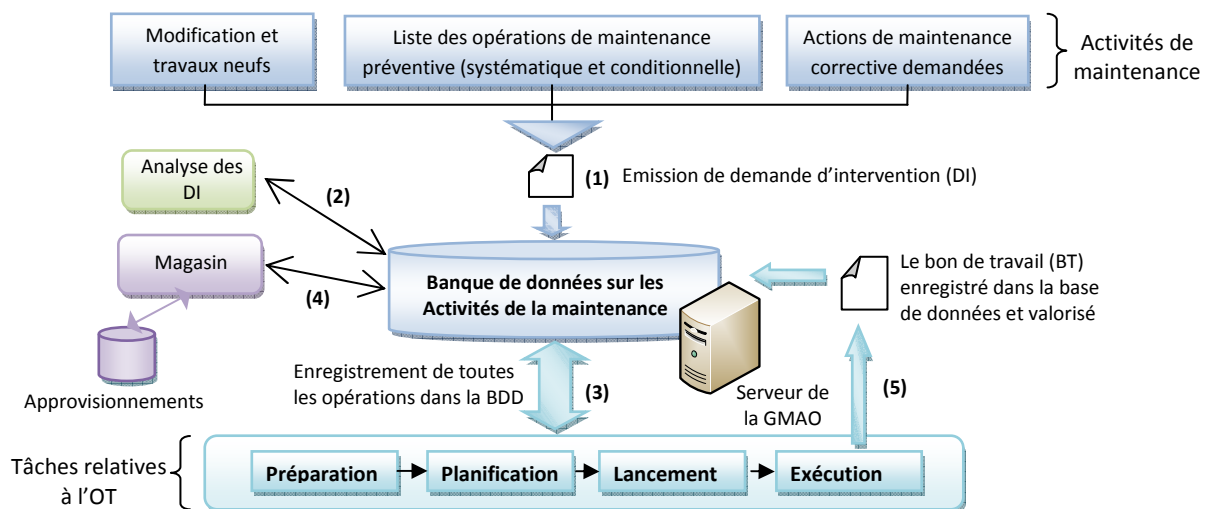


Figure 1.11 : Architecture globale d'un système GMAO (d'après [Frédéric, 03]).

b. Système de Documentation

Chaque action technique importante nécessite généralement la consultation d'une référence documentaire. La documentation relative aux équipements représente une des fonctionnalités les plus importantes de la maintenance. Cette fonction a un objectif principale d'avoir des connaissances technologiques et opérationnelles sur les équipements. Le système de documentation a donc pour rôle principale de gérer toutes ces ressources documentaire en offrant aux utilisateurs la possibilité de récupérer les informations pertinentes au bon moment.

c. SCADA (*Supervisory Control And Data Acquisition*)

SCADA est un système d'acquisition de données et de supervision destiné au suivi en temps réel des équipements industriels, et éventuellement la détection des dérives et alarmes en qui nécessitent une intervention immédiate ou sa programmation à une date ultérieure. Un exemple courant d'un système SCADA est celui des capteurs qui peuvent non seulement intercepter les dérives et les alarmes mais aussi de les traiter à fin de les envoyer sous une forme plus adaptée à un système de gestion maintenance (par exemple GMAO).

L'architecture d'un système SCADA est basée essentiellement sur les technologies relatives aux réseaux informatiques (figure 1.12).

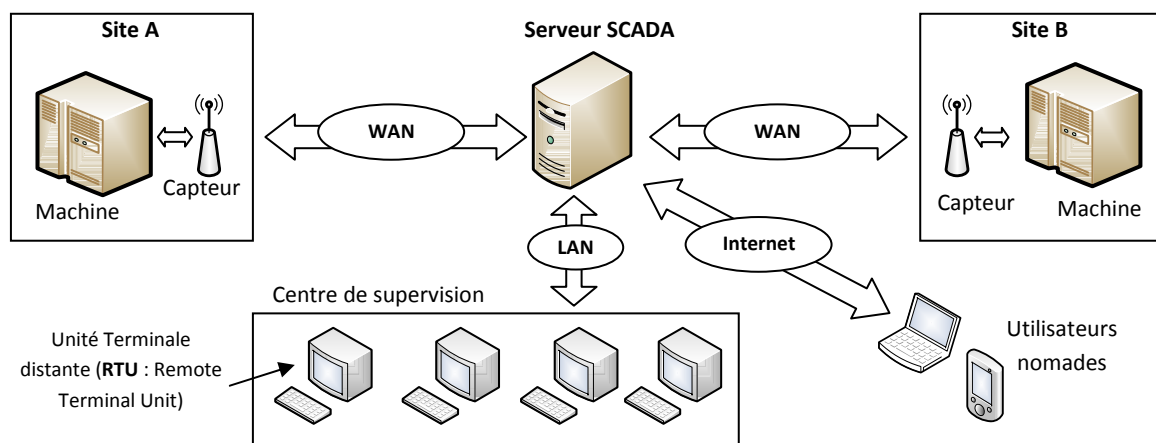


Figure 1.12 : Un exemple d'architecture d'un système SCADA [Barre et Fonash, 04].

Le schéma de la figure 1.12, montre un système SCADA composé d'un serveur central (*SCADA Master*) qui gère et maintient toutes les informations émises par les capteurs présents sur les sites de production. Par ailleurs les utilisateurs peuvent superviser et contrôler les installations soit au niveau du centre de supervision via les terminaux (RTU : *Remote Terminal Unit*) ou encore via Internet s'il s'agit d'utilisateurs nomades [Idoughi et Kolski, 06 b].

d. ERP (*Enterprise Resource Planning*)

Un ERP se présente comme un ensemble de composants logiciels avec lequel on peut construire un système d'information. Il propose de régler le problème d'organisation pouvant exister dans un système d'information. Il intègre toutes les informations de l'entreprise,

ressources humaines, gestion des commerciales, production, gestion comptable et évidemment la maintenance qui est en relation avec plusieurs fonctions de l'entreprise.

III.3. Evolutions dans les systèmes de maintenance

Les évolutions liées aux technologies industrielles, notamment au niveau des capteurs et les outils de mesure d'une part, et l'arrivée des nouvelles technologies d'informations basées sur le Web d'une autre part ont donné naissance à de nouvelles architectures pour les systèmes de maintenance. Dans cette section, on présente deux principales architectures suivantes : la télémaintenance et la e-maintenance.

III.3.1. La télémaintenance

La télémaintenance est une forme évoluée de la maintenance [Kolski et Millot, 93]. Elle est basée sur le principe suivant : les capteurs, mesurant des grandeurs intimement liées à l'état de la machine, sont reliés à une centrale de surveillance qui enregistre toutes les alarmes et les mesures. Cette technique permet d'une part, le suivi et l'enregistrement des données sur chaque machine à des fins de comparaison et d'autre part, la détection d'aléas de fonctionnement. L'agent de surveillance qui constate une évolution d'une dégradation ou l'apparition d'un défaut, a la responsabilité de mettre hors service, de consigner la partie lésée de l'installation et d'alerter les agents d'intervention.

Du coté architecturale, un système de télémaintenance est constituée de deux ou plusieurs systèmes ou sous-systèmes éloignés l'un de l'autre qui communiquent et échangent des données entre eux (figure 1.13). L'un des systèmes peut fonctionner comme un système d'acquisition de données; il représente l'émetteur de données structurées [Rasovska *et al.*, 07]. Le deuxième système est le récepteur, fonctionnant comme un système de traitement de données. Le système émetteur peut envoyer les données automatiquement ou comme réponse à une requête de la part du système récepteur des données.

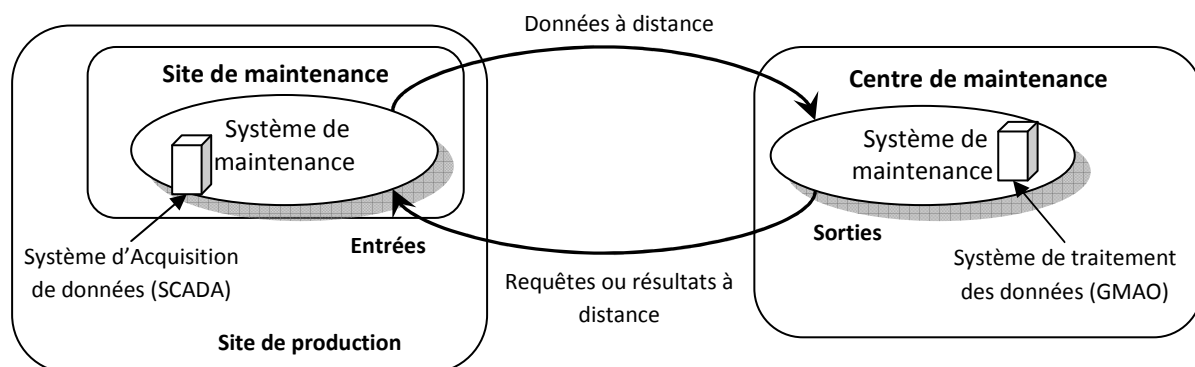


Figure 1.13 : Architecture d'un système de télémaintenance [Rasovska *et al.*, 07].

Les résultats du traitement de données (sorties) sont utilisés par les acteurs humains ou peuvent être renvoyés au système d'acquisition afin d'agencer l'acquisition des données. Les informations échangées doivent être structurées pour qu'elles soient acceptées par les systèmes communicants.

III.3.2. La e-maintenance

C'est une nouvelle forme de maintenance qui commence à prendre de l'ampleur au sein des entreprises industrielles. Cette forme intègre non seulement la notion de télémaintenance qui consistait à la récupération des données, au contrôle et supervision à distance des installations, mais aussi ajoute une dimension de coopération au niveau des informations et les différents acteurs (fournisseurs de pièces, expert, technicien,...) intervenants dans le processus métier de la maintenance et cela grâce aux technologies Web.

L'architecture d'e-maintenance est basée sur le réseau Internet ainsi que les technologies standards du Web. Cela permet de coopérer, d'échanger, partager et de distribuer les informations de maintenance entre les différents systèmes partenaires à travers le réseau d'Internet [Bangemann *et al.*, 05]. Le principe consiste à intégrer l'ensemble des différents systèmes de maintenance dans un seul système d'information (voir la figure 1.14).

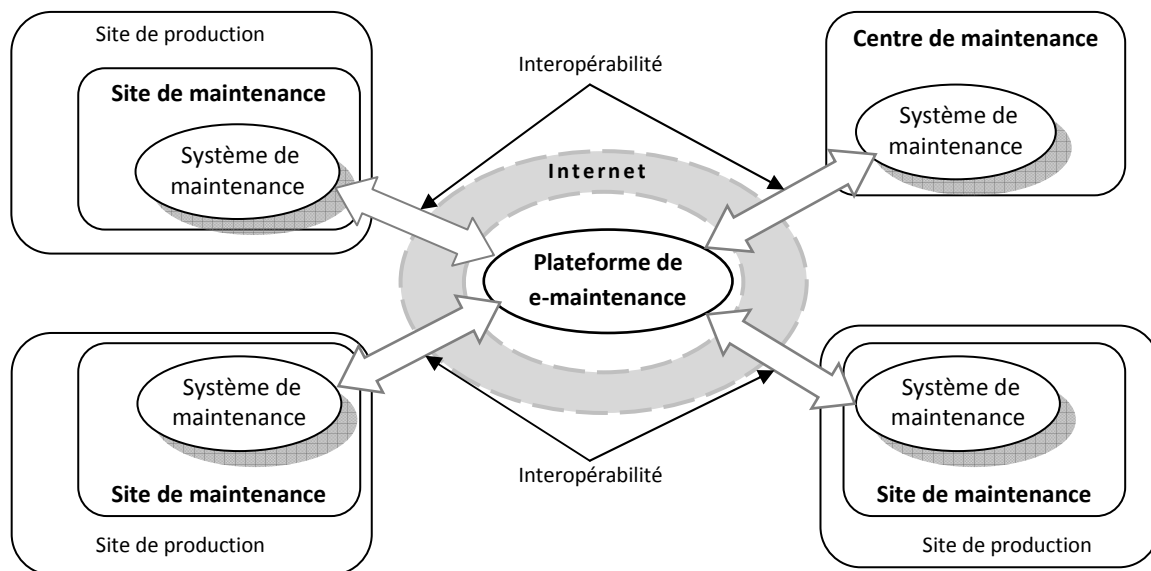


Figure 1.14 : Architecture d'un système de e-maintenance [Rasovska *et al.*, 07].

Ce type d'architecture a donné naissance à une nouvelle problématique liée à la compatibilité des échanges entre les systèmes de maintenance communicants. On parle alors de l'interopérabilité des systèmes. En effet, les systèmes qui interagissent à travers le réseau utilisent des informations qui ne sont pas forcément compatibles, d'où la nécessité d'intégrer de nouveaux outils au sein des systèmes de maintenance qui devront assurer leur interopérabilité.

Plusieurs architectures ont été proposées dans ce domaine dont la plateforme baptisée « Proteus ». Ce projet a pour vocation de concevoir et de développer un portail de maintenance à l'attention des opérateurs de maintenance. Au travers d'un site web, les opérateurs ont accès à certaines données de l'installation à maintenir, aux documentations techniques des éléments et des machines ainsi qu'à des bases de connaissances sur les métiers.

Dans l'architecture Proteus⁴, le portail de maintenance constitue une passerelle fédératrice entre les différentes entités en place aujourd'hui dans le domaine de maintenance [Rebeuf *et al.*, 05], à savoir : la gestion globale de l'entreprise ERP (Enterprise Resource Planning), les systèmes de supervision du processus industriel et la gestion de la maintenance assistée par ordinateur (GMAO).

L'objectif principal derrière ce projet est de fournir une plateforme distribuée coopérative d'e-maintenance incluant les systèmes existants (figure 1.15), notamment la gestion globale de l'entreprise ERP (Enterprise Resource Planning), les systèmes de supervision du processus industriel, de gestion de la maintenance assistée par ordinateur (GMAO), d'aide au diagnostic, de gestion de la documentation (e-Documentation), etc.

Tous ces systèmes communiquent via une plate-forme d'intégration unique (appelée noyau Proteus) permettant d'effectuer des tâches complexes impliquant plusieurs acteurs et systèmes.

Par ailleurs, le concept de cette plateforme est défini par la description unique et cohérente de l'installation à maintenir (connue sous le nom « d'ontologie »⁵), par l'architecture générique basée sur les concepts de Web services et en proposant des modèles et des solutions technologiques d'intégration [Bangemann *et al.*, 05].

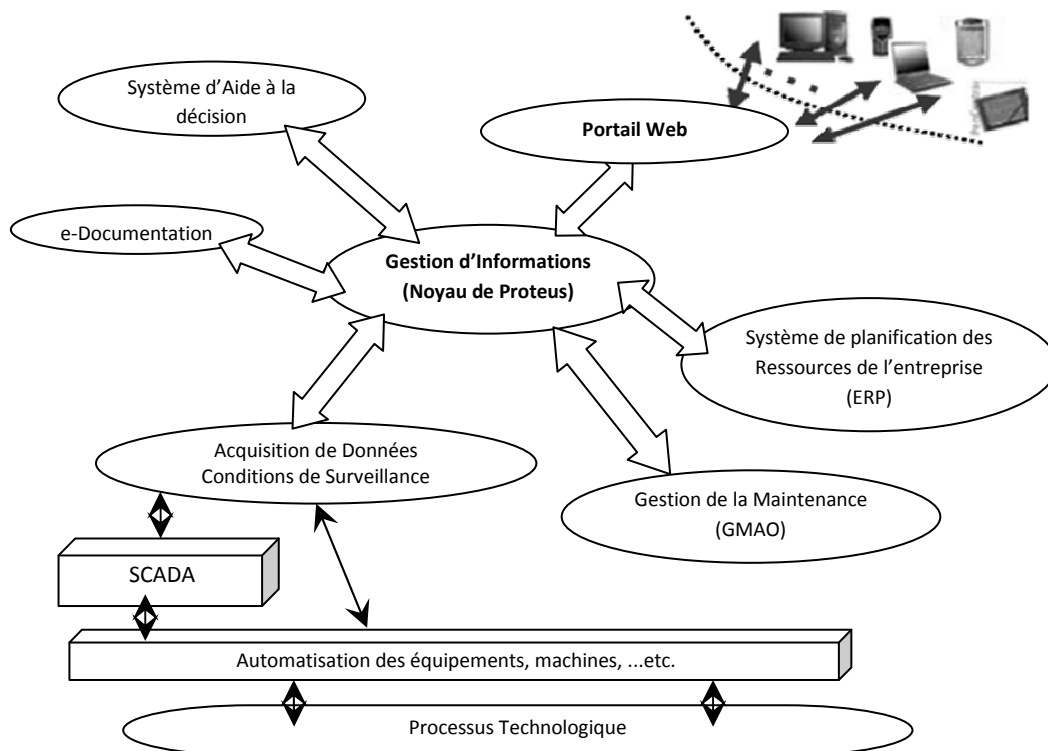


Figure 1.15 : La e-Plateforme de Proteus [Rebeuf *et al.*, 05]

⁴ Projet européen Proteus (<http://www.proteus-iteaproject.com>).

⁵ Une ontologie est une représentation formelle des connaissances partagées au sein d'une organisation sur un domaine particulier [Idoughi, 08].

IV. Synthèse et nouvelles motivations

A l'issue de ce qui a été présenté dans la section précédente, nous avons mis en avant un ensemble d'éléments caractéristiques aux systèmes de maintenance industrielle et inhérent à l'usage des technologies du Web pouvant être sources et origines de nouvelles motivations dans le processus de conception et de développement de nouveaux systèmes de maintenance. Ces nouvelles motivations sont présentées ci-après selon les axes suivants : (1) l'externalisation de la maintenance, (2) la problématique liée aux silos applicatifs, (3) l'interfaçage des applications, (4) la complexité des informations échangées et les dispositifs utilisés, (5) la nature des relations entre les systèmes impliqués dans la maintenance industrielle et enfin (6) sur les aspects liés aux interactions homme-machine.

IV.1. Vers une externalisation de la maintenance

Les entreprises d'aujourd'hui cherchent de plus en plus à améliorer leurs systèmes de maintenance par l'adoption de deux niveaux de stratégies [Abbou, 03] :

- Le premier niveau est de renforcer le service de maintenance interne à l'entreprise afin de mieux maîtriser les coûts de maintenance, et avoir un retour d'expérience sur la maîtrise technique des installations de production.
- Dans le second niveau, les entreprises veulent se concentrer sur les objectifs métiers majeurs, c'est-à-dire la production, par le biais des sous-traitants en réduisant ainsi le service de maintenance interne, on parle alors d'externalisation de la maintenance.

L'externalisation de la maintenance a des conséquences sur la structure des systèmes de maintenance qui sont désormais basés sur des environnements distribués. De plus, l'évolution des nouvelles technologies d'information et de la communication, notamment les technologies du Web et d'Internet, a contribué considérablement dans le développement des systèmes distribués.

De ce fait, l'externalisation et l'ouverture du service de maintenance aux différents partenaires externes à l'entreprise (fournisseurs, sous-traitants, etc.) donnent de nouvelles possibilités en matière d'interactions.

IV.2. Problématique des silos applicatifs

Comme nous l'avons déjà indiqué, l'aspect distribué des systèmes de maintenance fait que plusieurs types d'applications doivent communiquer, voir collaborer dans un environnement distribué. En effet, une application de gestion de maintenance (GMAO) exploite les informations issues des systèmes d'acquisition de données et de supervision (SCADA) qui peuvent se trouver dans des endroits différents, ou encore la gestion de production à travers le système GPAO (Gestion de Production Assistée par Ordinateur), qui est en relation étroite avec le système de maintenance, et enfin les progiciels de gestion globale de l'entreprise (ERP) fournissent des informations indispensables sur les ressources (humaines et matérielles) de l'entreprise pour le bon déroulement des activités de maintenance.

Cependant, on constate que ces applications sont regroupées au tour d'une fonctionnalité métier bien précise, et souvent installées d'une façon isolée au niveau des départements de l'entreprise. Par conséquent, les applications du SI sous-jacent sont généralement mal équipées pour communiquer avec les autres blocs du SI. On parle alors de « *fonctionnement en silo* », et cela à cause de l'hétérogénéité technologique du SI [Fournier-Morel *et al.*, 06].

Ce mode de fonctionnement entraîne la duplication des données et traitements, l'absence de processus métier transverses, la difficulté d'évolution des applications et la réutilisation qui devient impossible au niveau des composants logiciels.

La figure 1.16 illustre un exemple du fonctionnement d'un SI en trois silos applicatifs (A, B et C) relevant du domaine de la maintenance industrielle.

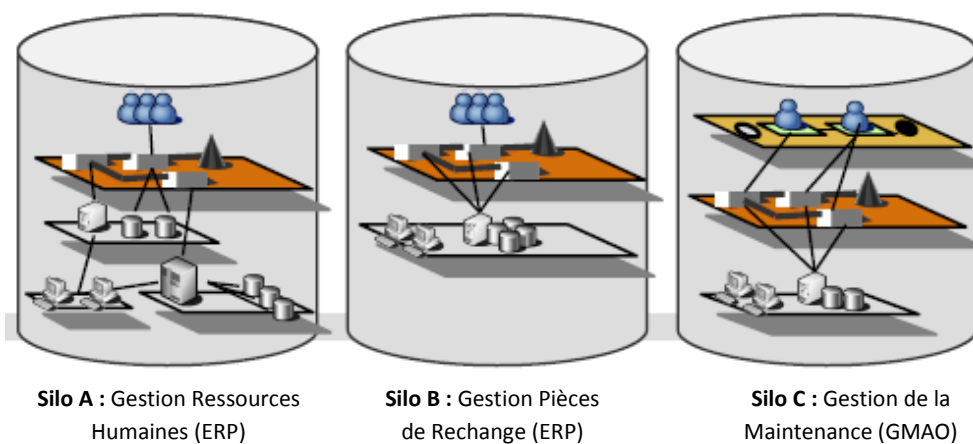


Figure 1.16 : Fonctionnement en silo. *Inspiré de* [Fournier-Morel *et al.*, 06]

On suppose dans cet exemple, qu'un opérateur de maintenance prépare une intervention sur un équipement. Pour cela, il établit une demande de réservation de ressources humaines ainsi qu'une demande de fourniture (DF) pour les pièces de rechange. Or dans ce cas, on constate qu'il est impossible d'exécuter ces tâches à partir du système de maintenance (GMAO dans notre cas) présent au niveau du service de la maintenance, car il n'existe aucun moyen permettant de lancer un tel processus métier qui est transverse, c'est-à-dire impliquant les différentes applications présentes aux niveaux des autres départements (silos applicatifs), notamment la gestion des ressources humaines (silo A) et celui de la gestion de stock (silo B).

IV.3. Interfaçage des applications

Dans les systèmes distribués, les applications sont souvent amenées à échanger des données et d'exécuter des traitements avec d'autres applications. Par conséquent, chacune de ces applications doit posséder une interface de communication définissant les types d'informations échangées et les fonctionnalités offertes, etc. Cette interface doit être définie pour toute application impliquée dans la communication. On parle alors de « *communication point à point* » [Manouvrier et Ménard, 07]. Cela peut engendrer un nombre important d'interfaces et aussi de canaux de communications (figure 1.17).

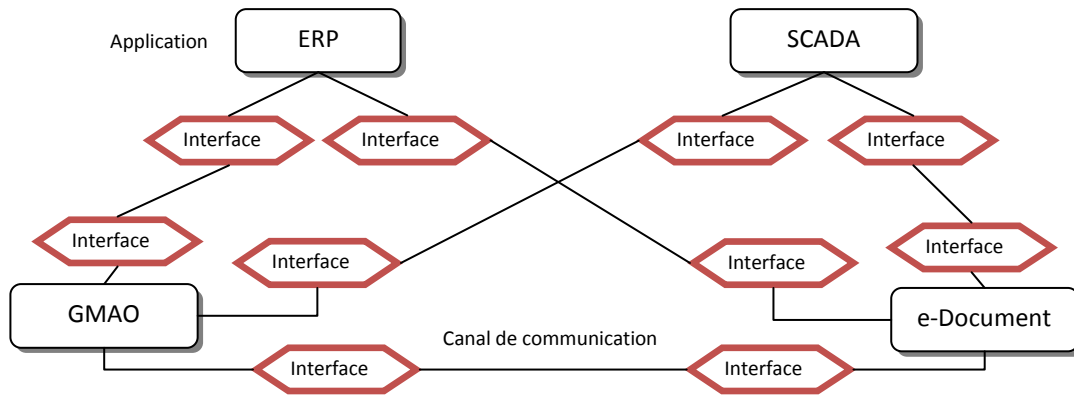


Figure 1.17 : L’interfaçage des applications. *Inspiré de* [Frédéric, 03]

En effet, le nombre maximal d’interfaces (N_{int}) qu’on peut avoir dans le système est donnée par la formule suivante [Manouvrier et Ménard, 07] :

$$N_{int} = N_{ap} * (N_{ap} - 1)$$

Avec N_{ap} : nombre total d’applications. Par ailleurs, le nombre maximal de canaux de communications (N_c) dans ce système est donnée par la formule suivante :

$$N_c = N_{ap} * (N_{ap} - 1) / 2$$

Par conséquent, si on a 8 applications communicantes, on peut avoir au total 56 interfaces et 28 canaux de communication. Par ailleurs, et selon [Frédéric, 03], les interfaçages dans le développement des applications représentent 30 à 40% des coûts d’un projet.

Dans la communauté scientifique, le problème de prolifération d’interfaces est connu sous le nom de « *syndrome du spaghetti* » [Manouvrier et Ménard, 07 ; Denney, 07 ; Alami, 05].

Afin de palier ce problème, une technique baptisée EAI (*Enterprise Application Integration*) a été spécifiée et qui permet de réduire le nombre d’interfaces entre les applications du SI par l’utilisation d’une part, d’un bus de communication (connu sous le nom de « *concentrateur ou Hub EAI* »), et d’un ensemble d’*adaptateurs* (ou connecteurs) pour chaque application connectée au concentrateur EAI de l’autre part. Ainsi, une application peut communiquer avec d’autres applications via un seul point d’accès comme illustré sur la figure 1.18.

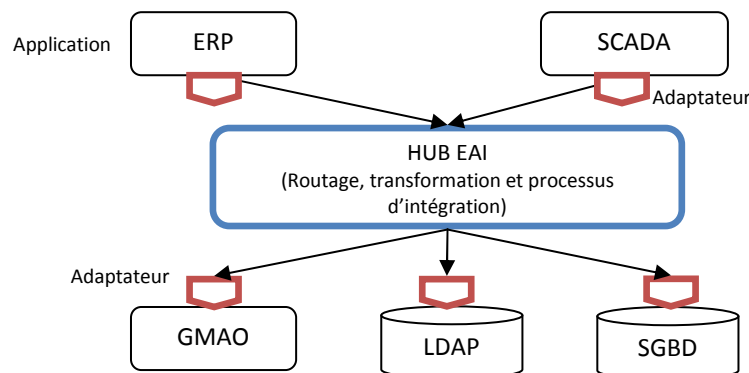


Figure 1.18 : Outils EAI (*Enterprise Application Integration*) [Fournier-Morel et al, 06]

Cependant, les solutions EAI souffrent des défauts suivant [Fournier-Morel *et al.*, 06] :

- *Solution propriétaire* : EAI offrent des mécanismes propre à l'éditeur de la solution. Ce qu'il les rend très dépendantes des technologies imposées par l'éditeur.
- *Manque d'exhaustivité des connecteurs* : en effet la plupart des outils EAI ne disposent pas d'assez de types de connecteurs permettant de connecter une large gamme d'applications hétérogènes.
- *Passage obligé par le Hub EAI limite le processus d'intégration* : en effet, le concentrateur (Hub) EAI, ne fait que le routage des informations vers les applications cibles présentes dans les différents silos applicatifs, ce qui favorise le fonctionnement en silo.

IV.4. Complexité liée aux informations échangées et le matériel de communication

L'information utilisée dans les différentes applications du domaine de maintenance a profondément changé ces dernières années suite à l'évolution des technologies d'information et aussi à la complexité croissante de l'environnement industriel [Bangemann *et al.*, 05].

En effet, dans les premiers systèmes de maintenance, les informations étaient généralement informelles (sous forme de papier ou échangées verbalement entre les opérateurs). Par contre aujourd'hui, l'information est devenue structurée et formalisée pour être manipulées par les systèmes informatiques [Rasovska *et al.*, 07]. En outre ces informations nécessaires à la maintenance industrielle sont devenues distribuées et l'accès à l'information, son traitement se font souvent dans des lieux différents [Idoughi et Kolski, 06 a]. Enfin, la variété des dispositifs utilisés dans les communications (PC, PDA, Pocket PC, Téléphone cellulaire, etc.), entre les opérateurs de maintenance d'une part, et d'autre part entre le système de maintenance et les opérateurs nécessite d'avoir des mécanismes de communication adaptables et souples face à cette diversité des dispositifs d'accès [Idoughi et Kolski, 06 b].

IV.5. Les aspects liés aux relations entre les systèmes de maintenance

L'évolution des NTIC ont permis également la migration des systèmes dits « autonomes », tels que GMAO, ERP et SCADA (section §III.2) vers des systèmes « intégrés » où la coopération et la collaboration sont devenues vitales pour tout fonctionnement [Rasovska *et al.*, 07]. La figure 1.19 montre l'évolution des relations entre les systèmes de maintenance en fonction de la complexité des informations échangées.

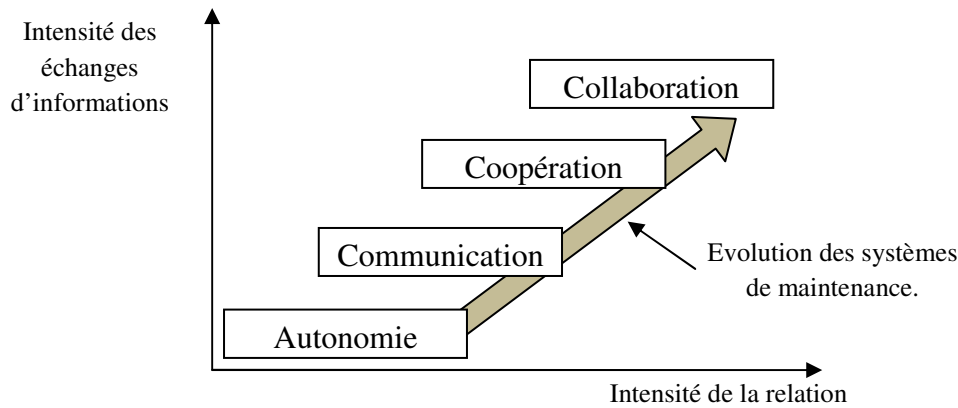


Figure 1.19 : Intensité de la relation entre les systèmes [Rasovska *et al.*, 07].

L'autonomie dans la relation suppose que chaque système dispose de toutes les ressources nécessaires afin de fonctionner indépendamment des autres systèmes. Dans ce cas, il n'existe aucun type de communication ou d'échange.

Quand aux relations de communication, il existe une ou plusieurs liaisons entre les systèmes qui permettent des transferts et d'échanges d'informations, dans ce cas on parle alors de télécommunication.

La relation de coopération est une forme évoluée de la communication. Ce type de relation suggère que chaque acteur (ou organisation) est responsable d'une partie de la résolution du problème. Pour ce faire, il est nécessaire d'avoir une coopération, en termes de compétences et de technologies, entre les systèmes indépendants afin de réaliser des projets communs de maintenance. On trouve généralement ce type de relation dans la maintenance distribuée (voir la section §II.2).

Par ailleurs, on assiste ces dernières années à un nouveau type de communication entre les systèmes de maintenance, il s'agit de la collaboration. Ce type de relation préconise « un partenariat stratégique » [Rasovska *et al.*, 07], en combinant les compétences des fournisseurs ou de matériels. La collaboration exige un effort coordonné pour résoudre un problème en mettant en commun des ressources, informations et des compétences.

Les aspects de coopération et de collaboration ont mis en avant de nouvelles problématiques liées à la conception des architectures pour les systèmes de maintenance capables de supporter l'intensité et la dynamique des relations entre les systèmes présentent dans des organisations différentes.

IV.6. Les aspects liés aux interactions Homme-Machine

L'interaction homme-machine joue un rôle important dans tout système de maintenance industrielle. Dans les premiers systèmes de maintenance, on considère que l'opérateur humain doit disposer d'une vue globale et unique sur le système à maintenir sans tenir compte de son profil, sa localisation, la nature des tâches effectuées et les moyens de communication dont il dispose [Idoughi et Kolski, 06 a]. En effet, le système de maintenance, installé au niveau du

centre de maintenance, n'offrait pas suffisamment de moyens d'interactions adaptés aux différents besoins et les situations dont l'opérateur de maintenance peut se retrouver.

Les problématiques liées aux interactions Homme-Machine peuvent se résumer en termes : (1) profil de l'utilisateur (expert, opérateur du terrain, etc.), (2) des dispositifs utilisés pour interagir avec le système (station de travail, PDA, téléphone portable, etc.) et enfin (3) en terme des tâches coopératives (diagnostiquer une panne, intervenir sur un équipement, analyser les résultats, etc.) et la localisation des opérateurs humains (au niveau du site d'incident, à l'extérieur de l'entreprise, etc.).

Par conséquent, le système de maintenance doit tenir compte de toutes les problématiques citées ci-dessus en offrant des mécanismes permettant aux opérateurs de réaliser leurs tâches selon leurs besoins, leurs dispositifs d'accès et la nature des activités accomplies.

En conclusion, tous les aspects présentés précédemment nous amènent à penser pour une refonte dans la modélisation et la conception de nouveaux systèmes de maintenance en tenant compte du domaine distribué, l'hétérogénéité, le caractère évolutif de ces systèmes et leurs modes d'interaction avec les acteurs humains. Cette refonte peut être fondée sur l'application de l'approche SOA (*Service-Oriented Architecture*) qui propose une nouvelle façon de concevoir les systèmes en mettant en avant des concepts liés aux services. Cette approche sera l'objet du prochain chapitre.

V. Conclusion

La maintenance industrielle constitue aujourd'hui un défi majeur pour les industriels, dans la mesure où les entreprises doivent disposer d'un système de production fiable pour répondre aux exigences d'une part, de ses clients en termes de qualité et quantité des produits et d'autre part, en termes de réactivité et de réduction des coûts de production. Afin de satisfaire ces demandes, l'entreprise doit disposer d'un système de maintenance efficace et peu coûteux.

Par ailleurs, l'évolution des formes (conditionnelle, prévisionnelle, etc.) ainsi que les stratégies (centralisée, sous-traitée, etc.) dans le domaine de maintenance industrielle a un impact considérable sur l'architecture du système de maintenance. En effet, les premiers systèmes étaient basés sur des architectures constituées de blocs applicatifs isolés et autonomes et qui ne peuvent être utilisés que dans le cadre d'une maintenance internalisée [Abbou, 03]. Par la suite, la maintenance est passée du caractère interne à l'entreprise vers un caractère externe où des acteurs et partenaires externes à l'entreprise peuvent être impliqués dans les activités du service de maintenance. Cela a donné de nouvelles motivations liées à la conception des systèmes de maintenance ouverts et souple face aux changements des objectifs métiers, de la structure interne et de l'écosystème (clients, fournisseurs, etc.) de l'entreprise. Ces systèmes doivent donc supporter les nouvelles exigences en termes de coopération et de

collaboration entre les différents acteurs et partenaires afin de réaliser des projets de maintenance en commun [Rasovska *et al.*, 07].

Par conséquent, il est nécessaire de construire les nouveaux systèmes de maintenance en tenant compte des aspects d'hétérogénéité, l'interopérabilité et surtout de l'intégration des différents systèmes informatiques participants dans le cadre de la maintenance industrielle.

C'est pourquoi l'adoption de l'approche Orientée Services (SOA) nous paraît une solution idéale pour construire des systèmes de maintenance agiles et évolutifs. Le chapitre 2 a pour objectif de présenter cette nouvelle approche, ses aspects et les outils fournis pour bâtir une architecture orientée services capable de soulever toutes les contraintes citées précédemment.

CHAPITRE II : L'ARCHITECTURE ORIENTEE SERVICES (SOA) ET LES SERVICES WEB

I. Introduction

L'architecture orientée services (SOA de l'anglais *Service-Oriented Architecture*) est le terme utilisé pour désigner un modèle d'architecture pour l'exécution d'applications logicielles réparties [Way, 08]. Ce modèle est défini par une approche destinée à rendre plus agile et réactifs les systèmes d'information (SI) distribués et hétérogènes en proposant une nouvelle manière d'intégrer et de manipuler les différentes briques et composants applicatifs d'un système informatique et de gérer les liens qu'ils entretiennent [Manouvrier et Ménard, 07]. Ces définitions mettent en avant le concept d'intégration applicative qui consiste à faire cohabiter plusieurs applications dites hétérogènes au sein du même système d'information.

Cependant, cette approche n'est pas nouvelle, car de nombreux modèles d'architecture ont été largement utilisés dans le domaine de l'informatique distribuée, notamment CORBA (*Common Object Request Broker Architecture*) et DCOM (*Distributed Component Object Model*). Néanmoins, ces deux modèles relèvent de l'architecture par composants logiciels répartis plutôt que de l'architecture orientée services, et le terme « **service** » est généralement absent de leur terminologie [Salatgé, 06]. En outre, ces modèles souffrent de l'absence de standard d'échange et c'est pourquoi l'approche SOA s'est réellement développée avec l'émergence des technologies liées aux services Web. De plus, un des avantages des SOA réside dans le fait que les applications réparties n'ont plus besoin d'un système de middleware⁶ réparti commun pour communiquer, mais seulement des protocoles et des technologies de communication interopérables (services Web) sur Internet.

Construire une architecture orientée services signifie donc d'abord concevoir une architecture basée sur un réseau de relations de services, entre applications réparties. La description d'une relation de services est formalisée par un contrat de services. Ce contrat décrit les engagements réciproques du fournisseur et du consommateur (client) du service (figure 2.1).

L'émergence des technologies de Services Web est censée apporter un niveau d'interopérabilité très élevé avec un degré de couplage faible [Bonnet, 05 a]. Une architecture d'applications réparties faiblement couplées (ou avec un degré de couplage très faible) est constituée d'un ensemble décentralisé d'applications réparties autonomes (Services Web), lesquelles interagissent sur la base de protocoles de communications, et sont mises en œuvre à l'aide de technologies ouvertes [Endrei *et al.*, 04].

Dans ce qui suit, nous présentons les éléments et concepts clés de l'approche SOA. Ensuite, nous passons en revue sur les technologies liées aux services Web, qui constituent aujourd'hui

⁶ Middleware : Ensemble des logiciels ou technologies informatiques qui servent d'intermédiaire entre les applications et le transport des données via le réseau. Source Wikipedia (<http://www.wikipedia.fr/>).

un moyen incontournable pour la mise en œuvre d'une architecture orientées services, tout en mettant l'accent sur certains aspects jugés utiles pour la modélisation orientée services d'un système de maintenance industrielle.

II. Eléments et concepts clés de SOA

Dans cette section, nous présentons quelques éléments correspondants aux aspects conceptuels de l'approche orientée services.

II.1. Le concept de service

Un service consiste en un mécanisme à travers lequel il est possible pour des acteurs (humains, systèmes des partenaires, etc.) d'accéder à une ou plusieurs ressources ou fonctionnalités métier en invoquant des opérations. Cet accès est défini via une interface dont l'appel est soumis à des contraintes et des règles (Contract & Policy) spécifiées dans la description du service [Alami, 05].

Un service suppose l'existence de trois éléments suivants : un producteur, des consommateurs et un contrat de service (figure 2.1). Le producteur est un acteur qui fournit le service ; les consommateurs sont considérés comme des clients ou les demandeurs de services; enfin, les termes du service rendu sont définis dans un contrat de service qui sert d'interface d'échange entre les deux parties (fournisseur et client).

Du point de vue analyse du SI, un service est considéré comme une fonction F . Il est défini par au moins deux flux d'informations, avec des directions opposées entre le «Demandeur du Service » et «Fournisseur du Service ». Donc on peut spécifier un service (F) en fonction de la demande et la réponse par la formule suivante [Gustienne et Gustas, 07] :

$$\text{Réponse Service} = F(\text{Demande Service})$$

Le fournisseur de service répond à la requête du demandeur de service par le biais d'un message sans avoir le contrôle direct sur le demandeur du service. Ainsi on peut voir un service comme une boucle d'interactions entre les deux participants. Cette idée est illustrée dans figure 2.1.

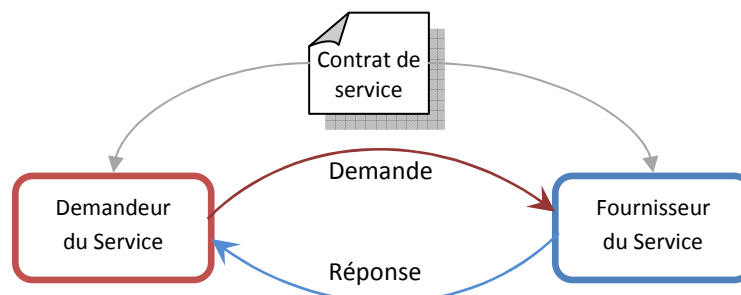


Figure 2.1 : Service : boucle d'interactions entre demandeur et fournisseur de service [Salatgé, 06]

II.2. Les principes de base des SOA

Afin de construire une solution SOA basée sur les services, on doit respecter certains principes fondamentaux. Ces principes représentent les éléments clés dans toute démarche SOA qui vise à bâtir des architectures logicielle autour d'une collection de services supportant les différentes fonctions métier et peuvent collaborer pour satisfaire tous les besoins métier.

Nous décrivons dans ce qui suit, neuf (09) principes de base dans l'approche orientée services. Il s'agit en fait d'une synthèse des travaux effectués par plusieurs auteurs notamment dans [Porter-Roth, 05 ; Vauquier et Bonnet, 06 ; Raymond, 07].

Couplage lâche (faible) : cela signifie que la logique contenue dans un service aura un minimum d'impact sur les autres services utilisés dans une même solution SOA. Le couplage lâche (*loose coupling* en anglais) est l'élément clé de l'approche SOA. Implémenter des services comme des parties logicielles à faible couplage permet de satisfaire les autres principes SOA, à savoir la réutilisation, l'autonomie des services sans états qui seront abordés par la suite. Donc il s'agit d'un point de départ de toute construction d'une architecture SOA.

Autonomie : signifie que les services contrôlent uniquement leurs logiques dont ils encapsulent. Ainsi, dans SOA, on ne doit pas avoir un état du service contrôlé par un autre service. Cela permet d'élaborer des solutions SOA plus flexibles, notamment dans la réutilisation et la composition des services.

Réutilisation : dans SOA, la réutilisation signifie que chaque service peut être utilisé par un ou plusieurs consommateurs de services (portails Web, applications composites, etc.). Bâtir une solution SOA sur des services réutilisables permet de composer dynamiquement ces services afin de satisfaire les nouveaux besoins métier, et donc arriver à l'alignement métier.

Composition : représente la capacité des services d'être regroupés dans une composition de services qui coordonne l'échange de données avec les autres services agrégés. Cela signifie qu'on peut construire des services à grande granularité (*coarse grained*) sur la base de services dotés d'une granularité fine (*fine grained*).

Services sans état : cela signifie que les services ne doivent pas maintenir leurs états spécifiques dans une activité. En d'autres termes, l'état actuel d'un service dans une activité ne dépend pas de ses anciens états au cours des activités qui se sont déroulées auparavant.

Contrat de service : désigne le lien entre le client qui requiert le service et le pourvoyeur capable de proposer le service. Les deux parties (client et fournisseur) doivent respecter certaines exigences formulées sous forme d'un contrat. Ce contrat doit contenir une description des opérations du service et fournir d'autres documents permettant aux programmeurs d'accéder et d'utiliser le service [Salatgé, 06]. Un contrat permet donc la réutilisation facile et garantit l'interopérabilité entre les services.

Un contrat contient en général les éléments suivants (figure 2.2) :

- Les signatures des opérations qui sont spécifiées par le triplet <nom opération, message de la requête, message de la réponse>,
- La structure et le format de chaque information,
- Indication sur les protocoles de communication utilisés,
- Informations sur la localisation des services (généralement l'adresse du serveur),
- Informations sur les contraintes non-fonctionnelles (qualité de service, sécurité, etc.).
- Les directives d'emploi à travers les pré-conditions et post-condition. Les pré-conditions permettent d'énoncer les règles à respecter pour le déclenchement de l'opération. Pour chaque pré-condition on précise également la post-condition qui définit les conditions d'émission du résultat de l'opération [Bonnet, 05 a].

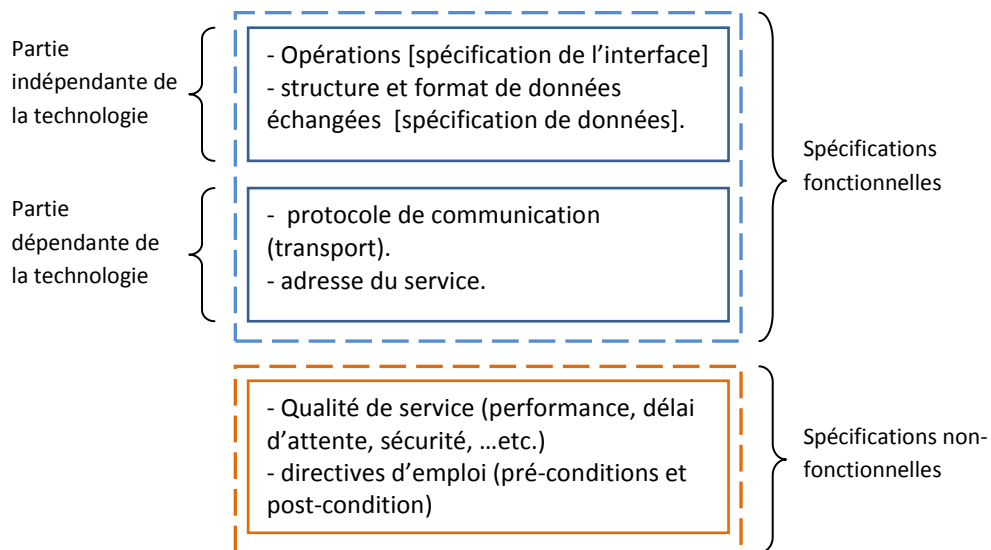


Figure 2.2 : La description d'un contrat de service

Comme illustré sur la figure 2.2, la description d'un contrat est constituée de type de spécification : fonctionnelle, c'est-à-dire utilisée par les programmeurs au niveau du client pour invoquer le service, et non-fonctionnelle, destinée aux maîtrise d'ouvrage (acteurs métiers) coté client pour vérifier les garanties ainsi que le niveau de qualité offert par le pourvoyeur (fournisseur) [Fournier-Morel *et al.*, 06].

En outre, la partie formelle peut être subdivisée en deux parties : partie indépendante de la technologie utilisée par les deux parties (pourvoyeur et consommateur), notamment dans la définition des formats des données échangées et les opérations offertes par le service, et une autre partie dépendante de la technologie, notamment dans la définition du protocole de transport utilisé (HTTP, SMTP, etc.) et aussi la localisation du service au niveau du serveur d'application en utilisant généralement le couple numéro du port et l'adresse du serveur.

Encapsulation de l'implémentation : signifie que le service doit uniquement exposer, d'une façon publique, sa logique métier décrite dans le contrat de service, et ainsi cacher aux consommateurs du service tous les détails sur l'implémentation. Ainsi, les services devront interagir avec le monde extérieur uniquement via leurs interfaces publiques.

L'interface de service est le cœur du concept SOA [Manouvrier et Ménard, 07]. En effet, un service est toujours destiné à un accès par programmation à partir d'un autre composant (consommateur de service). Cependant, cette interface doit disposer de suffisamment d'informations pour qu'un service puisse être identifié et utilisé sans avoir à connaître les détails sur sa conception interne ainsi que son contenu d'implémentation.

La figure 2.3 illustre un exemple d'interfaces définies pour un service de « Gestion de la documentation d'une machine » dans un système de maintenance industrielle.

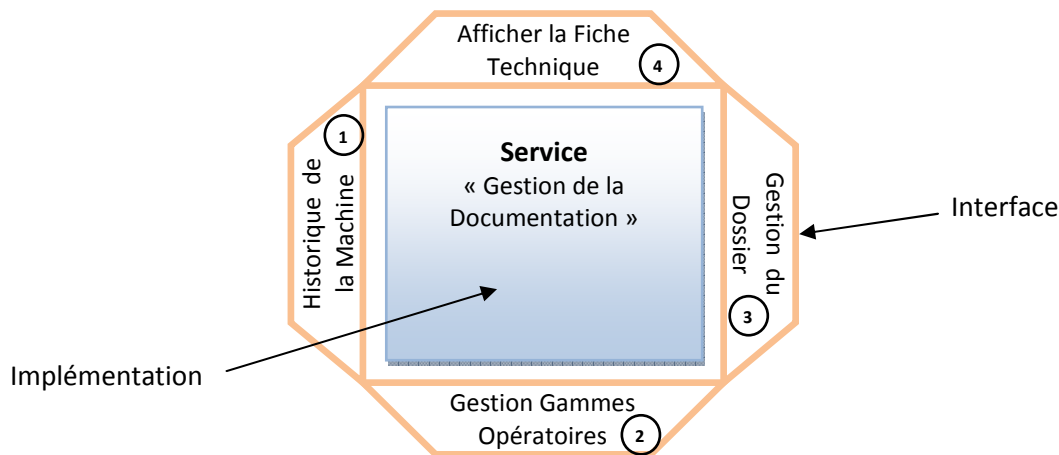


Figure 2.3 : Principe d'Encapsulation de l'implémentation

Le service « gestion de la documentation » possède donc quatre interfaces : (1) une interface pour le traitement de l'historique des interventions effectuées sur la machine, (2) une interface qui définit les opérations pour la gestion des gammes opératoires⁷, (3) une interface qui englobe l'ensemble des opérations pour la gestion du dossier de maintenance correspondant à la machine et enfin (4) une interface pour la consultation de la fiche technique définie par le constructeur.

La conception d'une architecture orientée service implique donc la définition de toutes les interfaces de service ainsi que leurs interactions, c'est-à-dire les interactions entre services et avec les consommateurs de services.

Interopérabilité : ce concept correspond à la capacité pour les services d'interagir avec d'autres services et applications indépendamment de leurs plateformes et aussi leurs implémentations. Cette interopérabilité est devenue possible grâce à l'utilisation des interfaces standards fournies à base des langages et protocoles standards notamment WSDL (*Web Service Definition Language*) et SOAP (*Simple Object Access Protocol*) qui seront décrits dans les prochaines sections.

Découverte dynamique de services : ce principe basé sur un ensemble de mécanismes permettant aux consommateurs de services de trouver dynamiquement les services dont ils ont besoins par le biais d'un annuaire (registre) de services [Chauvet, 02].

⁷ Gamme opératoire : représente les opérations à réaliser sur un équipement en se référant à un ensemble de critères et de paramètres (unités d'usage, rendement, ...etc.).

Ainsi, toute solution SOA devra fournir des outils efficaces pour la publication et la recherche des descriptions de services. La spécification UDDI (*Universal Description, Discovery and Integration*), dans le cas des Services Web, fournit (voir la section sur les Services Web) ce type de mécanismes, en permettant la publication des contrats de services, ainsi que leurs recherches d'une façon transparente et dynamique (c'est-à-dire, indépendamment de leurs emplacements et implémentations) à travers le registre (ou annuaire) de services.

Par conséquent, la découverte dynamique des services est souvent représentée par le triangle (figure 2.4) formé de : fournisseur de service, consommateur et le registre de services. On parle alors de « triangle de collaborations dans SOA » [Endrei *et al.*, 04].

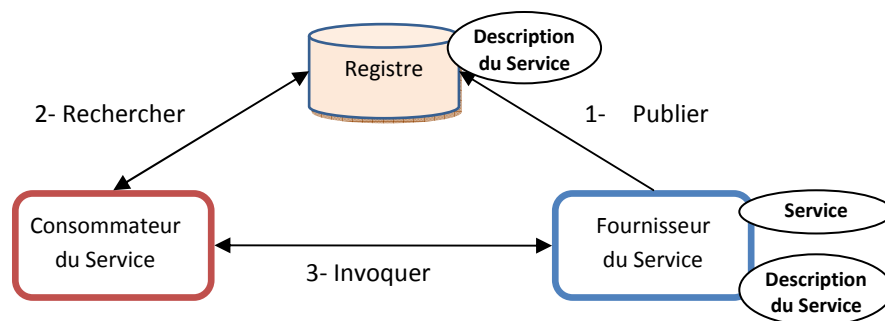


Figure 2.4 : Le triangle de collaborations dans SOA [Endrei *et al.*, 04].

Comme illustré sur la figure 2.4, le fournisseur de service doit d'abord publier la description des interfaces de son service ; le consommateur procède à la recherche du service requis à travers le registre et une fois le service trouvé ; le consommateur invoque le service via son interface.

II.3. Caractéristiques générales des services

Nous présentons dans cette section un ensemble de caractéristiques des services selon SOA.

Typologie des services

La typologie de services est une notion qui correspond aux types d'usages des services dans SI [Idoughi, 08]. On distingue deux (02) types de services lors de la modélisation des services dans l'approche SOA, les services métiers et les services techniques [Fournier-Morel *et al.*, 06].

- Les services métiers (ou organisationnel) : le service métier offre un ensemble cohérent de traitement métier. Il peut être un service d'accès à des informations, un service de calcul, un service de vérification de règles métiers, ou encore une composition de ces différents services. Voici quelques exemples de ce type de services : récupérer l'historique des interventions sur un équipement, calcul du coût de main-d'œuvre engendré par une intervention, etc.

- Les services techniques : le service technique (ou informatique) permet de fournir des accès aux ressources techniques spécifiques telle que les imprimantes, serveur de messagerie, outil de journalisation, accès à des données réparties, etc. Ces services peuvent être appelés par les services métiers pour exécuter certains traitements (par exemple, un service métier en lien

avec la gestion de maintenance, permettant de fournir un état annuel sur les coûts de maintenance, nécessitant le service technique d'impression).

Granularité de services

La granularité est un indicateur informel lié au périmètre fonctionnel couvert par le composant (service) [Raymond, 07], donc c'est une notion qui se réfère à la complexité du service. En fait, lorsqu'on définit les services, on doit choisir un niveau de granularité le plus approprié. Le bon niveau du service dépend de l'emplacement du service dans l'architecture. Les services de haut niveau délivreront de la valeur à la couche métier, notamment pour les clients (figure 2.5). Ces services sont définis comme étant d'une granularité à gros grains (*coarse grained*), alors que les services du grain le plus fin (*fine grained*) sont représentés par les services techniques, c'est le cas des services d'accès aux référentiels.

Tandis que les services avec grain moyen (*medium grained*) qui sont généralement représentés par les services fonctionnels se trouvent entre les deux niveaux déjà cités. Par exemple, les services de gestion de données sont des services de bas niveau qui vont mettre à la disposition des services fonctionnels (grain moyen) ou encore des services métier (gros grain) les données provenant des sources de données hétérogènes.

La figure 2.5 montre les relations entre les services les plus abstraits jusqu'au plus fins.

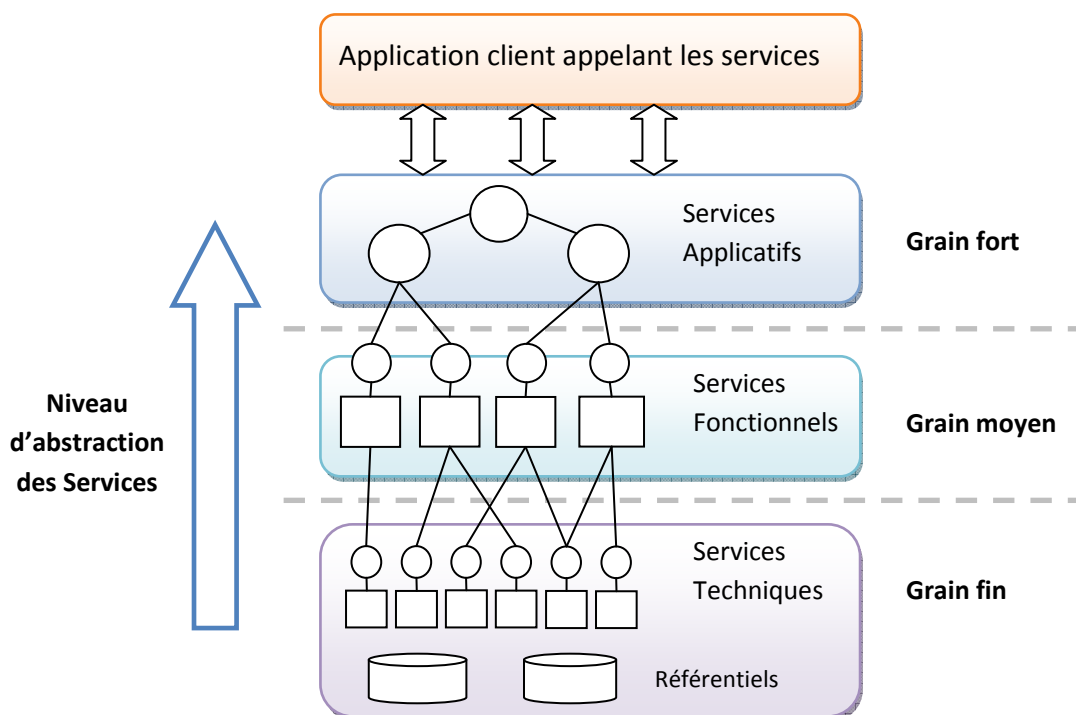


Figure 2.5 : Typologie et la granularité des Services [Bonnet, 05 b]

Comme le montre la figure 2.5, plus on monte en abstraction, vers le haut, et plus les services développés font appel aux services de niveaux plus bas.

Composants de services [Raymond, 07]

L'architecture orientée service est basée sur la notion de composants de services. En effet, le principe fondamental apporté par SOA est le suivant : découper le système d'information sous

forme de briques fonctionnelles, où chaque brique est associée à une fonction métier bien spécifique et qui doit être potentiellement simple à faire évoluer.

Ce découpage du système donne naissance à un ensemble de composants de services où chacun d'entre eux est décomposé en deux parties : vue externe qui est la spécification du service, et une vue interne qui représente la logique métier, c'est-à-dire tous les programmes de traitements, les classes utilitaires pour l'accès aux données, etc.

De ce fait, la partie externe est souvent appelée « boîte blanche » car elle permet d'exposer des traitements et la partie interne connue sous le nom de « boîte noire », car elle cache tous les détails liés à l'implémentation du composant de service.

La figure 2.6 illustre une vue simplifiée du découpage en un ensemble de composants de services dans une approche SOA.

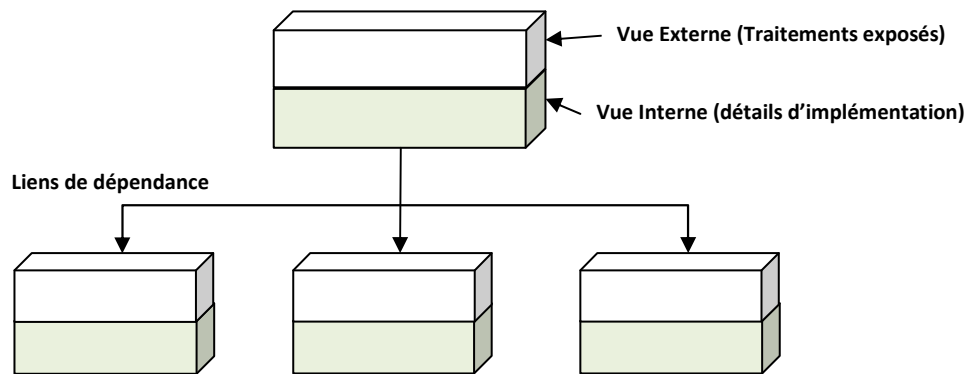


Figure 2.6 : Représentation du système par composants de services (d'après [Raymond, 07])

On peut remarquer que chaque composant est invoqué par les autres composants du système via sa partie externe (interface) et que la communication se fait de haut en bas, cela montre que les composants système de granularité fine n'appellent jamais les composants de granularité plus importante.

II.4. L'infrastructure SOA et ses composants

Dans cette section, nous allons présenter un modèle SOA (figure 2.10) qui met en évidence tous les éléments de base qui forment une architecture orientée services.

1) Applications composites

Ce sont des applications constituées par assemblage d'un ensemble de composants de services [Rodriguez *et al.*, 07]. En d'autres termes, ces applications sont construites en composant plusieurs appels aux services.

Par exemple, une application permettant à un utilisateur de récupérer les informations techniques sur un élément d'une machine industrielle (figure 2.7). Dans ce cas, l'application utilise une « composition d'appels » vers les services suivants : « Inventaire machines » pour

la récupération de la référence de la machine, ensuite elle fait appel au service « fiche technique » afin d'afficher le schéma industriel de la machine.

L'utilisateur peut ainsi sélectionner l'élément requis et l'application affichera alors les détails techniques sur cet élément. La figure 2.8 illustre le diagramme de séquences correspondant.

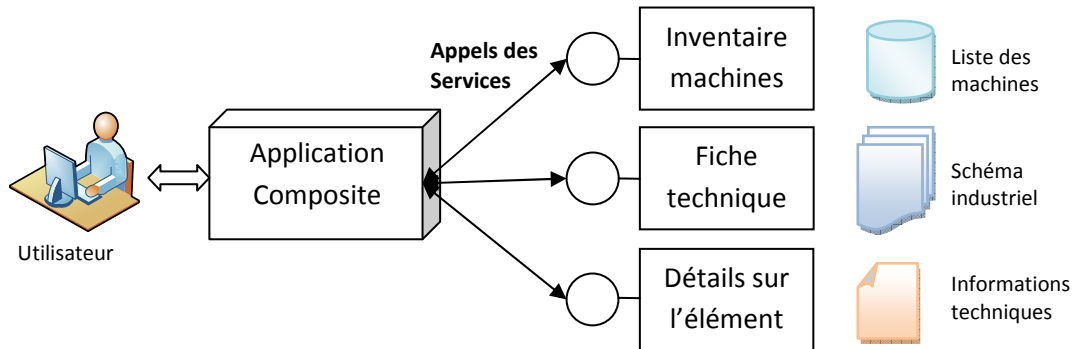


Figure 2.7 : Le principe des applications composites

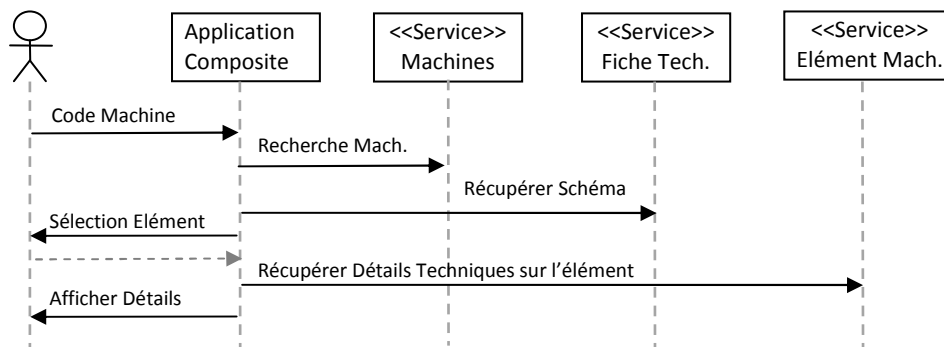


Figure 2.8 : Les interactions entre l'application composite et les services

2) Le moteur BPM (Business Process Management)

BPM (*Business Process Management*) ou la gestion des processus métier regroupe un ensemble de méthodes, outils et services qui permettent de modéliser, exécuter et optimiser les processus métier [Manouvrier et Ménard, 07].

Parmi ces outils, on trouve « le moteur BPM » qui instancie les processus suite à un événement déclencheur (ou stimulus), qui peut être par exemple la réception d'une demande d'intervention sur un équipement industriel. Il fait alors exécuter les différentes activités du processus en fonction des règles métier qui ont été définies.

La figure 2.9 illustre, dans le cadre de la maintenance industrielle, le déclenchement du processus « Ordre de travail (OT) » suite à la réception d'une demande d'intervention (DI) sur une machine.

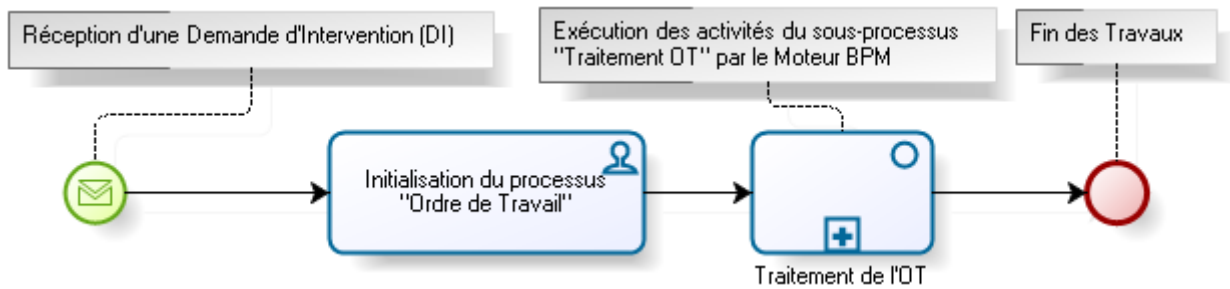


Figure 2.9 : Exemple d'exécution d'un processus métier par le moteur BPM

3) Bus de communication et d'intégration (ESB)

Pour bien mener les échanges de messages entre consommateurs et pourvoyeurs de services, il est nécessaire d'avoir une infrastructure d'échange de messages commune qui masquera l'hétérogénéité des divers systèmes communicant, c'est l'objectif du Bus de Message ESB : Entreprise Service Bus. Le bus doit supporter différents protocoles hétérogènes de communications, HTTP, FTP, .NET Remoting, etc. Il doit également permettre de déployer des services de technologies différentes telles que des Web Services, des composants .NET, ou encore des composants Java comme les EJB.

Le bus de message met également à la disposition des consommateurs de services deux modes d'appel de ces services : le mode synchrone et mode asynchrone. Dans le mode synchrone, lors de l'appel d'un service, le consommateur est bloqué jusqu'à ce que le service retourne une réponse et libère le consommateur pour procéder avec les activités suivantes. Tandis que dans le mode asynchrone, et lors de l'appel du service, le consommateur est libéré et n'attend pas que le service lui retourne une réponse. Le consommateur peut demander une garantie d'acheminement ou encore demander au service de lancer une opération dès que la réponse est obtenue.

4) Container de service ou SCA (Service Component Architecture)

C'est une spécification proposée par un groupe d'éditeurs, et qui a pour objectif de spécifier comment construire un service sur la base d'autres services. Pour cela, SCA permet de définir un mode d'assemblage des services en précisant la dépendance du service appelant vis-à-vis des autres. La spécification SCA⁸ définit aussi les méthodes d'invocation et d'encapsulation des services dans un langage XML, ainsi les échanges de messages peuvent être effectués indépendamment des protocoles de communication (SOAP, etc.) [Fournier-Morel *et al.*, 06].

5) Registre de services

L'ESB utilise un registre de service qui contient tous les contrats de service afin de vérifier que les messages échangés entre clients et fournisseurs sont conformes aux contrats.

6) Supervision des services et les processus métier [Manouvrier et Ménard, 07]

La supervision des différents services se fait à l'aide des outils baptisés **SAM** : *Service Activity Monitoring*. Ces outils permettent de surveiller et d'agir sur ses paramètres pour qu'il satisfasse les demandes des consommateurs et les contraintes des pourvoyeurs.

⁸ SCA : Service Component Architecture. Spécification d'OASIS, source <http://www.oasis-open.org/committees>.

Cela résume en suivi des erreurs, les performances, qualité de service et de la sécurité. La supervision des processus métier est assurée par des outils appelés **BAM** : Business Activity Monitoring. Les outils BAM se basent sur les fonctionnalités de SAM pour permettre de consulter l'état des processus, donner des statistiques sur l'utilisation des processus et gestion des alertes.

7) Accès aux référentiels

Les accès aux données contenues dans les différents référentiels de l'entreprise sont effectués par des services spécifiques appelés CRUD (*Create, Read, Update, Delete*) qui permettent de lancer les requêtes d'extraction ou de modification sur les sources de données.

La figure 2.10 illustre les différents composants cités précédemment d'une plate-forme SOA.

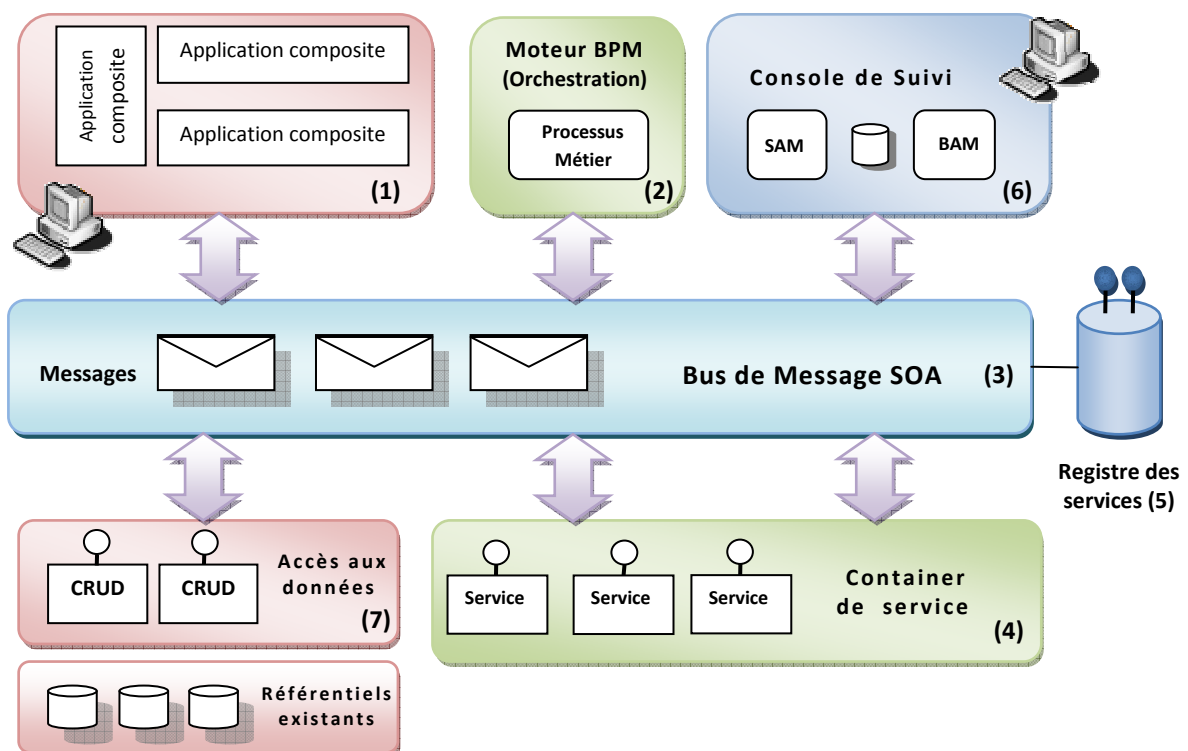


Figure 2.10 : Les composants d'une plate-forme SOA (d'après [Fournier-Morel et al, 06]).

III. Les services Web

Les architectures SOA ont l'avantage de permettre aux concepteurs et aux développeurs de construire leurs projets logiciels à base de Services implémentés dans diverses technologies, notamment les Services Web, CORBA (*Common Object Request Broker Architecture*) de OMG⁹, ou encore DotNet¹⁰ de Microsoft (msdn.microsoft.com/netframework/).

⁹ OMG : Organization Management Group (www.omg.org)

¹⁰ DotNet : Plateforme de Microsoft pour le développement des applications distribuées (msdn.microsoft.com/netframework/).

Néanmoins, la communauté SOA recommande l'utilisation des Services Web pour l'implémentation des différents services. Et cela grâce à leurs capacités en termes d'interopérabilité entre les composants logiciels hétérogènes et aussi la réutilisation des composants dans d'autres applications.

Il est important de souligner la séparation entre les deux notions : celle de SOA, qui représente un « style architecturale permettant d'émerger une couche Services au sein du SI », et celle des Services Web qui représentent un ensemble de technologies (protocoles et langages du Web) permettant de définir d'une manière standard l'interaction entre les applications à travers le Web.

Dans cette section, nous allons introduire le concept des Services Web ainsi que la pile de spécifications liées à cette technologie.

III.1. Concept et caractéristiques des Services Web

Un service Web est un composant logiciel représentant une fonction applicative (ou un service applicatif). Il peut être accessible depuis une autre application (un client, un serveur ou un autre service Web) à travers le réseau Internet en utilisant les protocoles de transports disponibles. Ce service applicatif peut être implémenté comme une application autonome ou comme un ensemble d'applications. Il s'agit d'une technologie permettant à des applications de dialoguer à distance via Internet, et ceci indépendamment des plates-formes et des langages sur lesquelles elles reposent. Pour ce faire, les services Web s'appuient sur un ensemble de protocoles standardisant les modes d'invocation mutuels de composants applicatifs [Bonnet, 05 b].

Les services Web sont caractérisés essentiellement selon les points suivants [Vasiliev, 07] :

Basés sur le Web (*Web based*): les services Web sont basés sur les protocoles et les langages du Web, en particulier le protocole de transfert HTTP et le langage de formatage XML.

Auto-Descriptifs et Autonomes (*Self-described, self-contained*) : le cadre des Web services contient en lui-même toutes les informations nécessaires à l'utilisation des applications, sous la forme de trois fonctions : trouver, décrire et exécuter. Ces fonctions sont réalisées respectivement à l'aide des technologies : UDDI, WSDL et SOAP qui seront présentées dans les paragraphes suivants.

Modulaires (*Modular*) : les Web services fonctionnent de manière modulaire et non pas intégrée. Cela signifie qu'au lieu d'intégrer dans une seule application globale toutes les fonctionnalités, on crée (ou on récupère) plusieurs applications spécifiques qu'on fait interopérer entre elles, et qui remplissent chacune une de ces fonctionnalités. Cela permet de développer sous forme Services Web des fonctions pouvant être réutilisées dans d'autres applications, et aussi de les composer avec d'autres fonctions pour former une nouvelle application, c'est le concept des applications composites.

III.2. Les standards des Web Services

La pile des Web Services se compose essentiellement en cinq couches [Fournier-Morel *et al.*, 06] qui sont les suivants (figure 2.11) :

- 1) Couche de l'infrastructure de base : regroupe les normes fondamentales : SOAP, WSDL et UDDI qui sont chargées respectivement de la communication, description et la découverte.
- 2) Couche regroupant des normes qui sont d'ordre technique. Ces normes couvrent les aspects liés à la sécurité, garantie d'acheminement et la gestion des transactions.
- 3) Couche pilotage et la supervision des services Web.
- 4) Couche qui regroupe toutes les spécifications sur la coordination et la composition de services.
- 5) Couche pour la représentation des services. Cette couche a pour rôle fondamentale d'agrèger les services au sein d'une interface commune. Une spécification a été définie par OASIS¹¹ appelée WSRP qui signifie *Web Services for Remote Portlets*. WSRP permet à un service d'associer aux messages SOAP un modèle de présentation qui pourra être consommé par un portail compatible WSRP [Chauvet, 02]. Cette spécification permet aussi la publication et la découverte des Portlets (fragment de page dans une interface portail) au travers d'un registre UDDI.

La figure 2.11 illustre ces différentes couches.

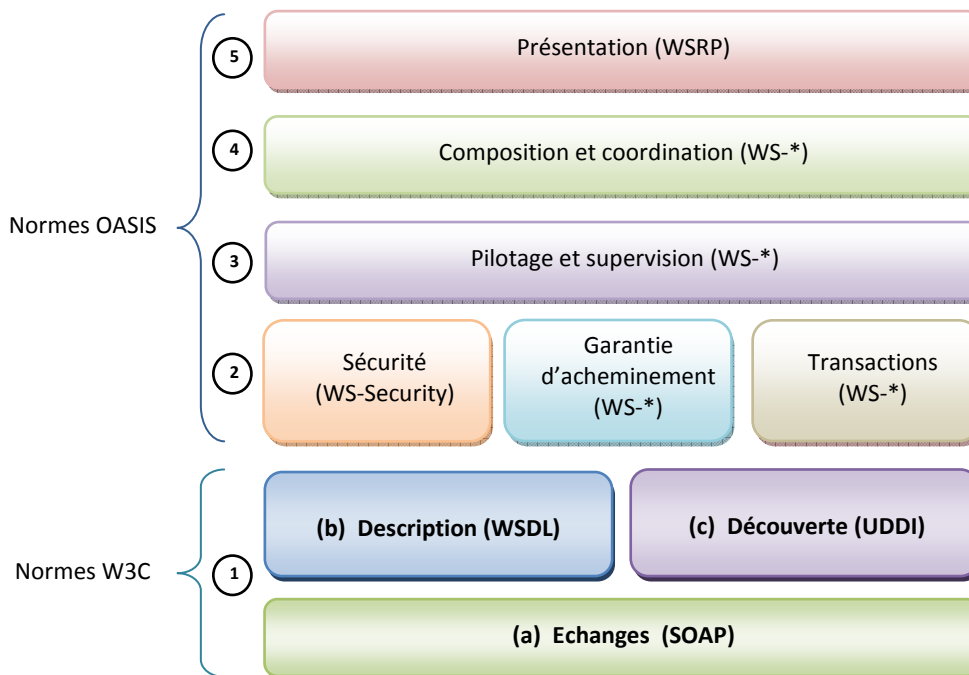


Figure 2.11 : La pile des Spécification des services Web [Fournier-Morel *et al.*, 06].

Pour chacune des couches une norme a été spécifiée. Le consortium W3C (*World Wide Web Consortium*), a normalisé SOAP, UDDI et WSDL qui forment les bases des services web.

¹¹OASIS : Organization for the Advancement of Structured Information (www.oasis.org)

Avec l'utilisation des services web de nouvelles spécifications plus complètes ont été rédigées par l'organisme de normalisation OASIS (*Organization for Advancement of Structured Information Standards*). Dû à l'intérêt porté par les différents éditeurs, de nouvelles spécifications ont été écrites qui n'ont pas toujours été soumises à l'organisme de normalisation afin d'être validées. C'est pourquoi il existe énormément de spécifications, souvent noté **WS-***, et que parfois des problèmes d'interopérabilités sont rencontrés.

Ce problème a été rapidement pallié par la création d'une nouvelle organisation, le WS-I (*WebServices Interoperability*) qui réunit les organismes de normalisations et les plus grands éditeurs du domaine afin de restreindre les libertés offertes par les spécifications.

Dans la suite de cette section, nous limitons notre étude sur les spécifications de base (c'est-à-dire SOAP, WSDL et UDDI) des services Web.

a) SOAP (Simple Object Access Protocol)

C'est un protocole basé sur le langage XML permettant la transmission de messages entre objets distants, ce qui veut dire qu'il autorise un objet à invoquer des méthodes d'objets physiquement situés sur un autre serveur. Le transfert se fait le plus souvent à l'aide du protocole HTTP, mais peut également se faire par un autre protocole, comme SMTP [Kadima et Monfort, 03] (une description sommaire est donnée en annexe A).

b) WSDL (Web Service Definition Language)

WSDL est un langage de description des services XML basés sur le réseau. Il permet de définir le format des objets, quel que soit le protocole (SOAP, XML) ou le codage utilisé par ces objets spécifiques à chaque système. Il décrit de manière abstraite et indépendante du langage de programmation, l'ensemble des fonctionnalités offertes par un service. Il permet de connaître les protocoles, les serveurs, les ports, le format des messages, les entrées, les sorties, les exceptions possibles et les opérations réalisées par un service web (une description sommaire est donnée en annexe A).

Les définitions WSDL facilitent donc l'application des services web en les rendant "auto-descriptifs". Autrement dit, le langage WSDL permet aux services web de décrire ce qu'ils font, comment ils le font et comment les clients peuvent les exploiter [Bonnet, 05 b].

c) UDDI (Universal Description, Discovery and Integration)

UDDI est un annuaire de services basé sur XML et plus particulièrement destiné aux services Web. L'annuaire de services UDDI est conçu comme un registre consultable, répertoriant les services web disponibles ainsi que leurs descriptions. UDDI permet ainsi de découvrir les services désirés de manière entièrement automatique (une description sommaire est donnée en annexe A).

III.3. Composition et coordination des Web Services

Orchestration de Services et la Chorégraphie

Les orchestrations se présentent comme un ensemble de mécanismes pour la construction d'un nouveau service (dit composite) dans une application composée de l'ensemble des

services atteignables. Au niveau des orchestrations sont définies des variables qui permettent de partager les informations entre les différentes invocations [Joffroy *et al.*, 07].

L'orchestration définit un service Web central pour orchestrer un processus de déroulement des opérations (appelé : *workflow*) au sein d'une entreprise comme par exemple : traitement d'un bon de commande, ajout des articles à un inventaire, et traitement d'une facture pour un client [Porter-Roth, 05]. La figure 2.12 illustre un exemple d'utilisation d'un orchestrateur.

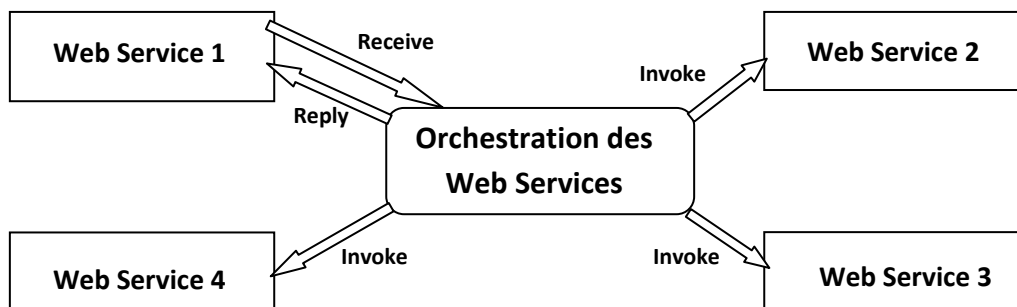


Figure 2.12 : L'orchestration de Services [Porter-Roth, 05].

Les orchestrations permettent de faire collaborer de manière efficace plusieurs Services Web, en définissant des Services Composites par assemblage de services. Ce formalisme est adopté par les industriels, à travers différents langages (BPEL, WSFL,...) et différentes plateformes (.NET, WebSphere, AXIS,...). Au sein d'une entreprise, elles sont un moyen de respecter les principes d'une architecture SOA autorisant une expression explicite des communications entre les Services Web.

Cependant, il existe un autre type de processus Workflow qui s'appelle la « chorégraphie » et qui est différent par rapport aux orchestrations, car il n'y a pas un service Web central pour orchestrer d'autres services, mais plutôt caractérisée par une dépendance faible entre les services. Ainsi ces derniers fonctionnent indépendamment les uns des autres mais d'une façon coordonnée [Joffroy *et al.*, 07].

La chorégraphie modélise la séquence des échanges de messages entre services web et conditions dans lesquelles ces messages sont échangés entre des clients, des fournisseurs et des partenaires. La chorégraphie est typiquement associée à l'échange de messages publics entre les services web, alors qu'un processus métier est exécuté de manière centralisé par un orchestrateur.

Un langage appelé WS-CDL (Web Services Choreography Description Language) a été spécifié et qui est un langage basé sur XML pour décrire les collaborations pair à pair entre les participants et l'échange de messages entre ces participants dans l'objectif d'accomplir un but commun dans la logique métier. La figure 2.13 illustre ce type de processus.

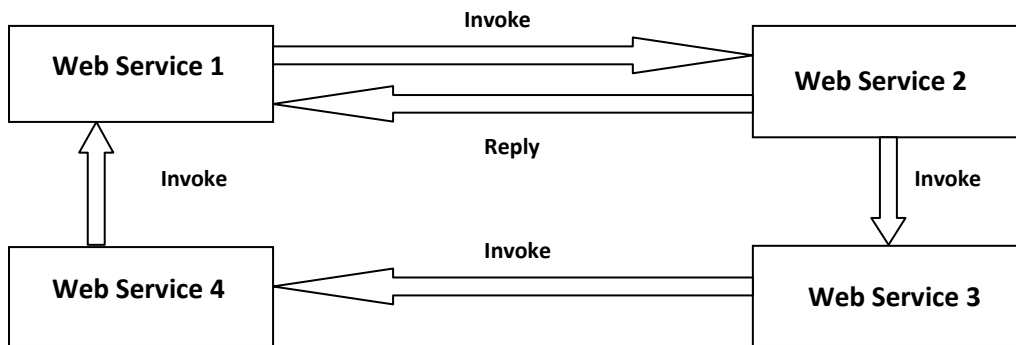


Figure 2.13 : La chorégraphie de Services [Porter-Roth, 05]

Dans la suite de cette partie, on présentera le langage de spécification BPEL (*Business Process Execution Language*) des processus métier, ainsi que la notation BPMN (*Business Process Modeling Notation*) utilisée dans la modélisation des processus métier.

BPEL (*Business Process Execution Language*)

C'est une spécification d'IBM, Microsoft, et BEA. Elle remplace les précédentes spécifications XLANG de Microsoft, et WSFL (*Web Services Flow Language*) d'IBM [Chauvet, 02]. Le modèle de processus BPEL forme une couche au-dessus de WSDL et qui basé sur le standard XML (*eXtensible Markup Language*). Il définit la coordination des interactions entre l'instance du processus métier et ses partenaires.

En outre, BPEL permet de décrire une ou plusieurs compositions (ou combinaisons) des services dans un processus et permet donc de fournir la description de l'orchestration de services dans le modèle de processus. Les processus dans BPEL exportent et importent les fonctionnalités en utilisant des interfaces de services web uniquement (la spécification du langage BPEL est donnée en annexe B).

BPMN (*Business Process Modeling Notation*)

Développé par le groupe BPMI (*Business Process Management Initiative*), le BPMN est une notation graphique permettant de modéliser les processus métier à l'aide d'éléments graphiques et de diagrammes en séparant les informations métiers des informations techniques [Manouvrier et Ménard, 07].

Le BPMN peut être comparé à UML, mais pour la gestion des processus. En ayant un standard, cela permet à différents éditeurs de créer des outils permettant d'utiliser une notation commune pour modélisation des processus métiers et de permettre une interopérabilité entre différentes applications, de la modélisation à l'exécution des processus. Les diagrammes sont constitués d'éléments simples compréhensibles par les analystes métiers (la spécification de la notation BPMN est donnée en annexe C).

IV. Aspects méthodologiques liés à l'approche orientée services

Dans les approches de développements traditionnels d'applications, les concepteurs ont souvent une vision de découpage verticale du SI global. Ce découpage vise à traiter chaque silo applicatif de façon isolé comme déjà évoqué dans le chapitre 1 (section §IV.1). Par conséquent, il est souvent difficile d'intégrer ces applications monolithiques et donc de les faire « coopérer » ou encore de créer une « collaboration » entre ces différentes applications.

Par contre dans l'approche orientée services (SOA), les différentes applications du SI global sont vues comme des composants de services capables de fournir des services pour d'autres applications et acteurs métier (utilisateurs) et aussi de pouvoir être composées ou réutilisées pour construire d'autres services afin de répondre aux nouveaux besoins du métier.

Par ailleurs, SOA propose aux architectes des SI une vision de modélisation par couches d'abstraction entre le monde métier (processus et fonctions métier) et le monde technique qui représente les différentes technologies sous-jacentes (applications, réseaux, etc.) [Rivard *et al.*, 05].

Le lien entre ces deux mondes est concrétisé par l'émergence d'une couche de services (figure 2.14) qui répertorie les fonctions métier sous-jacentes sous forme de services [Fournier-Morel et al, 06].

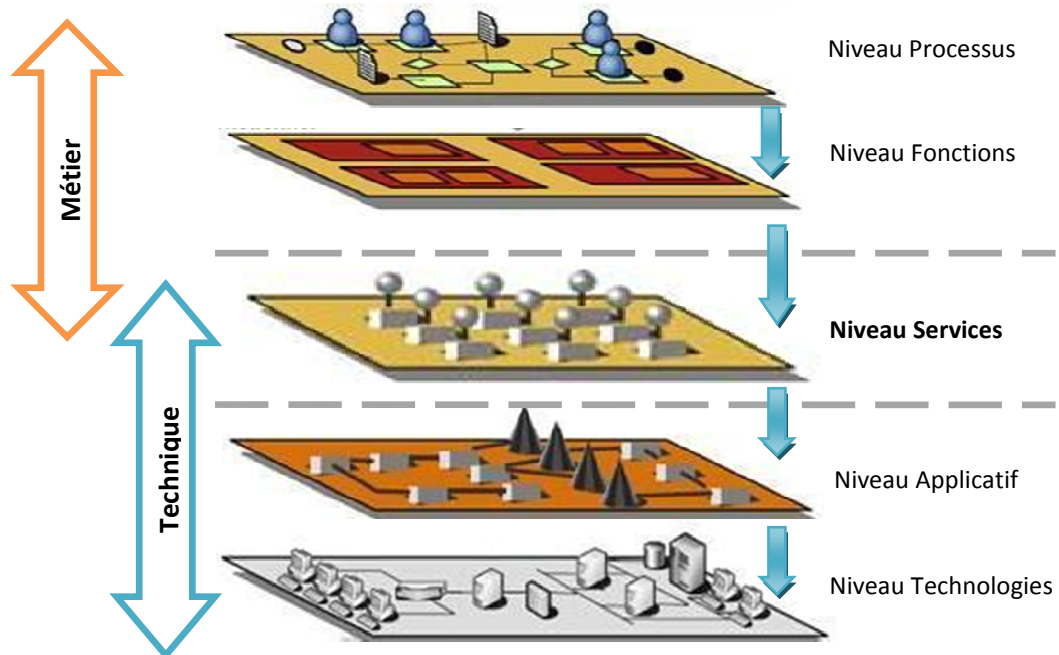


Figure 2.14 : L'Approche SOA : liaison entre vue métier et vue technique [Fournier-Morel et al, 06]

Enfin, l'approche orientée services se distingue des autres par la mise en avant du métier plutôt que de guider l'architecture par la technologie [Rivard *et al.*, 05].

Dans la suite de cette section, nous mettons l'accent sur des éléments méthodologiques fondamentaux liés à l'approche orientée services et qui pourront être intégrés dans une démarche d'analyse et de modélisation d'un système de maintenance industrielle.

IV.1. Analyse et modélisation orientées services

Dans cette étude nous employons souvent les deux termes : analyse et modélisation. En effet, l'analyse implique la description du système à développer à travers le plus haut niveau d'abstraction qui est le niveau conceptuel [Portier, 07]. Dans l'approche SOA, la phase d'analyse doit avoir en entrée un ensemble de besoins métiers et aussi les constituants du SI existant (applications, documents, etc.) et comme sortie une description de ce qu'on doit construire, c'est-à-dire le future système, et qui servira comme entrée pour la phase de modélisation. Par ailleurs, la modélisation implique la description détaillée du système qu'on doit construire et surtout comment le construire [Portier, 07].

Comme nous l'avons souligné, SOA met en avant l'émergence d'une couche de services. Ces derniers caractérisés par leur typologie (métier ou technique), leur granularité, leur composition, etc. Par conséquent, il est nécessaire de considérer tous ces points en vue de la conception d'approche orientée services [Idoughi, 08]. En effet, cela peut être traité lors de la modélisation des services qui selon [Arsanjani, 04] et [Zimmerman *et al.*, 04] doit passer par trois étapes essentielles : (1) l'identification des services, (2) spécification des composants et interfaces de services et (3) la réalisation. Ainsi, le modèle de services obtenu lors de la modélisation représentera un élément méthodologique clé dans l'approche SOA.

IV.2. Modélisation métier

Nous avons évoqué précédemment que la démarche orientée services met en avant une vision métier plutôt qu'une vision technique. Par conséquent, l'approche orientée services se focalise dans sa phase de modélisation sur les processus métier. Ces derniers décrivent l'ensemble des activités que l'entreprise doit mener pour traiter un événement métier (demande de fournitures, demande d'intervention, etc.) [Fournier-Morel *et al.*, 06].

Cependant, ces processus métier peuvent couvrir plusieurs domaines fonctionnels de l'entreprise tel que, gestion des ressources humaines, gestion des stocks, et peuvent encore s'étendre au delà des frontières du SI de l'entreprise vers d'autres organisations externes (fournisseurs de pièces, sous-traitants, etc.) [Idoughi, 08]. Ainsi, chaque processus transverse (impliquant plusieurs départements de l'entreprise et aussi des partenaires externes) représente une collaboration (ou orchestration) entre plusieurs unités et acteurs afin de délivrer une valeur ajoutée pour l'entreprise [Raymond, 07].

Par ailleurs, la modélisation des processus préconise trois éléments essentiels pour la définition d'un processus métier [Fournier-Morel *et al.*, 06] : (1) l'événement métier qui représente le déclencheur (appelé aussi catalyseur) du processus ; (2) les activités qui doivent être exécutées selon des règles métiers et enfin (3) les acteurs (humains ou systèmes) qui sont impliqués dans l'exécution des activités du processus.

Enfin, les enchainements d'activités, les règles métiers et les éléments associés à un processus métier peuvent être pris en compte lors de la modélisation des processus métier.

V. Conclusion

L'approche orientée services offre un nouveau modèle permettant de construire des SI homogènes, évolutifs et rapidement adaptables face aux changements du domaine métier. La démarche SOA propose des solutions pour répondre aux exigences liées à l'intégration des applications hétérogènes et à la valorisation des applications existantes en externalisant certaines de leurs fonctions applicatives sous forme de services. Cette intégration est devenue possible grâce à la gestion des processus métier transverses des différents silos applicatifs du SI de l'entreprise.

Par ailleurs, la réussite d'un projet SOA dépend de plusieurs facteurs, notamment dans la démarche suivie et aussi de la technologie utilisée. De ce fait, l'approche SOA préconise l'utilisation des services Web comme une technologie capable de fournir la réutilisation, le découplage et l'interopérabilité entre les différents constituants (services) de l'architecture.

Enfin, comme déjà souligné, SOA est avant tout une façon de concevoir des systèmes informatiques complexes. Ainsi, la mise en œuvre d'un projet utilisant l'approche orientée services doit suivre une démarche d'analyse et de modélisation adéquate.

C'est pourquoi le troisième chapitre étudie un ensemble de démarches SOA en vue d'identifier et de mettre en avant les briques de base pour proposer une architecture orientée services d'un système de maintenance industrielle.

CHAPITRE III : VERS UNE APPROCHE D'ANALYSE ET DE MODELISATION ORIENTEE SERVICES POUR LES SYSTEMES DE MAINTENANCE

I. Introduction

Dans le chapitre 1, nous avons expliqué que la fonction de maintenance a connu beaucoup d'évolutions ces dernières années notamment sur les politiques (ou formes) de maintenance appliquées (prévisionnelle, conditionnelle, etc.) ou encore sur le plan organisationnel (télémaintenance, e-maintenance, etc.). Par ailleurs, nous avons évoqué que ces évolutions ont un impact sur l'architecture du système de maintenance qui est passé du statut centralisé vers décentralisé à travers notamment de la maintenance distribuée.

Par conséquent, les systèmes de maintenance fonctionnent désormais dans un environnement distribué et caractérisé par son dynamisme permanent (changement d'organisation interne à l'entreprise, nouveaux partenaires, nouvelles installations, etc.). Ajoutant à cela, l'hétérogénéité des différentes applications impliquées dans la maintenance industrielle, et cela en termes de technologies (langage d'implémentation, systèmes d'exploitation, etc.), et aussi sur le mode de communication (en temps réel pour les systèmes SCADA, systèmes transactionnels tel que la GMAO).

La nouvelle problématique est donc de construire des systèmes de maintenance capable d'intégrer l'ensemble des applications et systèmes dits hétérogènes et de supporter les changements liés d'une part aux SI global de l'entreprise, l'environnement externe (ou l'écosystème) à travers les SI des partenaires d'autre part.

Pour répondre à de nouveaux besoins en termes d'intégration et d'agilité, nous avons choisi l'approche orientée services (SOA), présentée dans le chapitre 2, qui représente aujourd'hui un cadre de développement le plus approprié pour construire des SI évolutifs, rentables et aussi alignés par rapport aux objectifs métier.

Cependant, l'adoption d'une approche orientée services dans tout projet de conception, notamment dans le cadre des systèmes de maintenance industrielle, suggère l'application d'une démarche efficace qui permettra d'aboutir aux objectifs métiers. Par ailleurs, nous constatons qu'il n'existe pas dans le domaine de l'orienté services une démarche de référence pour l'analyse et la modélisation orientée services. En effet, plusieurs méthodes ont été proposées par des acteurs potentiels du monde SOA, notamment par IBM avec la méthode « SOMA » (*Services Oriented Modeling and Architecture*) [Arsanjani, 04], « Praxème » de *Unilog* et *OrchestraNetworks* [Vauquier et Bonnet, 06], ou encore à travers des travaux de recherche scientifique notamment celle proposée dans [Idoughi, 08] pour les systèmes de supervision industrielle à base des services Web.

Soulignons enfin que malgré cette relative diversité des approches proposées, nous constatons que toutes ces méthodes s'accordent sur la nécessité de mettre en avant les aspects du métier par rapport à ceux de la technologie.

Dans ce chapitre, nous présentons deux de ces approches représentatives issues du SOA, notamment celle évoquée dans [Idoughi *et al.*, 08] et la méthode SOMA d'IBM [Arsanjani, 04 ; Zimmerman *et al.*, 04]. Par la suite, nous allons faire une synthèse

de ces approches afin d'aboutir à une démarche d'analyse et de modélisation orientée services qui s'inscrit dans un cadre méthodologique pour la conception des systèmes de maintenance industrielle.

II. Méthode 1 : Approche orientée services d'IHM pour la supervision industrielle [Idoughi, 08]

Dans cette approche, l'auteur définit un cadre méthodologique global relatif à la conception d'IHM¹² de supervision orientée services. La démarche proposée s'appuie d'une part sur les principes d'une conception centrée utilisateur, et d'autre part sur les aspects méthodologiques de développement des services Web.

L'intérêt marquant de ce cadre est de rapprocher les modèles issus du GL (Génie Logiciel) d'une part, des technologies et ingénierie du Web et la gestion des processus métier d'autre part. Ainsi, l'approche vise à intégrer le développement orienté services sous les angles du GL, de l'IHM tout en ayant une vision métier.

L'approche propose trois grandes phases (figure 3.1) : (1) étude de l'organisation globale de supervision et analyse métier ; (2) analyse et expression des besoins et (3) spécification et conception de l'IHM de supervision à base des services Web.

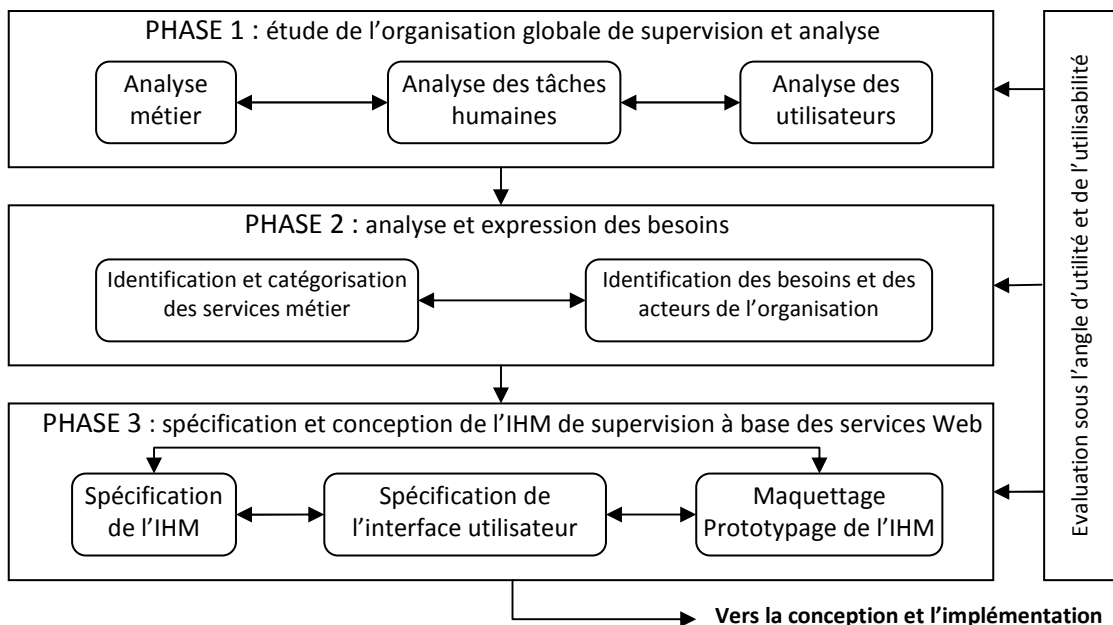


Figure 3.1 : Les trois phases de l'approche [Idoughi, 08]

¹² **IHM : Interface Homme-Machine** ou **Interaction Humain-Machine** étudie la façon dont les humains interagissent avec les ordinateurs ou entre eux à l'aide d'ordinateurs, ainsi que la façon de concevoir des systèmes informatiques qui soient ergonomiques, c'est-à-dire efficaces, faciles à utiliser ou plus généralement adaptés à leur contexte d'utilisation. Source wikipedia (http://fr.wikipedia.org/wiki/Interface_Homme-machine).

Dans la suite de cette section, nous décrivons brièvement ces phases ainsi que leurs étapes.

II.1. Phase 1 : Etude de l'organisation globale de supervision et analyse métier

Elle consiste en une étude préalable de l'organisation complexe globale existante visant l'identification du problème, des objectifs relatifs au métier et au système homme-machine global afin d'établir une vision globale de la situation, des principales exigences, processus clés et contraintes ainsi que le recensement des principaux risques (organisationnels, techniques, etc.). Dans cette phase, on trouve trois étapes essentielles (figure 3.1) : (1) analyse métier ; (2) analyse des tâches humaines et (3) analyse des utilisateurs.

Etape 1 : Analyse métier

Cette étape a pour objectif de comprendre la nature et le contexte de l'environnement métier au sein de l'organisation. L'analyse métier comprend trois aspects : (1) L'identification des processus métier donnant de la valeur ajoutée à l'organisation en liaison avec le domaine métier. Cette identification donne lieu à un ensemble de services métier. La figure 3.2 illustre le processus d'identification des services métier.

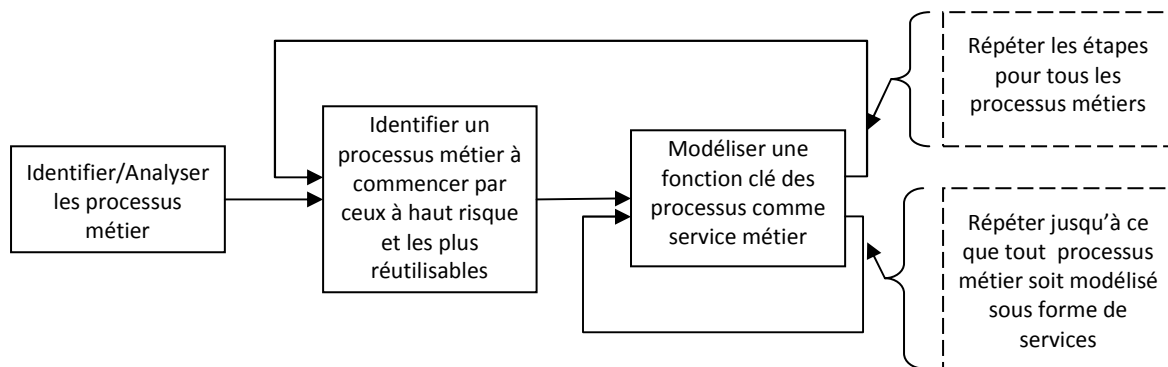


Figure 3.2 : Le processus d'identification des services métier [Idoughi, 08]

(2) L'analyse et modélisation de processus métier qui consiste à décrire la succession des activités du processus et le contenu de chaque activité : rôle de chaque acteur, les données qu'il manipule, etc. La modélisation métier doit s'appuyer sur des « scénarios métier » afin de pouvoir déduire ou extraire des définitions fonctionnelles. Le modèle de description d'un scénario pour un processus métier est donné sur la figure 3.3.

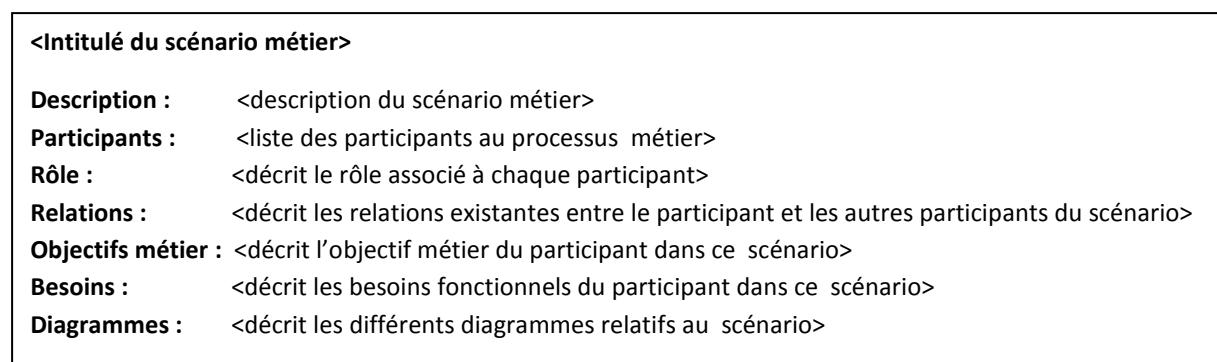


Figure 3.3 : Description globale d'un scénario d'un processus métier [Idoughi, 08]

(3) L'étape suivante est d'extraire les cas d'utilisation sous forme de spécifications fonctionnelles du plus haut niveau d'abstraction (1^{er} niveau) en termes de services métier. Ainsi, on obtient un « modèle de cas d'utilisation métier » exprimant l'usage métier des services Web à partir des scénarios métier (figure 3.4).

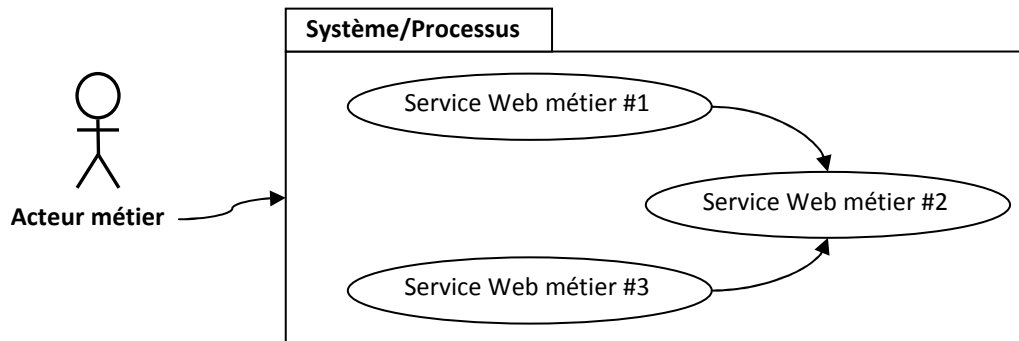


Figure 3.4 : Exemple d'usage métier des services Web [Idoughi, 08]

A l'issue de cette première étape (analyse métier), on aura une décomposition en couches distinctes d'un processus métier dans le cadre d'une SOA comme le montre la figure 3.5.

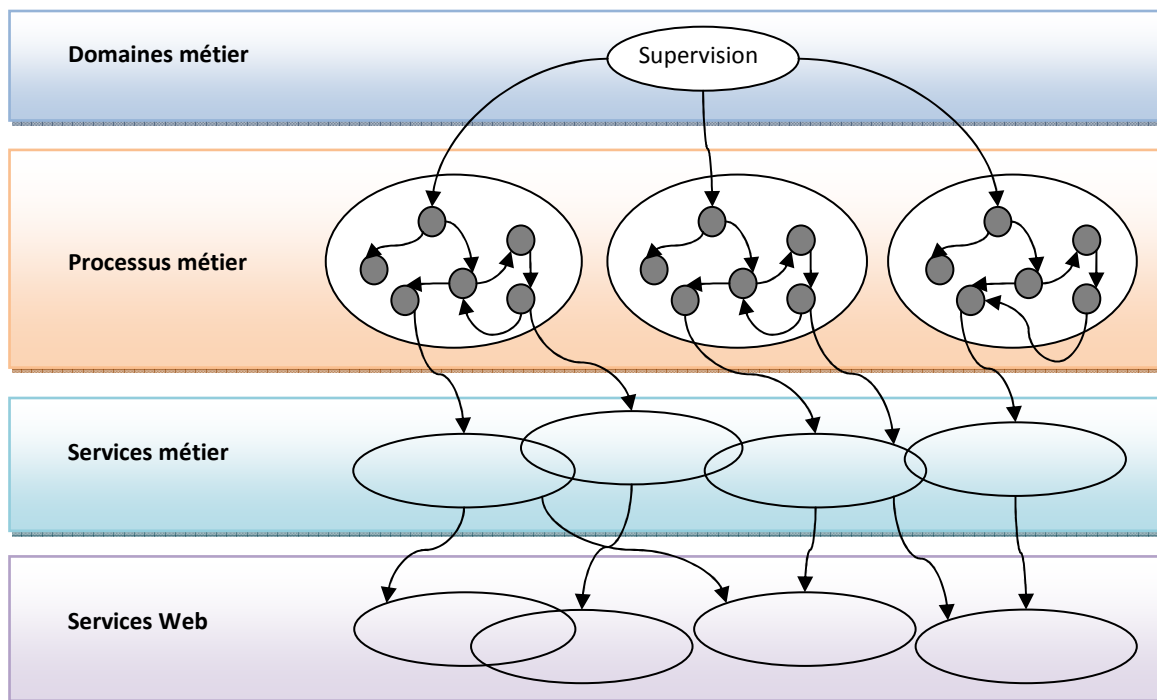


Figure 3.5 : Vue métier de la supervision dans le cadre de SOA [Idoughi, 08]

Etape 2 : Analyse des tâches

Cette étape permet de mettre en évidence la décomposition du système homme-machine global selon différents sous-systèmes ou processus distincts et plus simple à considérer. La décomposition du système homme-machine global s'effectue en plusieurs sous-systèmes hiérarchiquement fonctionnels reflétant ainsi l'organisation complexe (figure 3.6). Pour le découpage fonctionnel, on a recours à une méthode cartésienne telle que SADT.

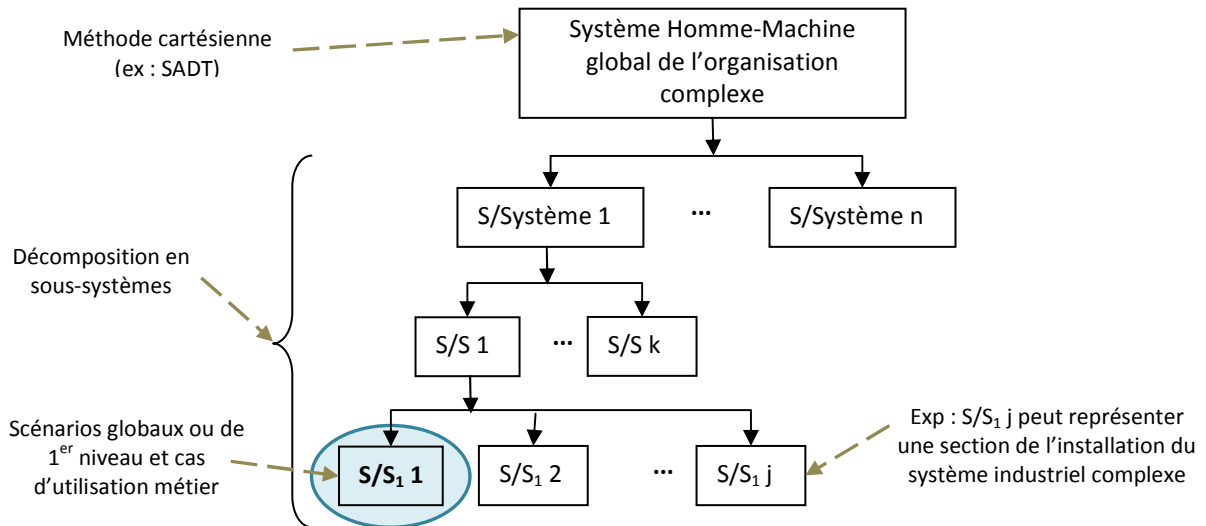


Figure 3.6 : Décomposition hiérarchique d'un système homme-machine global [Idoughi, 08]

A la suite de cette décomposition, on obtient un ensemble de modèles groupant les scénarios et les cas d'utilisation métier du 1^{er} niveau d'abstraction.

Etape 3 : Analyse des utilisateurs

Cette étape conduit à l'identification des principaux acteurs métier impliqués dans les scénarios métier construits lors de l'analyse métier et des tâches. Au cours de cette étape, les profils des utilisateurs potentiels de la future IHM doivent être identifiés, en recueillant des données et informations pertinentes les concernant.

Enfin, la première phase permet donc d'avoir un ensemble de processus métier, cas d'utilisation métier issus des scénarios d'usage des services Web, ainsi que les services métier et les acteurs métier impliqués dans le processus métier.

II.2. Phase 2 : Analyse et expression des besoins

Dans cette phase, on utilise l'ensemble des cas d'utilisation métier en lien avec les principaux acteurs métier impliqués dans l'organisation existante et exprimant les besoins métier vis-à-vis de l'IHM, une 2^{ème} analyse de ces besoins pour capturer d'avantage d'autres besoins manquants à l'aide des services Web vis-à-vis de la mobilité des acteurs concernés et des tâches coopératives. Deux étapes sont définies dans cette phase :

Etape 1 : Identification et catégorisation des services Web métier

Cette étape vise à catégoriser les services Web métier déjà identifiés lors de la première phase et d'identifier les services Web candidats exprimant une ou plusieurs opérations. Cela donne lieu à un « modèle de cas d'utilisation de services métier » contenant des fonctionnalités métier et qui sont exprimées sous forme de packages (figure 3.7).

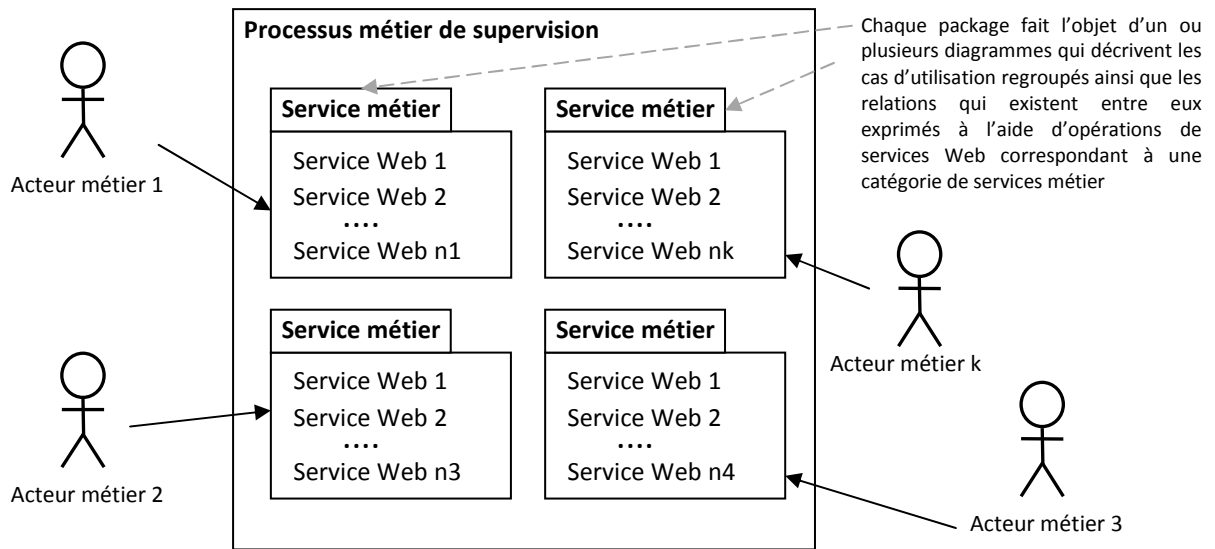


Figure 3.7 : Modèle de cas d'utilisation de services métier [Idoughi, 08]

Par ailleurs, cette étape met en évidence les liens possibles entre les services métier et les services relatifs à leurs représentation visuelle, donc leurs IHM (figure 3.8).

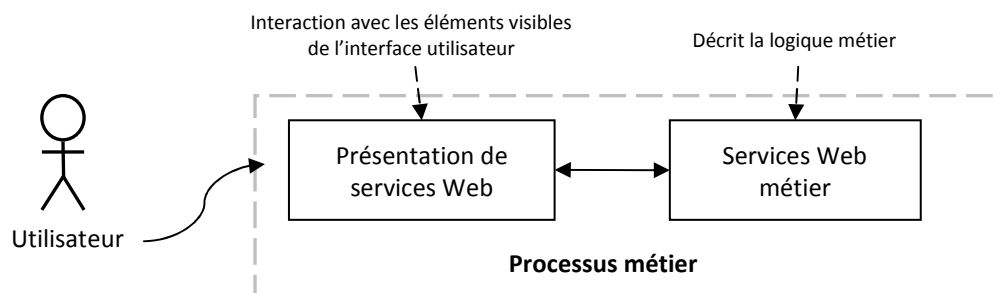


Figure 3.8 : Différents services Web d'un processus métier [Idoughi, 08]

Etape 2 : Identification des besoins et des acteurs de l'organisation

Cette étape consiste à (1) exprimer les besoins identifiés en termes de services Web d'une part et de capter d'autres besoins vis-à-vis (2) de la mobilité et (3) des tâches coopératives des acteurs tout en identifiant aussi d'autres acteurs impliqués dans des scénarios de mobilité et de coopération de tâches.

L'expression des besoins en termes de services Web passe d'abord par l'expression des scénarios sous forme de cas d'utilisation et pour chaque cas sélectionné, on considère chaque enchaînement ou action du scénario comme étant des éléments pouvant définir un service Web candidat. Ainsi les messages échangés correspondant aux opérations de service Web pouvant composer les services métier interagissant avec les différents composants de l'IHM. Notons que ce processus est appliqué pour chaque package de services métier identifié.

La figure 3.9 illustre la démarche d'expression des besoins en termes de services Web.

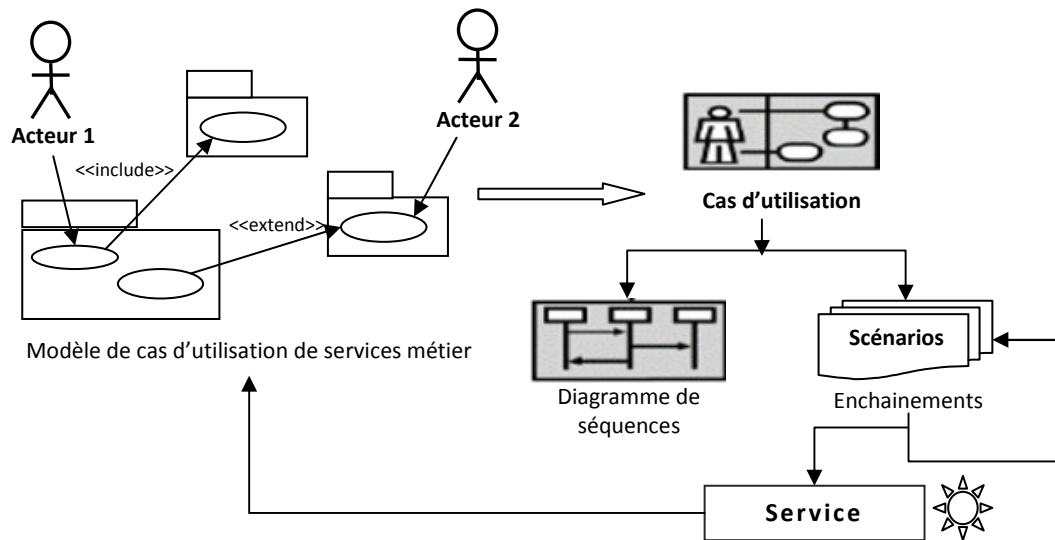


Figure 3.9 : Processus d'expression des besoins à l'aide de services Web [Idoughi, 08]

L'expression des besoins vis-à-vis de la mobilité des acteurs humains passe par les étapes suivantes : (1) identification des scénarios potentiels issus du domaine métier pour lesquels des acteurs humains sont nomades [Idoughi et Kolski, 06 b], ou ceux susceptibles de le devenir; (2) pour chacune des classes des scenarios identifiés, on procède à l'identification ou à extraire d'autres besoins sous forme de nouvelles fonctionnalités pouvant être exposées sous forme de services Web. (3) La dernière étape consiste à caractériser le contrat d'utilisation de chaque service Web en décrivant à l'aide d'un ensemble d'informations tel que : classe de scénario, profil utilisateur, etc.

L'expression des besoins vis-à-vis des tâches coopératives des acteurs humains permet d'exprimer des tâches de supervision sous forme coopérative en termes de services Web. La démarche à suivre comprend globalement deux étapes : (1) l'identification des scénarios potentiels du domaine métier pour lesquels des acteurs humains de l'organisation étudiée exécutent leurs tâches d'une façon coopérative ; (2) La seconde étape consiste à l'expression de ces tâches coopératives à l'aide des services Web. Les services Web obtenus sont caractérisés par leur granularité forte du fait qu'il représente une composition d'autres services Web de granularité inférieure. Ces derniers sont orchestrés pour la réalisation des tâches métier.

II.3. Phase 3 : Spécification et conception de l'IHM

Cette phase a pour objectif de spécifier et de concevoir des IHM de supervision où les fonctionnalités contenant les traitements métier sont exposées sous forme de services Web. Trois étapes sont définies dans cette phase : (1) spécification de l'interface homme-machine dans le cadre d'un processus métier ; (2) spécification de l'interface utilisateur et (3) la dernière étape consiste au maquettage et prototypage de l'IHM.

Etape 1 : Spécification de l'interaction homme-machine

Dans cette étape, on décrit à l'aide d'une spécification des services Web, l'ensemble du déroulement ou enchaînement des différents services composants un processus métier de

supervision. Plus précisément, il s'agit de considérer les aspects de spécification relatifs aux services web concernant la communication entre différents services web (envoi et réception de messages), leurs tâches internes (transformation, manipulation de données et appel à des applications), les règles de dépendance entre les tâches (séquence, choix, traitements conditionnels, etc.), en plus de la gestion des événements et exceptions pouvant exister lors de l'exécution, etc.

Etape 2 : Spécification de l'interface utilisateur

Cette étape consiste à décrire les objets d'interaction (boutons, listes déroulantes, etc.) correspondant aux différents services Web métier à l'aide d'une spécification permettant de créer des interfaces utilisateurs pour les services Web orientés présentation. La figure 3.10 illustre les niveaux de spécification de services Web orientés présentation.

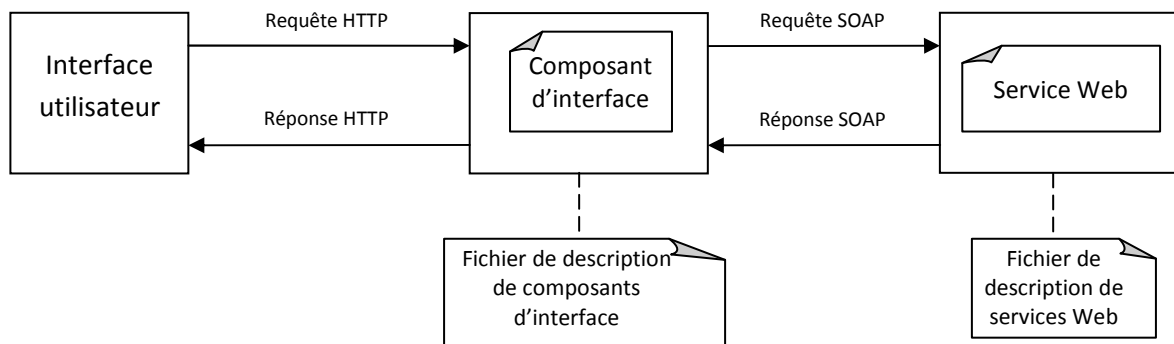


Figure 3.10 : Les niveaux de spécification de services Web orientés présentation [Idoughi, 08]

Etape 3 : Maquettage et prototypage de l'IHM

L'objectif de cette étape est de présenter un modèle d'interface utilisateur qui soit le plus proche de la réalité, sous forme de maquettes (puis de prototypes), ceci afin de bien visualiser les interfaces, les tester, les commenter, simuler leur utilisation et proposer des améliorations éventuelles tout en se rattachant aux scénarios de supervision.

La figure 3.11 illustre à travers un organigramme le processus de maquettage et prototypage de l'interface utilisateur.

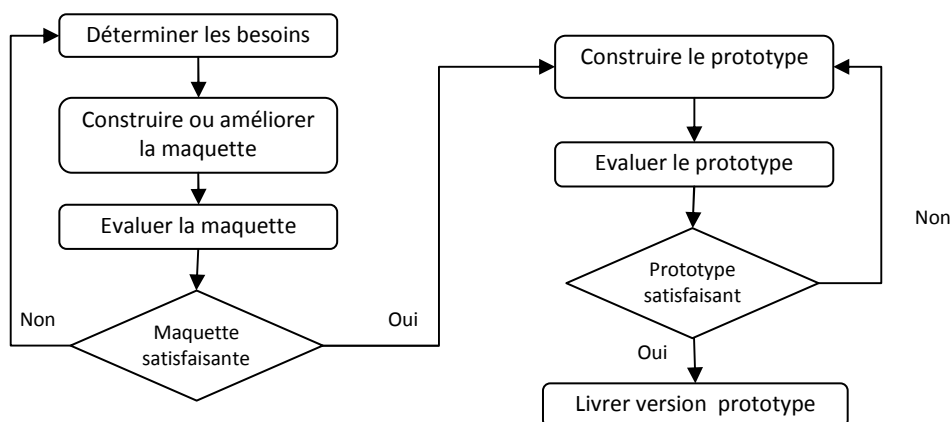


Figure 3.11 : Le processus de maquettage et prototypage de l'interface utilisateur [Idoughi, 08]

II.4. Conclusion sur la méthode 1

La première méthode présentée est relative au domaine de l'interaction homme-machine (IHM) et qui permet de définir un cadre méthodologique en vue de la spécification des systèmes interactifs dans les organisations complexes.

La démarche propose des éléments importants qu'on peut exploiter dans l'analyse et la modélisation orientées service d'un système de maintenance, notamment dans les phases préliminaires d'analyse métier, les acteurs intervenants et aussi la spécification des interactions d'une part entre les acteurs du métier et le système informatique à travers le processus métier, et entre les services Web orientés présentation et le processus métier d'autre part.

Comme nous l'avons souligné, cette démarche a été initialement destinée à être appliquée dans le domaine de la supervision industrielle. Cependant, vue la similitude et la relation qui existent entre la maintenance et la supervision, on peut considérer donc certains éléments pertinents qui peuvent nous servir dans l'analyse et la modélisation orientée services des systèmes dédiés à la maintenance industrielle. La section suivante est consacrée à une deuxième méthode nommée SOMA (*Services Oriented Modeling and Architecture*) proposée par IBM dans le cadre de la modélisation orientée services des systèmes distribués.

III. Méthode 2 : SOMA (*Services Oriented Modeling and Architecture*)

Vers la fin de l'année 2004, les travaux de recherche menés par une équipe d'architectes de la société IBM ont abouti à une nouvelle méthode pour l'analyse et la modélisation orientée services des systèmes d'information baptisée « SOMA » ou Architecture et modélisation orientée services (de l'anglais *Services Oriented Modeling and Architecture*) [Zimmerman *et al.*, 04 ; Arsanjani, 04]. Cette méthode définit un nouveau cadre méthodologique basé sur des disciplines déjà existantes notamment l'approche orientée objet OOAD (*Object Oriented Analysis and Design*), les outils de conception (Framework) d'architectures d'entreprise EA (*Enterprise Architecture*) et la gestion des processus métier BPM (*Business Process Modeling*) déjà vu dans le chapitre 2 (§II.2). Les détails sur les approches EA, OOAD et BPM sont donnés en annexe (annexe D).

L'idée du rapprochement entre ces trois domaines (OOAD, EA et BPM) est fondée sur le constat suivant : « aucune de ces méthodes nous permet de couvrir d'une part tous le cycle de vie du projet SOA (c'est-à-dire, analyse, conception et développement) et aussi les domaines du projet (métier, architecture et application) ».

En effet, malgré que la méthode orientée objets (OOAD) offre un cadre méthodologique très intéressant pour SOA, notamment sur l'encapsulation des objets, l'héritage, polymorphisme, etc., elle ne couvre cependant que le niveau d'abstraction le plus bas dans l'architecture à travers les classes et objets.

Quand au Framework EA, il ajoute au dessus des solutions architecture un plan d'urbanisation du système, mais n'offrent pas la possibilité d'une part d'identifier les services et d'une autre part la réutilisation des services.

En fin pour les approches BPM, elles permettent d'avoir une vue de bout-en-bout pour les unités fonctionnelles du métier, mais ne couvrent pas les domaines architecture et implémentation du cycle de développement. Par ailleurs, il est important à souligner qu'aucune des approches citées ne montre comment valoriser et réutiliser les systèmes existants (appelés aussi applications légataires).

Comme illustré sur la figure 3.12, la méthode SOMA a donc pour vocation de fédérer toutes les approches citées précédemment tout en rajoutant une nouvelle dimension, celle du service à travers de nouveaux concepts (orchestration de services, le bus d'intégration ESB, découverte dynamique de services par UDDI, etc.) [Gustienne et Gustas, 07].

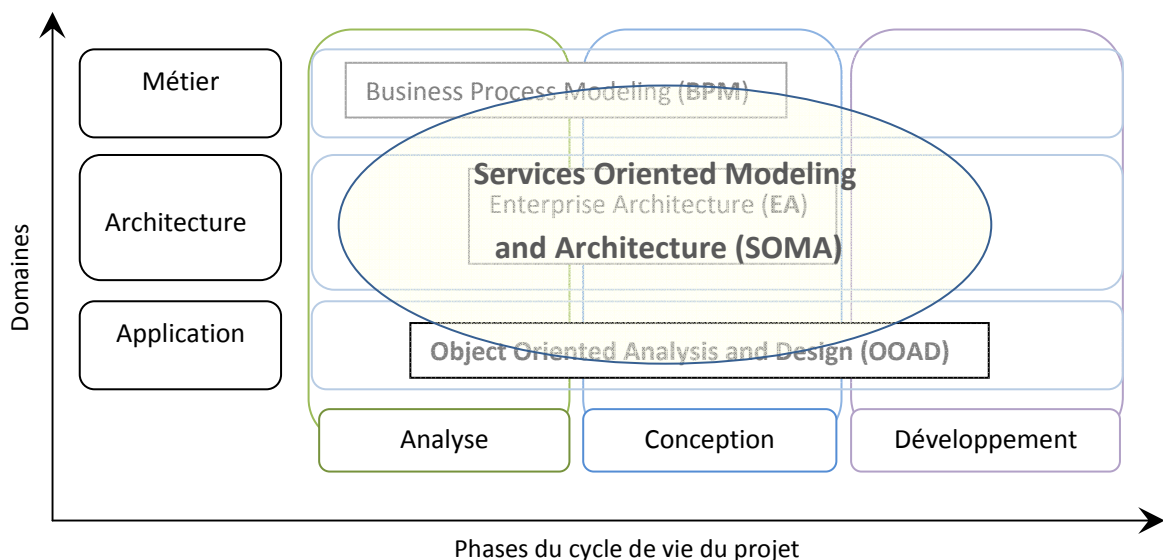


Figure 3.12 : SOMA par rapport aux approches OOAD, EA et BPM [Zimmerman *et al.*, 04].

L'approche SOMA définit trois grandes phases : (1) identification ; (2) spécification et (3) réalisation. La première phase correspond au domaine d'analyse tandis que les deux autres (spécification et réalisation) sont considérées dans un cadre de modélisation orientée services [Portier, 07].

Dans la suite de cette section, nous décrivons ces trois phases ainsi que les étapes impliquées dans chaque phase (figure 3.13).

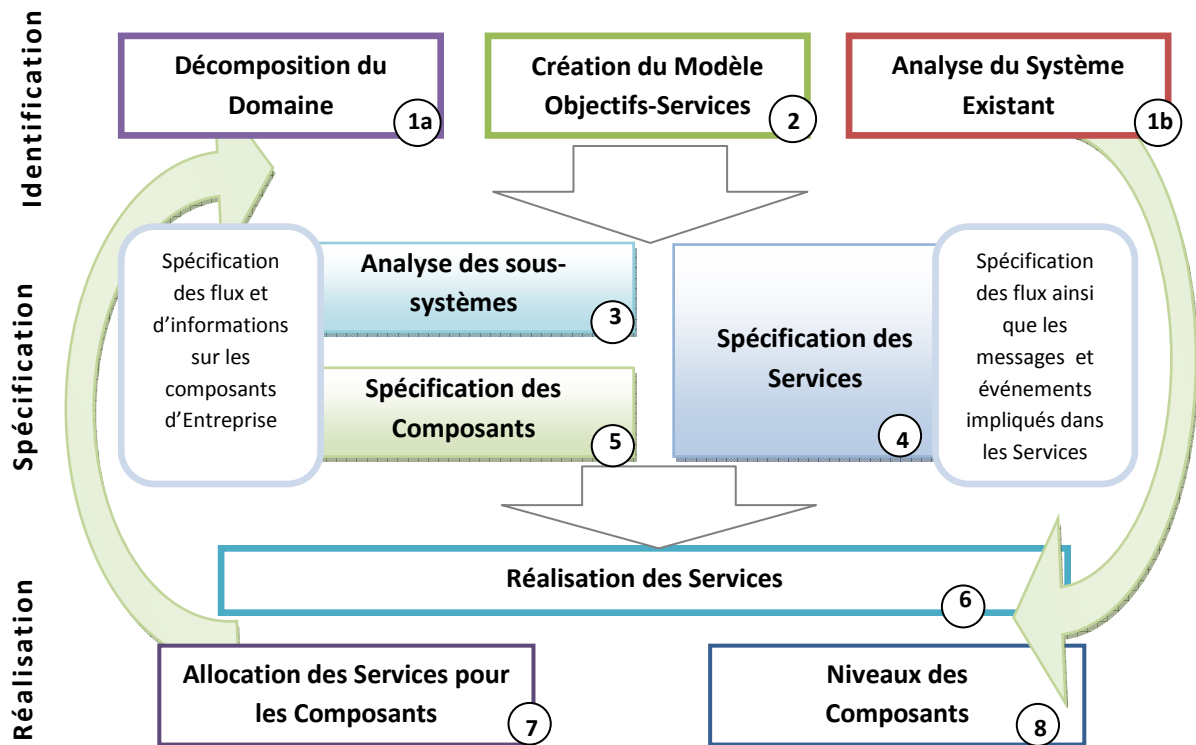


Figure 3.13 : Démarche SOMA pour l'analyse et modélisation orientée services [Arsanjani, 04].

III.1. Phase 1 : Identification

Cette phase commence par l'application de trois techniques complémentaires [Ganci *et al.*, 06] incluant : (1) la décomposition du domaine métier ; (2) l'analyse du système existant et (3) la modélisation objectifs-services. L'objectif majeur de toutes ces activités est d'élaborer un « modèle de services » [Arsanjani, 04]. Ce modèle comprend un ensemble de services candidats qui peuvent supporter les services métier, les processus ainsi que les objectifs métiers de l'entreprise.

Par ailleurs, chacune de ces activités est basée sur une des approches suivante : approche descendante (*top-down*), approche ascendante (*bottom-up*) et l'approche intermédiaire (*middle-out*). Dans la suite de ce paragraphe, nous décrivons ces différentes étapes (activités) ainsi que les approches appliquées dans ces étapes.

Etape 1 : Décomposition du domaine

Dans cette étape, on procède selon une approche descendante (*top-down*) où le domaine métier est décomposé sous forme de plusieurs « domaines fonctionnels » suivant leurs valeurs métiers. Après la décomposition en domaines fonctionnels, l'étape suivante est d'extraire à partir de chaque domaine les processus métier, les sous-processus.

Chacun de ces processus (ou sou-processus), est associé à un ou plusieurs « cas d'utilisation métier » (figure 3.14). Les cas d'utilisation métier représentent un élément clé dans le début d'application de toute démarche SOA. En effet, l'expérience montre que ces cas d'utilisation obtenus sont souvent considérés comme de « bons candidats » pour les services qui peuvent

être finalement exposés sous forme de services Web [Endrei et al, 04]. La figure 3.14 montre à titre illustratif, un exemple de décomposition du domaine métier.

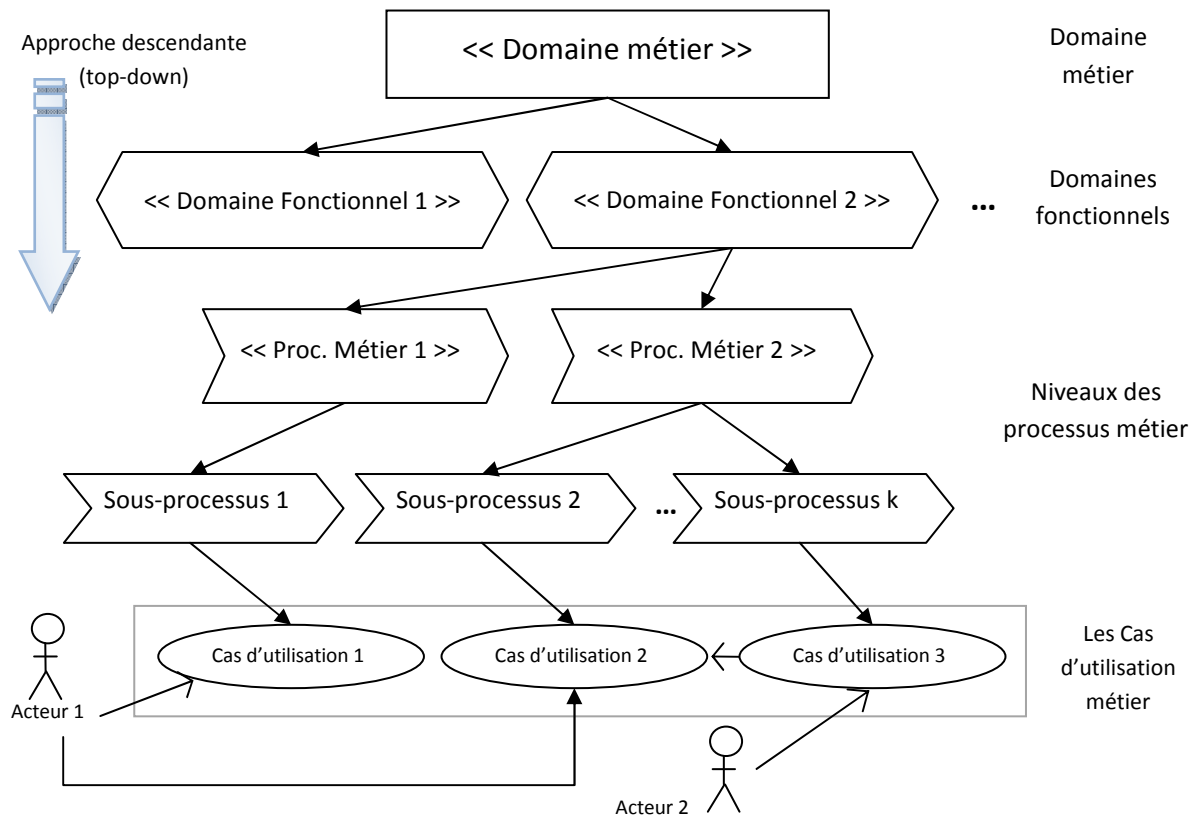


Figure 3.14 : Décomposition du Domaine (d'après [Ganci et al., 06]).

Étape 2 : Analyse du système existant

La plupart des projets SOA débutent sur une base contenant déjà un ensemble d'applications, d'où la nécessité d'analyser ces applications existantes, souvent connues sous le nom « systèmes légataires », afin de déduire et sélectionner les modules et les composants candidats les mieux adaptés pour une éventuelle intégration dans l'architecture SOA.

Cette analyse est effectuée sur la base des fonctionnalités du système légataire qui peuvent être supportées par le processus métier. Dans certains cas, il est nécessaire de composer et moduler ces systèmes afin de satisfaire les fonctionnalités des services métier. De ce fait, l'approche ascendante (*bottom-up*) est la plus recommandée dans cette étape.

Étape 3 : Création du modèle Objectifs-Services

Cette étape permet d'identifier et de valider tous les services qui n'ont pas été capturés durant les étapes précédentes (en appliquant les approches top-down et bottom-up). Ainsi, on utilise au cours de cette étape, une approche appelée : « approche intermédiaire » (*middle-out*) qui complète les approches top-down et bottom-up.

Pour créer ce modèle, on identifie les objectifs et sous-objectifs, qui seront structurés sous forme d'un arbre (figure 3.15), dont le sommet représente les objectifs majeurs du métier, afin de s'aligner avec la vision globale de l'entreprise. Le développement et l'exploration de ces

objectifs permet de découvrir de nouveaux services nécessaires pour la réalisation de ces objectifs. Dans cette étape, on a souvent recourt à l'analyse des besoins métier à travers les interviews effectués auprès des acteurs de l'entreprise.

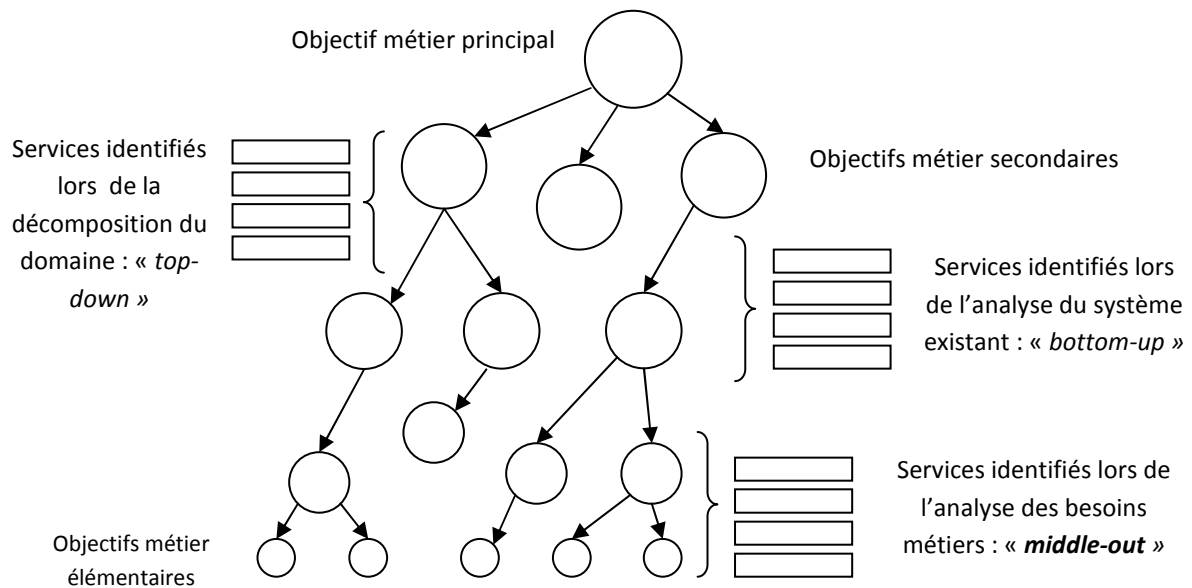


Figure 3.15 : Modèle de services-objectifs (d'après [Endrei *et al.*, 04]).

III.2. Phase 2 : Spécification

Au cours de cette phase, on procède à l'identification et à la spécification des composants réalisant les différents services. Cette phase comprend trois étapes [Arsanjani, 04] : (1) analyse des sous-systèmes ; (2) spécification des services et (3) spécification des composants.

Etape 1 : Analyse des sous-systèmes

Après avoir identifié les différents sous-systèmes dans l'étape décomposition du domaine, l'analyse des sous-systèmes permet de spécifier les dépendances et les flux d'informations échangés entre ces sous-systèmes.

L'analyse des sous-systèmes consiste donc à la création du modèle de composants représentant les traitements et les modèles liés au contenu des sous-systèmes qui devraient réaliser les services exposés. Cette étape permet de raffiner d'avantage les cas d'utilisation métier sous forme de cas d'utilisation système (informatique).

Les sous-systèmes seront composés à partir de trois types de composants [Ganci *et al.*, 06] : (1) les composants services qui ont une granularité importante et qui fournissent les fonctionnalités requise par le sous-système ; (2) les composants fonctionnels qui sont décrits à partir des besoins fonctionnels et qui sont utilisés par les composants services afin d'accomplir certaines tâches (gestion des données dans les référentiels, recherche d'un élément dans une base de données, etc.) ; (3) les composants technique qui sont généralement utilisés pour des besoins « non fonctionnel » tels que, la sécurité, gestion des fichiers log, etc. Ces différents types de composants sont illustrés sur la figure 3.16.

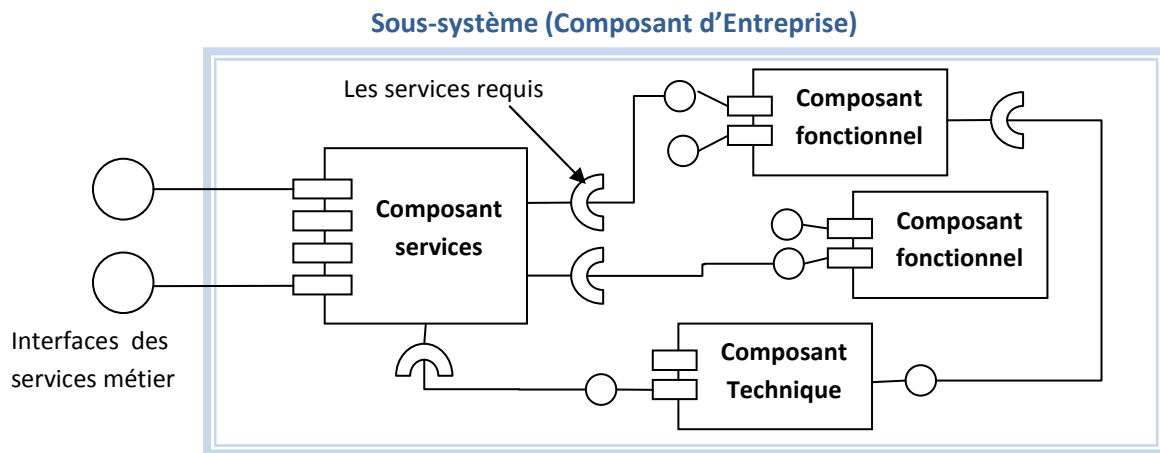


Figure 3.16 : Les composants d'un sous-système

Etape 2 : Spécification des services

Les étapes de décomposition du domaine, modélisation objectif-service et l'analyse du système existant ont permis d'identifier tous les différents services, selon leurs granularité et leur domaine fonctionnel. L'étape suivante consiste à la spécification de services. Celle-ci possède les activités suivantes :

- Classification des services en se référant la hiérarchie SOA définie par IBM (figure 3.17) : cela permet de déterminer les services qui devraient être composés ou qui seront impliqués dans le processus de chorégraphie et d'orchestration de services [Porter-Roth et al, 05].
- Catégorisation de services qui consiste à associer un ensemble de services à une catégorie : cela est particulièrement intéressant pour les services dotés d'une granularité fine, notamment les services CRUD [Fournier-Morel et al, 06]. Les catégories de services sont modélisées à l'aide des diagrammes de paquetages définis par la notation UML [Muller et Gaertner, 02].
- Spécification des messages échangés ainsi que les événements traités par les services : cette spécification peut être développée à l'aide du langage WSDL si on utilise les technologies des Web services.

Par ailleurs, au cours de cette étape, des décisions cruciales doivent être prises concernant les services à exposer et cela en référant aux principes SOA déjà évoqués dans le chapitre 2 (§II.2). Dans ce qui suit, on présente quelques aspects qu'on peut mettre en avant lors de cette étape [Ganci *et al.*, 06].

- Exposition/découverte : *le service peut-il être exposé à l'extérieur de l'organisation ? Le service possède-t-il une interface et une description bien définie pour être recherché par les clients à travers l'annuaire de services (UDDI) ?*

- **Traçabilité** : *Peut-on suivre la trace du service pour vérifier l'alignement du SI globale face aux objectifs métiers ?* En d'autres termes, il s'agit de considérer les services utiles pour la vérification de l'alignement métier.
- **Réutilisabilité** : *Le service peut-il servir pour d'autres processus métier ? Peut-on utiliser ce service pour créer de nouveaux services métier (composition de services) et processus métiers de haut niveau ?*
- **Etat du service (stateless)** : *Le service a-t-il besoin d'informations ou d'un état entre les requêtes ?* Il s'agit donc de vérifier la propriété de « sans-état » des services à exposer.

Etape 3 : Spécification des composants

Après avoir identifié tous les composants métier, fonctionnels et techniques, ainsi que les dépendances et les flux de données échangés entre les composants, on doit dans cette étape spécifier la structure interne dans chaque composant à travers les diagrammes de classes, ainsi que les relations de dépendance qui existent entre les différents éléments (packages, classes, etc.) du composant et ceux des autres composants appartenant au même sous-système. Il s'agit donc d'élaborer le modèle de classes d'objets relatif à chaque composant identifié.

III.3. Phase 3 : Réalisation

Au cours de cette phase, on considère les décisions sur l'architecture et la réalisation qui devront être prise lors de l'implémentation des différents composants. Ces décisions peuvent se porter sur des aspects fonctionnels tels que : *Quel module du système existant pouvant réaliser un service donné ? Quels sont les services qui seront exposés au-delà des frontières du SI global de l'entreprise ? Lesquels des services devront être implémentés au sein de l'entreprise et ceux qui seront développés par les partenaires externes ?*, etc.

Cependant, d'autres décisions peuvent être prises sur des aspects non-fonctionnels tels que : la sécurité, la qualité de service (QoS), la gestion et la surveillance (*monitoring*) des services.

Par ailleurs, cette phase comprend deux étapes de réalisation : (1) l'allocation des services dans les composants ; (2) l'assignation des composants par rapport aux différents niveaux SOA.

Etape 1 : Allocation des services

Cette activité permet d'assigner chaque service dans les sous-systèmes déjà identifiés. Elle consiste aussi à déterminer lequel des composants fourni une implémentation et une gestion pour chaque service spécifié. Donc il s'agit d'assurer que chaque service identifié possède « un socle » [Endrei et al, 04]. L'allocation des services est un élément clé dans la méthodologie SOAD, car elle permet d'assigner les services ainsi que leurs composants à travers les niveaux de l'architecture SOA choisie.

Etape 2 : Niveaux des composants

Après avoir alloué tous les services dans les composants appropriés, l'étape suivante est d'assigner chacun des composants par rapport aux différents niveaux de l'architecture SOA. La figure 3.17 montre ces différents niveaux selon l'approche SOMA d'IBM.

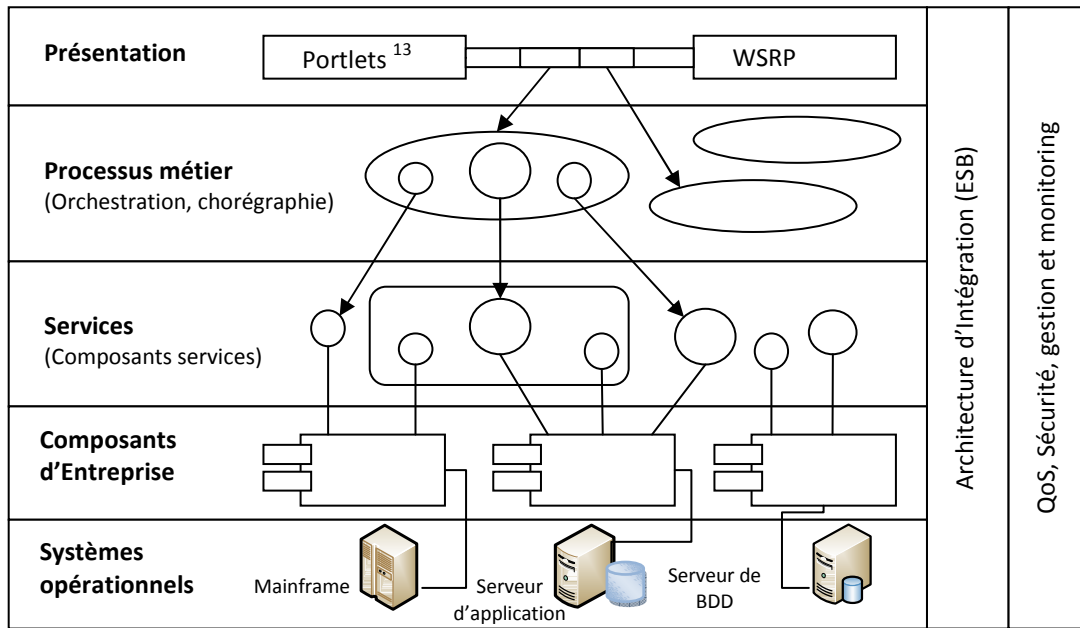


Figure 3.17 : Les niveaux SOA selon IBM [Arsanjani, 04]

III.4. Conclusion sur la méthode 2

Nous avons présenté dans cette section une autre approche appelée SOMA (*Services Oriented Modeling and Architecture*) proposée par IBM et qui a pour vocation principale de fédérer l'ensemble des méthodes déjà approuvées dans la modélisation des processus (BPM), au niveau architecture (EA) et enfin, dans les plus bas niveaux d'architecture, notamment les composants et les classes d'objets (OOAD).

SOMA décompose le processus d'analyse et de modélisation orientée services en trois phases essentielles : identification des services, spécification des services ainsi que leurs composants et enfin, la réalisation des différents composants et l'allocation des services à travers les niveaux SOA.

Par ailleurs, cette étude nous a permis d'acquérir de nouvelles techniques dans le cadre d'analyse et de modélisation orientée services notamment sur l'application des trois approches : (1) descendantes pour la découverte des nouveaux services, (2) ascendante pour extraire les services à partir du système existant et (3) l'approche intermédiaire à travers l'exploration de l'arbre correspondant aux objectifs métiers.

¹³ Portlet : module intégré à un portail d'entreprise, qui permet à l'utilisateur de disposer, dans la même fenêtre, d'un accès centralisé et convivial à différentes ressources (données, applications, sites Web, etc.), de modifier l'interface du portail selon ses besoins et de personnaliser ainsi son environnement de travail (source : <http://www.journaldunet.com/encyclopedie/definition/398/51/20/portlet.shtml>).

IV. Synthèse des approches présentées

Nous avons présenté dans les sections précédentes deux méthodes (ou approches) qui s'inscrivent dans le cadre d'analyse et de modélisation orientée services des SI. Chacune de ces méthodes a des particularités par rapport à l'autre, notamment au niveau de la couverture du cycle de développement d'un projet SOA (c'est-à-dire analyse, modélisation, implémentation, intégration et gestion) et aussi par les techniques et approches utilisées d'une part lors de l'analyse du système existant à travers les besoins métier, acteurs, applications existantes, etc. et dans la découverte, la spécification et le filtrage des services métier d'autre part.

Nous présentons dans ce qui suit, et selon plusieurs axes (analyse métier, identification des services, etc.), les aspects importants qui caractérisent chacune de ces méthodes afin d'aboutir à extraire les éléments pertinents qui nous serviront dans notre démarche d'analyse et de modélisation orientée services et qui est appliquée dans le cadre des systèmes de maintenance industrielle.

IV.1. Les approches et l'analyse du domaine métier

Dans cette section, nous comparons les éléments définis par chacune des approches en termes d'analyse du domaine métier (interactions, utilisateurs, processus, etc.).

Interaction homme-machine

Nous avons évoqué lors de l'étude de la première méthode, dédiée à la spécification d'IHM de supervision [Idoughi, 08], que les aspects liés aux interactions entre les acteurs humains et le système industriel (procédés, machines, etc.) sont mis en avant, notamment dans l'implication de ces acteurs (techniciens de maintenance, opérateurs, gestionnaires, etc.) dans le processus métier et aussi dans la spécification des services métier exposés aux utilisateurs.

En revanche, l'étude sur l'interaction Homme-Machine n'existe pas dans la seconde méthode (SOMA). En effet, cette dernière se focalise beaucoup plus sur les processus de type B2B (*Business to Business*) où leurs vocation est d'automatiser certaines tâches de haut niveau qui d'une part, traversent les frontières des SI des organisations, et qui délivrent la valeur ajoutée pour les deux parties (fournisseur et client) d'autre part. En outre, l'étude de l'interaction entre les acteurs et le système est limitée à un type de processus métier nommés B2C (*Business to Customer*) en considérant cette fois-ci que le consommateur du service (client), interagissant avec le processus, est à l'extérieur de l'organisation.

Ainsi dans la méthode SOMA, il n'existe pas une réelle analyse des interactions Homme-Machine. Néanmoins, cette étude peut être effectuée lors de la modélisation des « cas d'utilisation métier » après avoir identifié tous les domaines fonctionnels ainsi que les processus métier.

Dans notre cas d'étude, c'est-à-dire la maintenance industrielle, il est important de considérer les modèles d'interactions Homme-Machine comme un élément clé lors de l'analyse métier. En effet, l'étude sur l'interaction Homme-Machine s'avère impérative lorsqu'il s'agit de

modéliser par exemple un système de diagnostic de pannes. Ce dernier joue le rôle de « médiateur » entre l'acteur humain et l'équipement industriel.

Identification et Analyse des acteurs humains et tâches coopératives

Dans l'approche de spécification d'IHM [Idoughi, 08], l'identification des acteurs métier ainsi que leurs profils joue un rôle important dans la modélisation des services métier et dans la spécification d'IHM en liaison avec le domaine métier (supervision, maintenance, etc.). En effet, on retrouve les activités liées à l'identification et l'analyse des utilisateurs à travers plusieurs niveaux de l'approche, notamment dans l'analyse métier, analyse des utilisateurs, expression des besoins en termes de mobilité, etc.

Cependant, l'identification des acteurs métier ainsi que leurs besoins métier n'est pas considérée par l'approche SOMA d'IBM lors des phases d'analyse (décomposition du domaine, analyse de l'existant, etc.). Ajoutant à cela, la méthode SOMA ne prend pas en compte les aspects liés à la mobilité (opérateurs nomades, dispositifs d'accès, etc.) et aussi les tâches coopératives exécutées par les différents acteurs métier à travers les divers dispositifs de communication (PDA, PC Pocket, etc.), ainsi que les systèmes informatiques [Idoughi et Kolski, 06 b]. En effet, la méthode SOMA considère les tâches coopératives sous l'angle des activités transverses, c'est-à-dire qui implique le SI du fournisseur du service et celui du consommateur, on parle alors de « chorégraphie de services » (voir le chapitre 2 ; §II.4).

Décomposition du domaine métier

Un autre aspect important à souligner dans l'analyse métier est celui relatif à la décomposition du domaine métier. En effet, l'approche SOMA d'IBM considère que l'activité de décomposition du domaine comme un point de départ pour l'identification des différents constituants de l'architecture SOA, notamment les processus, sous-processus, cas d'utilisation métier et les services métier. Cette décomposition est basée sur les fonctionnalités potentielles issues du domaine métier (figure 3.14). Dans le domaine de la maintenance industrielle, on peut citer les domaines fonctionnels suivants : maintenance préventive, maintenance corrective, diagnostique, l'approvisionnement, etc.

Par ailleurs, dans l'approche de spécification d'IHM [Idoughi, 08], l'identification des processus métier est basée uniquement sur l'étude préalable de l'organisation existante. En outre, la décomposition du domaine est réalisée selon une vision Homme-Machine. Ainsi, les constituants de la décomposition représentent en fait les composants physiques de l'organisation industrielle (figure 3.6) tels que les sites de productions, procédés industriels, les machines, etc.

Le tableau comparatif 3.1 récapitule les caractéristiques de chaque méthode par rapport à l'analyse du domaine métier.

Méthode	Approche orientée services d'IHM [Idoughi, 08]	Approche SOMA d'IBM
Caractéristiques		
Identification et analyse des acteurs métier	supportée	absente
Modélisation de l'interaction Homme-Machine	supportée	absente
Analyse des tâches coopératives	Coopération entre les acteurs métier à travers les processus métier	Chorégraphie entre les services métier des organisations
Aspects de mobilité des acteurs métier	supportés	absents
Décomposition du domaine métier	Limitée au découpage du système Homme-Machine	Découpage sous forme de domaines fonctionnels
Identification des processus métier	supportée	supportée

Tableau 3.1 : Les approches et l'analyse du domaine métier

IV.2. Les approches et le cycle de développement

Les deux approches se distinguent par la façon de couvrir les étapes du cycle de développement (analyse, conception, implémentation, intégration et gestion). En effet, dans l'approche de spécification d'IHM, on trouve les activités liées à la phase d'analyse à travers plusieurs étapes de l'approche notamment lors de l'analyse du métier, des utilisateurs, les besoins en termes de services Web, etc. Ainsi, on remarque que la plupart des activités définies dans cette approche se rapportent au domaine d'analyse.

Cependant, les activités de conception sont considérées lors de la spécification des services métier, la catégorisation de services et aussi lors de la spécification d'IHM. De plus, l'approche présentée dans [Idoughi, 08] se concentre sur la spécification des services métier d'un point de vue IHM et aussi des utilisateurs métier, ce qui explique que les résultats obtenus par cette approche sont dirigés et exploités dans les phases de conception et d'implémentation comme illustré sur la figure 3.1. Tandis que dans la seconde approche (SOMA), les activités incluses dans les trois phases (identification, spécification et réalisation) couvrent tous les domaines d'analyse et de conception.

En effet, on trouve les activités d'analyse lors de la première phase d'identification de services, et les activités liées à la conception apparaissent à travers les deux autres phases (spécification et réalisation des services).

Par conséquent, les résultats obtenus par cette démarche sont directement exploités par les développeurs lors de l'implémentation de l'architecture SOA.

La figure 3.18 illustre d'une façon plus-ou-moins détaillée, une comparaison entre les deux approches par rapport aux aspects d'analyse et de conception.

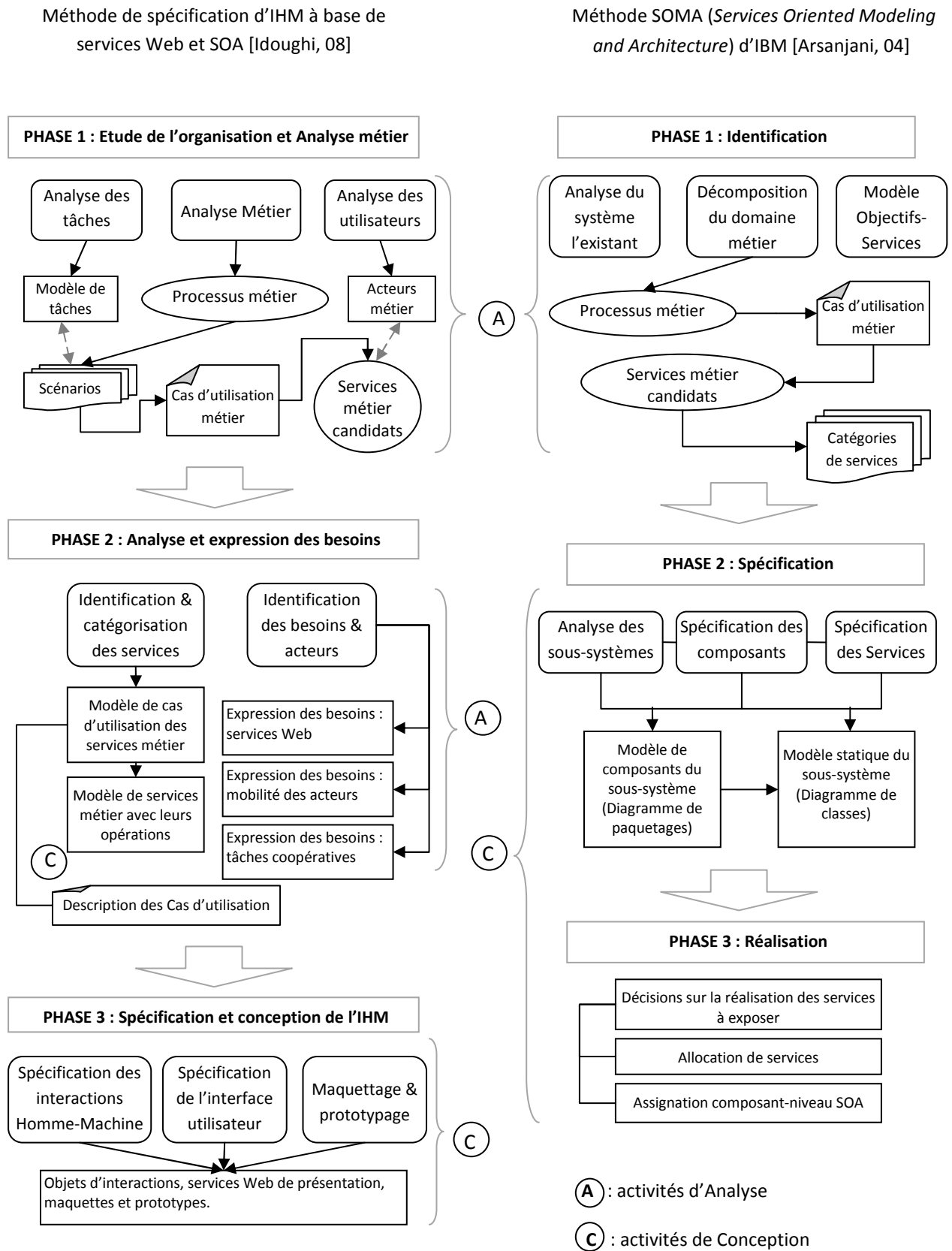


Figure 3.18 : Synthèse des approches vis-à-vis de l'analyse et de la conception

IV.3. Identification des services

L'étude des deux approches nous a conduit au constat suivant : « *le processus d'identification des services métier suit globalement les mêmes étapes dans les deux méthodes* ». En effet, cette identification commence par l'identification et l'analyse des processus métier, les différents scénarios liés aux déroulements de chaque processus et enfin, l'élaboration du modèle de cas d'utilisation métier ainsi que les acteurs métier qui permettra de dégager un premier ensemble de « services métier candidats ».

Néanmoins, la démarche d'identification de services diffère selon la vision adoptée par chacune des méthodes. En effet, dans l'approche de spécification d'IHM [Idoughi, 08], l'identification de services est établie selon les besoins métier des utilisateurs en termes de leur mobilité, l'interaction avec le système industriel à travers le processus métier, etc. il s'agit donc d'appliquer une vision « d'interaction Homme-Machine » pour élaborer « le modèle de services métier ».

Cependant, la méthode SOMA [Arsanjani, 04] procède à l'identification des services, en exploitant plusieurs techniques : *top-down*, *bottom-up* et *middle-out*, et cela à travers toute la phase d'identification (décomposition du domaine, analyse de l'existant et modèle objectifs-services).

En outre, SOMA préconise de prendre des décisions sur l'exposition des services métier notamment en vérifiant les principes SOA (voir chapitre 2 §II.2). Ces décisions permettent en fait de « filtrer » l'ensemble des services identifiés lors de la première phase (identification). On retrouve aussi ce filtrage de services dans la dernière phase (réalisation), en tenant compte cette fois-ci des aspects liés à la technologie utilisée, les modules réalisant les services, etc.

La figure 3.19 montre le principe de filtrage de services dans la méthode SOMA.

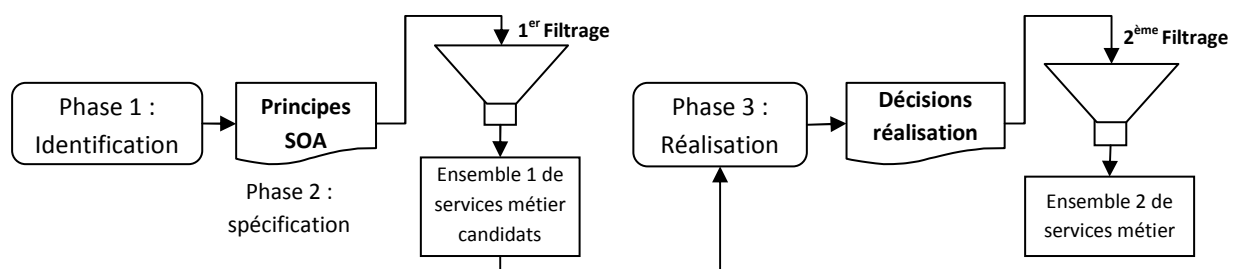


Figure 3.19 : Le filtrage des services dans SOMA (d'après [Dennerly, 07])

IV.4. Analyse de l'existant

Le dernier point de cette synthèse est celui relatif à l'analyse du système existant. Nous avons déjà évoqué dans l'introduction ainsi que dans le chapitre 2 (§III), que « la valorisation de l'existant » à travers la réutilisation des modules applicatifs en termes de services Web représente une pierre angulaire dans toute démarche SOA.

Dans le domaine de la maintenance industrielle, l'analyse de l'existant est primordiale, dans la mesure où plusieurs systèmes existants doivent coopérer voir même collaborer à travers les

fonctionnalités pouvant être offertes par chaque système (GMAO, SCADA, ERP, etc.) impliqué dans la maintenance.

Par conséquent, chacun de ces systèmes doit être soigneusement analysé afin d'extraire les fonctions potentielles à être exposées sous forme de services Web et aussi afin d'intégrer l'ensemble de ces systèmes au sein d'une même plate-forme de maintenance industrielle comme déjà développé dans le cadre du projet de *Proteus* [Bangemann *et al.*, 05] (chapitre 1, §III.3.b).

L'analyse de l'existant est traitée par les deux approches d'une manière différente. En effet, dans l'approche de spécification d'IHM [Idoughi, 08], l'analyse du système existant est réalisée à partir de l'étude des différents acteurs métier ainsi que les besoins métier (profils des utilisateurs, tâches coopératives, etc.). Ces besoins métier sont exprimés, comme déjà vu (§II.2), en termes de services Web, mobilité des acteurs et aussi des tâches coopératives accomplies par ces acteurs métier.

Par ailleurs, la méthode SOMA consacre toute une étape « *Analyse du système existant* » dans la première phase et qui utilise une approche ascendante (*bottom-up*) pour la décomposition des applications existantes (GMAO, ERP, etc.) en modules applicatifs. Cette décomposition est suivie par l'identification et la spécification des services par la définition des interfaces à exposer. Les modules obtenus représentent les composants implémentant les services Web.

La figure 3.20 illustre le principe général d'analyse de l'existant adopté par les deux approches.

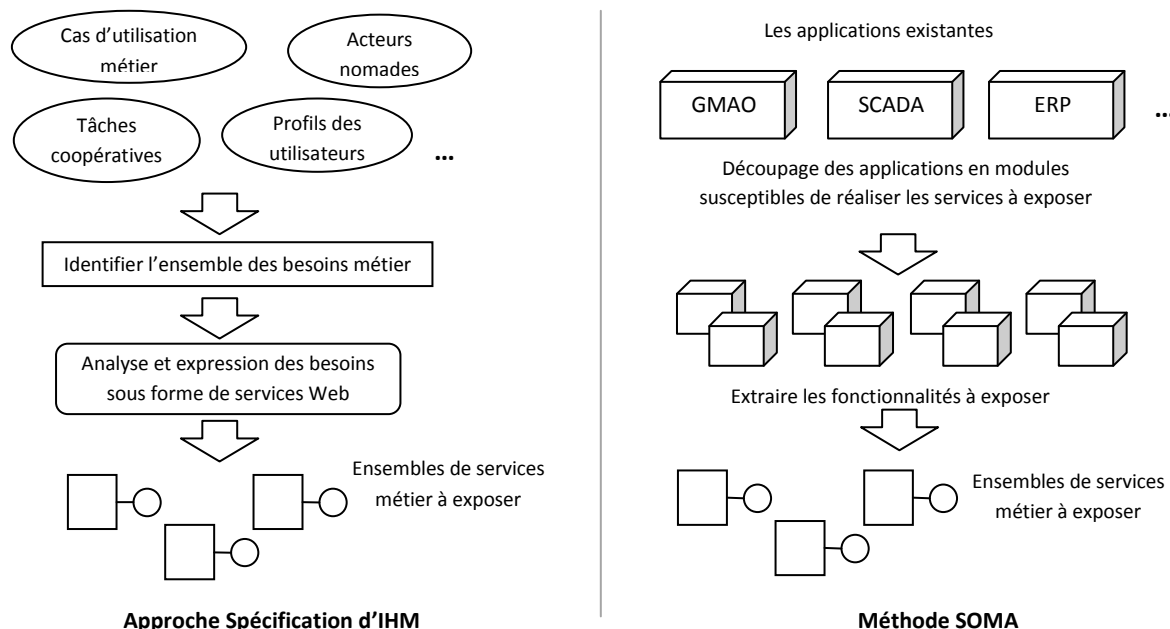


Figure 3.20 : Les deux approches et l'analyse de l'existant

V. Vers une approche d'analyse et de modélisation orientée services d'un système de maintenance industrielle

Après avoir étudié deux approches représentatives d'analyse et de modélisation orientées services, ainsi qu'une synthèse comparative des approches, nous présentons, dans cette section, une démarche en s'appuyant sur les méthodes déjà décrites ([Idoughi, 08] et SOMA [Arsanjani, 04]) à travers les étapes et les techniques qu'on juge importantes à considérer lors des phases d'analyse et de modélisation orientée services d'un système de maintenance industrielle.

La démarche proposée s'articule au tour de trois grandes phases : (1) étude de l'organisation de maintenance et analyse métier ; (2) analyse et expression des besoins et (3) spécification de l'architecture SOA du système de maintenance. La figure 3.21 montre une vue globale de la démarche appliquée dans notre projet.

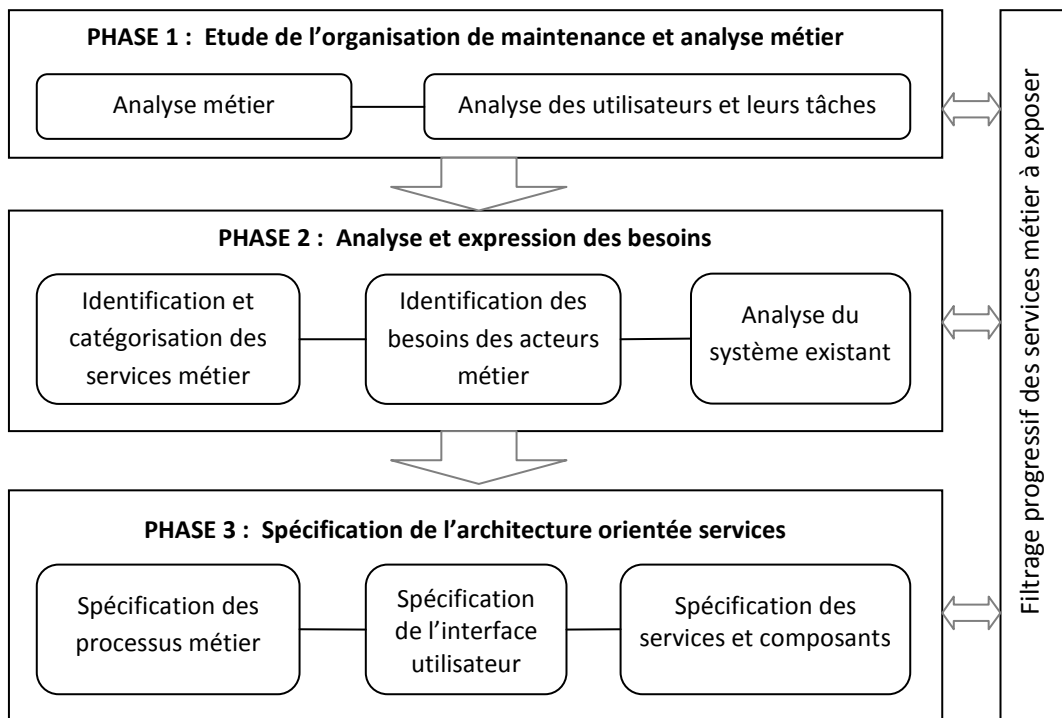


Figure 3.21 : Vue globale de la démarche proposée

Notre démarche s'appuie essentiellement sur les phases déjà définies dans l'approche de spécification d'IHM [Idoughi, 08]. Cependant, nous avons choisi d'intégrer certaines activités et techniques d'analyse et de modélisation définies à travers la méthode SOMA d'IBM, notamment dans l'identification de services en utilisant les trois approches déjà vues (*top-down*, *bottom-up* et *middle-out*).

Le choix adopté dans cette démarche est principalement justifié par le caractère et la spécificité de la maintenance industrielle en termes d'interactions homme-machine et aussi de coopération entre les différents acteurs métier ainsi que la collaboration qui peut se former entre le centre de maintenance de l'entreprise et ceux des partenaires externes (voir chapitre 1 §II.2).

En effet, les systèmes de maintenance sont caractérisés par la diversité d'interactions qui existe entre les différents acteurs métier (technicien, expert, magasinier, etc.) et le système industrielle (capteurs, équipements, etc.) en exploitant les dispositifs de communication de différente nature (PDA, PC Pocket, station de travail, etc.). Par conséquent, la problématique de la spécification d'IHM pour la maintenance s'impose comme déjà mentionné dans la chapitre 1 (§IV).

Par ailleurs, les phases d'identification des « bons candidats » pour les services métier à exposer d'une part aux acteurs métier et aux applications de l'organisation ainsi qu'aux partenaires externes d'autre part, doivent être effectuées d'une façon rationnelle afin d'éviter toute « prolifération de services » [Manouvrier et Ménard, 07; Ganci *et al.*, 06] mais aussi de contourner le « syndrome du fonctionnement en silos » déjà évoqué dans le chapitre 1 (§IV). De ce constat, nous avons inclus certaines techniques et approches de la méthode SOMA sur l'identification des services candidats à l'exposition et aussi sur « le filtrage progressif » [Dennerly, 07] de l'ensemble de services métier identifiés et qui sont susceptibles d'être exposés sous forme de services Web (figure 3.19).

Dans cette section, nous décrivons les trois phases de la démarche ainsi que les étapes impliquées dans chaque phase tout en se référant aux approches déjà présentées ([Idoughi, 08] et SOMA [Arsanjani, 04]).

V.1. Phase 1 : Etude de l'organisation de maintenance et analyse métier

Dans cette phase, nous nous sommes inspirés de l'approche proposée dans [Idoughi, 08], qui consiste à une étude préliminaire de l'organisation menée à travers deux types d'analyse : (1) analyse métier et (2) analyse des utilisateurs et leurs tâches comme illustré sur la figure 3.22.

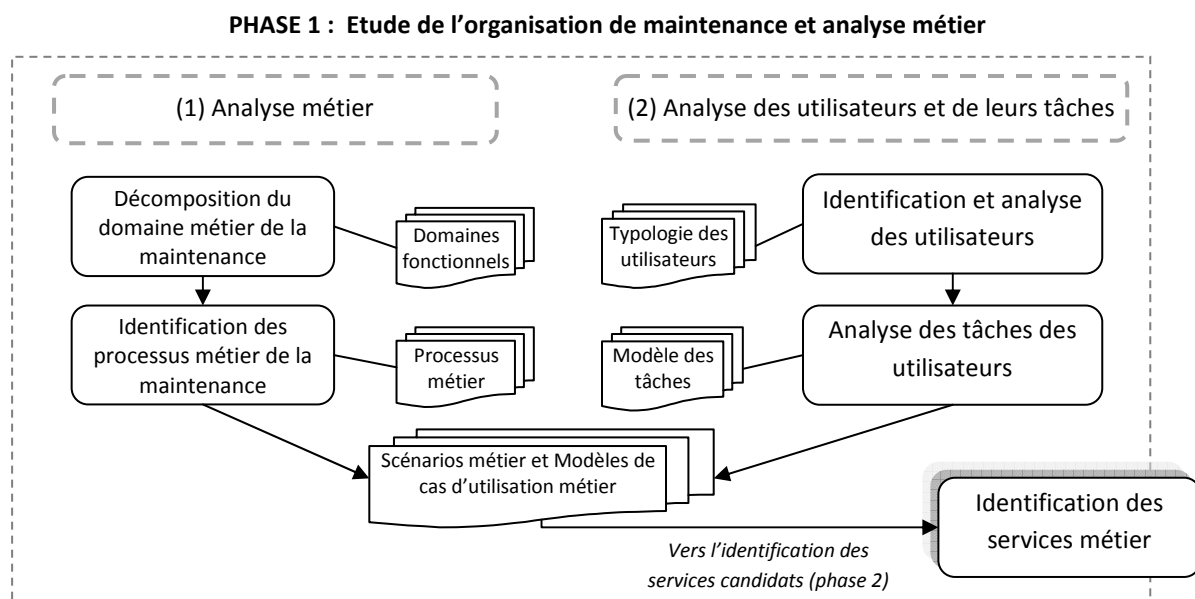


Figure 3.22 : Les étapes de la première phase

Etape 1 : Analyse métier

L'objectif principal de cette étape est d'identifier les processus métier à travers les différents scénarios de la maintenance. Ces scénarios seront utilisés afin d'extraire les cas d'utilisation métier relevant du domaine de la maintenance industrielle.

Pour cela, nous avons choisi au départ, de décomposer le domaine métier en un ensemble de domaines fonctionnels (figure 3.22) comme déjà vu dans la méthode SOMA (§III.1). Ainsi, nous identifions un ensemble de processus métier correspondant à chaque domaine fonctionnel (*functional area*).

La prochaine étape est d'utiliser la démarche de spécification d'IHM [Idoughi, 08] dans l'analyse et la modélisation des processus métier en se basant sur des scénarios cibles issus de la maintenance industrielle. A l'issue de cette étape, nous obtenons un « modèle de cas d'utilisation métier » qui exprime les fonctionnalités potentielles du système de maintenance. Par la suite, nous déduisons un premier ensemble de services métier décrits sous forme de scénarios métier à l'aide des diagrammes UML (diagrammes de séquences, cas d'utilisation, etc.). Notons enfin que nous nous inspirons des modèles définis par les deux méthodes ([Idoughi, 08] et SOMA) pour la description des scénarios et les cas d'utilisation métier.

Etape 2 : Analyse des utilisateurs et de leurs tâches

Lors de cette étape, il s'agit de définir et d'identifier les profils des utilisateurs à travers un modèle hiérarchique définissant la typologie des utilisateurs (figure 3.21). En outre, une analyse des tâches de ces utilisateurs est menée au cours de cette étape afin de recueillir les éléments importants pour la spécification des « services de présentation ». Ces derniers sont associés à la couche présentation de l'architecture SOA comme déjà illustrer sur la figure 3.17. Notons aussi qu'on peut utiliser plusieurs techniques et outils pour modéliser les tâches dont SADT (*Structured Analysis and Design Technique*, une description sur SADT est donnée en annexe E) pour la modélisation statique des activités et la modélisation par les arbres de tâches des utilisateurs CTT (*Concur Task Trees*, une description sur CTT est donnée en annexe E) pour les aspects dynamiques.

A l'issue de cette étape, nous obtenons un modèle reflétant la vue globale du système de maintenance (sous forme de diagramme de cas d'utilisation) avec tous les acteurs métier impliqués dans le processus métier ainsi que leurs tâches respectives (figure 3.23).

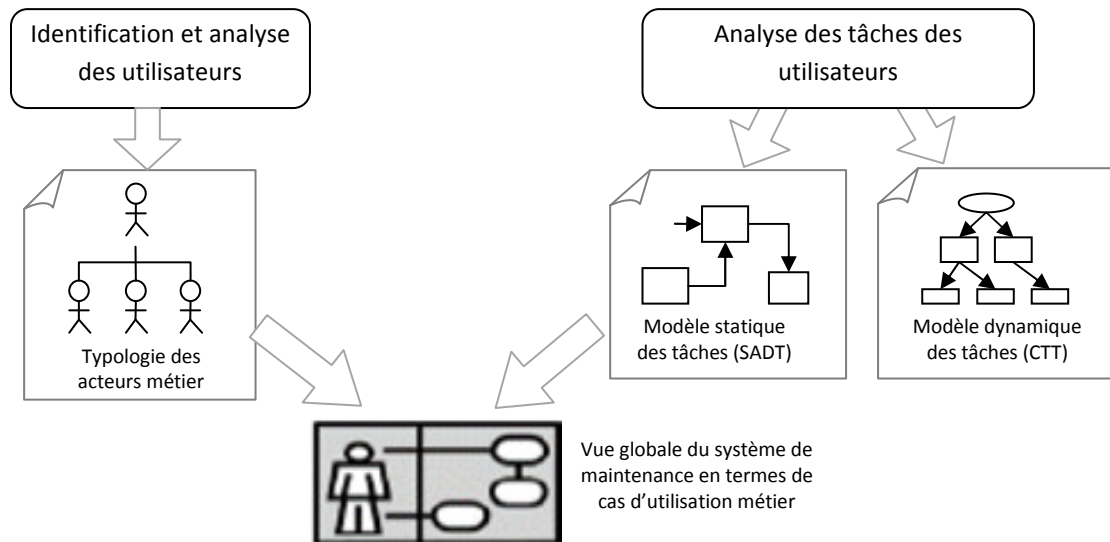


Figure 3.23 : Analyse des utilisateurs et leurs tâches

V.2. Phase 2 : Analyse et expression des besoins

Il s'agit dans cette phase d'effectuer un deuxième niveau d'analyse de l'ensemble des cas d'utilisation de maintenance et aussi des processus métier obtenus lors de la première phase d'identification, en tenant en compte cette fois-ci de l'identification et la catégorisation de ces services, des besoins métier en termes de services Web, de mobilité des acteurs et aussi des tâches coopératives (figure 3.24).

Néanmoins, une analyse du système de maintenance existant s'avère importante pour extraire les fonctionnalités susceptibles d'être exposées sous forme de services Web. Par conséquent, on reprend l'étape définie dans la méthode SOMA d'IBM, notamment celle relative à l'analyse de l'existant (§III.1) afin d'identifier les modules applicatifs (ou composants logiciels) capables de fournir une réalisation pour les services à exposer.

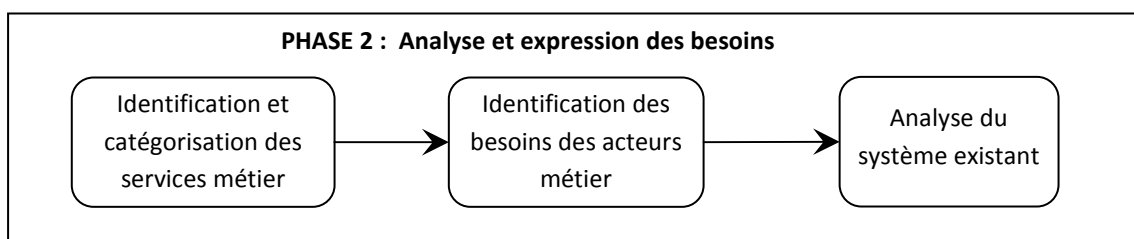


Figure 3.24 : Analyse et expression des besoins

Etape 1 : Identification et catégorisation des services métier

Cette étape consiste à identifier les services métier candidats, regrouper et les classer sous forme de packages (figure 3.25), où chaque package reflète une « catégorie de services métier » (ou objet métier). Le concept de catégorisation de services est utilisé dans plusieurs approches SOA, notamment dans *Praxème* [Bonnet, 05a ; Vauquier et Bonnet, 06] où le SI est cartographier sous forme d'unités regroupant un ensemble de services métier autour d'un « sujet métier » bien spécifique tel que : *intervention, équipement, fournisseur, etc.*

Par ailleurs, on retrouve aussi ce principe dans la démarche proposée dans [Idoughi, 08] et qui vise à identifier les nouveaux services à travers la composition ou l'orchestration des services déjà identifiés en obtenant ainsi des services de « forte granularité ». L'objectif principale de cette étape est de contourner les contraintes déjà citées dans la problématique, notamment celle relative au fonctionnement en silos applicatifs (chapitre 1 §IV).

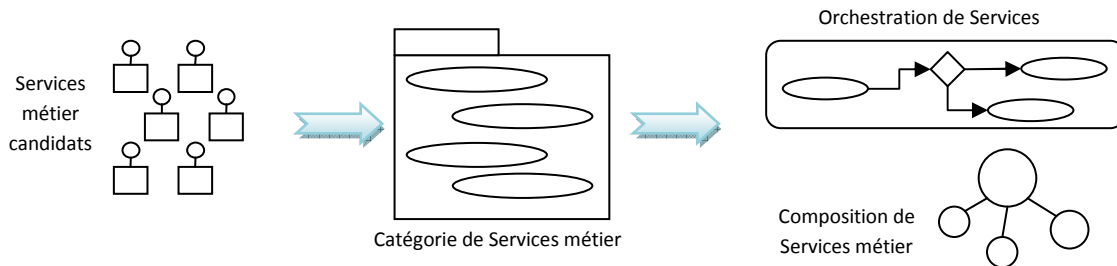


Figure 3.25 : Identification et catégorisation des services métier

Étape 2 : Identification des besoins métier

L'objectif de cette étape est d'extraire ou de déduire les besoins métier des utilisateurs en utilisant les scénarios métier bien ciblés. Ces besoins sont exprimés en termes : (1) des services Web à usage métier reflétant les interactions entre les utilisateurs et le système de maintenance industrielle ; (2) de mobilité des acteurs humains (localisation des opérateurs humains, dispositifs d'accès, etc.) ; et enfin (3) des tâches coopératives des acteurs métiers qui permettent de spécifier l'orchestration entre les différents services du système de maintenance afin d'accomplir la tâche coopérative visée par les acteurs humains.

A l'issue de cette étape, nous obtenons un ensemble de services Web exprimant les besoins métier en termes des concepts précédemment cités. La figure 3.26 montre les éléments essentiels de cette étape.

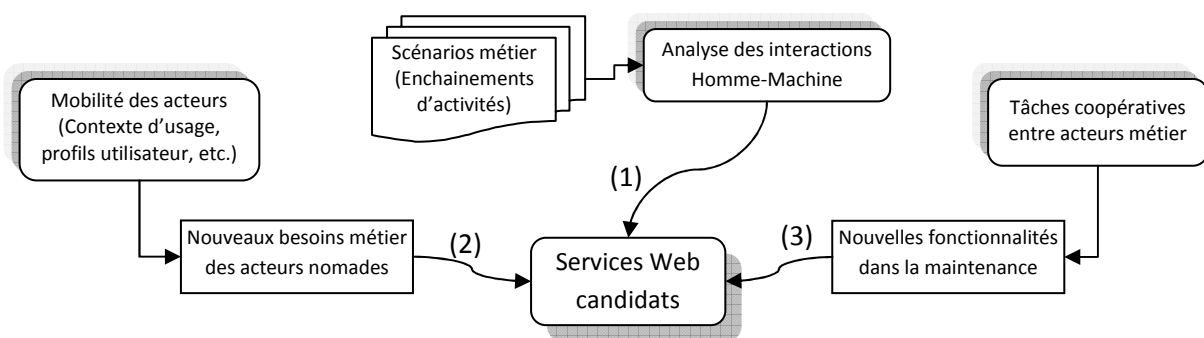


Figure 3.26 : Identification des besoins métier

Étape 3 : Analyse du système existant

On utilise lors de cette étape la décomposition des applications existantes au sein du système de maintenance (GMAO, SCADA, ERP, etc.) sous forme de modules applicatifs capables de fournir une implémentation (réalisation) pour les services métier préalablement identifiés. Il s'agit donc d'appliquer une approche ascendante (*bottom-up*), c'est-à-dire en partant du

système existant jusqu'aux services métier et les processus métier, comme déjà présenté dans la synthèse sur les deux approches (§IV.4 ; figure 3.18).

V.3. Phase 3 : Conception de l'architecture orientée services du système de maintenance

Dans cette phase, il s'agit de spécifier et de concevoir l'ensemble des éléments architecturaux (services, processus, etc.) préalablement identifiés et définis lors des phases précédentes. En d'autres termes, les activités de cette phase visent à concevoir tous les constituants de l'architecture orientée services (SOA) dédiée aux systèmes de maintenance industrielle.

Pour cela, nous avons choisi de mener cette phase de conception à travers trois étapes : (1) spécification et modélisation des processus métier ; (2) conception de l'interface utilisateur ; (3) spécification et conception des services et composants. Chacune de ces étapes est décomposée en sous-étapes permettant d'affiner progressivement la conception de l'architecture orientée services du système de maintenance industrielle (figure 3.27).

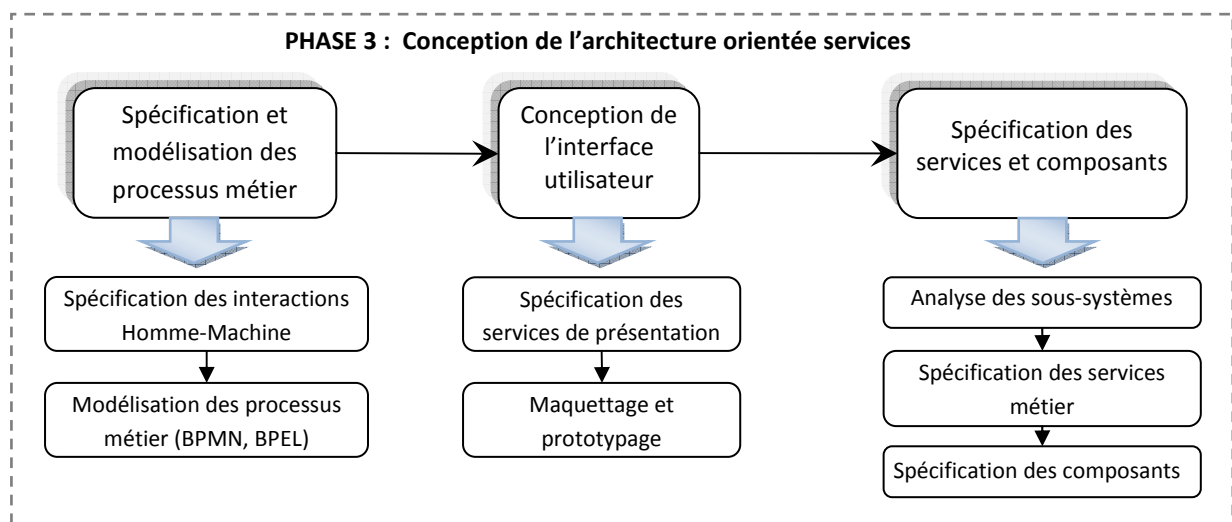


Figure 3.27 : Conception de l'architecture orientée services

Dans ce qui suit, nous décrivons chacune de ces étapes.

Etape 1 : Spécification et modélisation des processus métier

Cette étape consiste à définir la spécification des interactions Homme-Machine pour chaque processus métier identifié lors de la première phase d'analyse. Par ailleurs, nous procédons à la modélisation de l'enchaînement d'activités impliquées dans chaque processus, notamment ce qui correspond à l'orchestration de services, en utilisant la notation standard BPMN (*Business Process Modeling Notation*) présentée dans le chapitre 2 (§II.4).

Nous décrivons dans la suite de cette section, les sous étapes menées lors de cette étape.

a) Spécification des interactions homme-machine dans les processus

La spécification des interactions homme-machine à travers un processus métier, revient à formaliser les flux des différentes tâches exécutées par les acteurs humains, on parle alors de

tâches manuelles, ou celles exécutées par le système informatique à travers les services Web, on parle alors de tâches automatiques.

Par ailleurs, cette spécification doit considérer d'autres aspects, notamment ceux qui concernent la communication entre les services, la spécification des sous processus et les règles métier à appliquer (synchronisation des tâches, parallélisme, condition d'exécution, etc.) au sein du processus.

Pour cela, nous empruntons la spécification définie dans l'approche [Idoughi, 08] relative à la spécification des systèmes interactifs, et qui est implémentée à l'aide du langage standard BPEL (*Business Process Execution Language*) présenté dans le chapitre 2 (§II.4).

Cette spécification met en avant plusieurs éléments, dont les partenaires externes (acteurs métier, services Web, etc.) interagissant avec le processus et aussi les variables d'échange qui sont internes au processus et qui permettent de maintenir l'état des échanges au sein du processus. Ainsi, l'utilisation de ces variables permet de compenser la caractéristique des services métier relative à l'absence d'état (*stateless*) déjà évoquée dans le chapitre 2 (§II.2).

La figure 3.28 illustre les éléments essentiels correspondant aux interactions homme-machine au sein du processus métier de la maintenance.

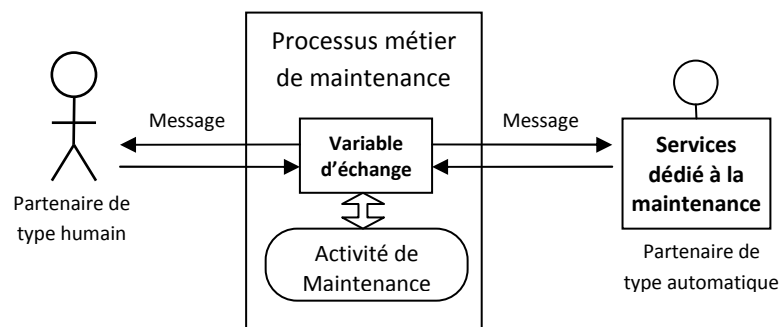


Figure 3.28 : Les éléments impliqués dans un processus métier. *Inspiré de* [Idoughi, 08]

La spécification consiste à définir le processus métier en termes de services Web ainsi que l'ordre dans lequel sont exécutés ces services (exécution des opérations et interactions avec les partenaires externes).

Voici un exemple (figure 3.29) de cette spécification d'interactions dans un processus métier de la maintenance.

```

<Définition>
<Variables>
<variable nom='nom_variable' type_message= 'type du message que la variable doit véhiculer'>
...
</Variables>
<messages>
<message nom='nom_message'>
  <partie nom='nom_parite' type='type de données'>
...
</message>
</messages>
<Services>
  <Service ='nom_du_service'>
    <Opération nom='nom_operation'>
      <message en entrée nom='nom_message' />
      <message en entrée nom='nom_message' />
      <message erreur nom='nom_message' />
    </Opération>
...
  </Service>
</Services>
</Définition>

```

Figure 3.29 : Exemple de spécification d'interactions dans un processus métier [Idoughi, 08]

b) Modélisation des processus métier

Cette activité consiste à transformer les modèles élaborés lors des phases d'analyse (décomposition du domaine, analyse des utilisateurs, etc.) décrits sous forme de diagrammes UML (diagrammes de séquences, activités et de collaborations), ainsi que les spécifications définies à travers l'étape précédente (spécification d'IHM), en un ensemble de « modèles conceptuels » décrivant d'une façon succincte l'enchaînement d'activités, les interactions avec les partenaires, les règles métier, les exceptions, etc.

Ces modèles sont construits à l'aide de la notation BPMN qui fournit un ensemble d'outils permettant aux concepteurs système, mais aussi aux acteurs métier (*ou MOA : Maîtrise d'OuvrAge*) de modéliser leurs processus métier sans avoir des prérequis sur les technologies d'information (IT). Les modèles obtenus, seront traduits (mapper) vers des programmes exprimés à l'aide du langage d'exécution des processus BPEL. Le fichier BPEL résultant est utilisé par le moteur BPM (*Business Process Management Engine* ou l'orchestrateur de services) pour générer une instance d'exécution du processus métier comme illustré sur la figure 3.30.

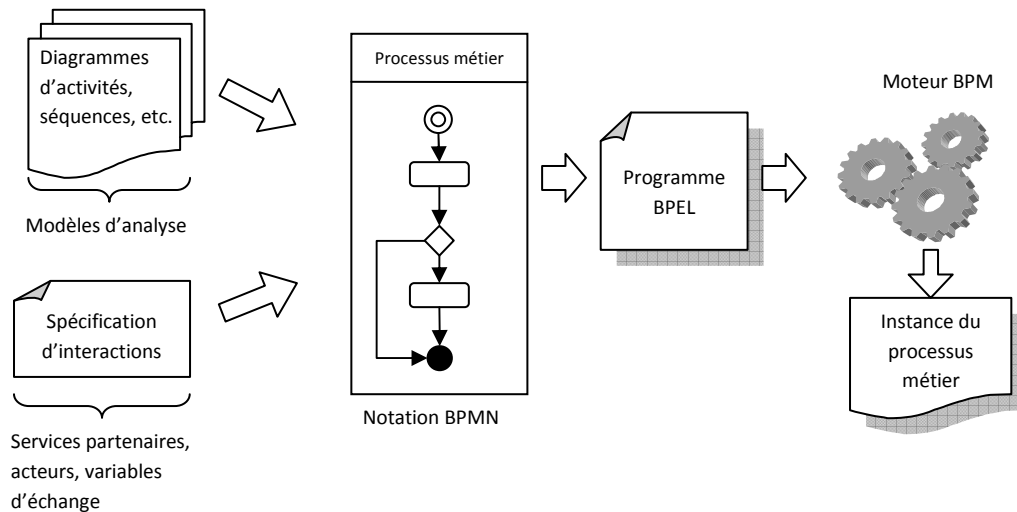


Figure 3.30 : Conception d'un processus métier

Etape 2 : Conception de l'interface utilisateur

Au cours de cette étape, nous modélisons les différents éléments constituant l'interface utilisateur. En effet, il s'agit de concevoir les services dédiés à la présentation qui seront intégrés dans des « applications composites » (chapitre 2, §II.3.1) directement exploitables par les utilisateurs finals de la maintenance industrielle (experts, techniciens, etc.). Nous avons choisi de mener cette étape en deux sous étapes : (a) spécification des services présentation et (b) maquettage et prototypage.

a) Spécification des services de présentation

Dans cette étape, nous décrivons les objets d'interaction (fenêtre, vue, boîtes de dialogue, etc.) correspondant aux différents services Web métier de la maintenance industrielle. Pour cela, nous utilisons une spécification permettant de créer des interfaces utilisateurs pour les services Web adaptables aux besoins et aux profils des utilisateurs.

En effet, nous avons choisi d'utiliser la spécification WSRP (*Web Services for Remote Portlets*) proposée par OASIS (www.oasis.org). Une description de cette spécification est donnée en annexe (annexe F). Cette spécification nous permet de concevoir tous les fragments (ou *Portlets*) du portail Web de la maintenance où chaque fragment correspond à un service Web orienté présentation. L'objectif majeur visé par cette spécification (également d'autres spécifications de la couche présentation SOA), est de permettre de concevoir l'architecture de la couche présentation SOA indépendamment des technologies sous-jacentes (EJB, DCOM, Corba, etc.). La figure 3.31 montre la position des services Web orientés présentation dans l'architecture orientée services.

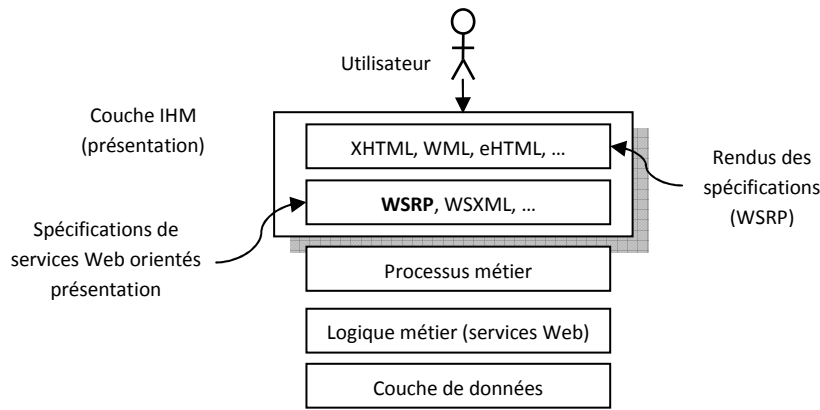


Figure 3.31 : Position services Web orientés présentation dans SOA. Inspiré de [Idoughi, 08]

b) Maquettage et prototypage

Après avoir modélisé les différents constituants de l'interface utilisateur, nous procédons dans cette étape à l'élaboration des maquettes ainsi que les prototypes de l'interface utilisateur du système de maintenance.

Pour cela, nous créons les éléments d'interface (fenêtres, boîtes de dialogue, etc.) conformément aux besoins métier des utilisateurs et aussi de leurs tâches liées à la maintenance industrielle. Précisément, l'élaboration de ces maquettes doit s'appuyer sur des scénarios issus des activités de la maintenance industrielle tels que, « la demande d'intervention », « sous-traitance de la maintenance », « approvisionnement en pièces de rechange », etc. Tous ces scénarios devront nous permettre de représenter les services offerts par le système de maintenance, sous forme de maquettes, en concordance avec les exigences des utilisateurs et les contraintes du métier.

L'étape suivante est d'évoluer ces différentes maquettes vers des « prototypes d'interface » offrant une vue assez exhaustive de l'interface du futur système de maintenance. Le processus de maquettage/prototypage qu'on a adopté suit celui montré sur la figure 3.11. Le principe appliqué dans cette étape est illustré sur la figure 3.32.

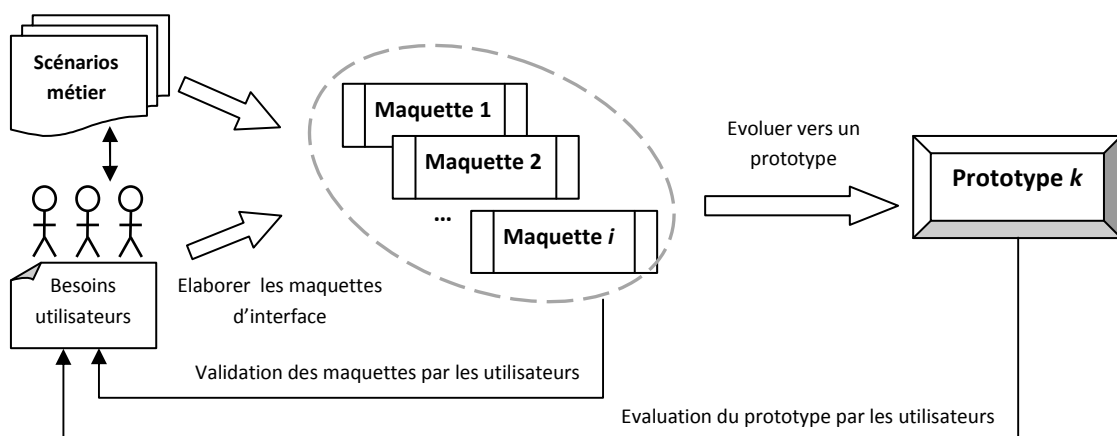


Figure 3.32 : Maquettage/prototypage de l'interface utilisateur

Etape 3 : Spécification des services et composants

Cette phase de conception représente le cœur de la démarche orientée services adoptée dans notre projet. En effet, il s'agit de s'appuyer sur les concepts et les techniques apportées par la méthode SOMA d'IBM afin de réaliser la conception de tous les services déjà identifiés (services métier, services CRUD, services orientés IHM, etc.), ainsi que les différents composants (composants services, techniques et fonctionnels) fournissant l'implémentation pour les services spécifiés.

A l'issue de cette étape, nous aurons l'ensemble des modèles conceptuels qui forment l'architecture orientée services (SOA) du système de maintenance. Ces modèles peuvent être directement exploités par les développeurs lors de la phase d'implémentation.

Nous avons choisi de mener cette étape à travers trois sous-étapes : (a) analyse des sous-systèmes ; (b) spécification des services et (c) spécification des composants. Dans ce qui suit, nous décrivons ces différentes étapes.

a) Analyse des sous-systèmes

Dans cette étape, il s'agit d'affiner tous les cas d'utilisation métier définis à travers la première phase (analyse métier) sous forme de « cas d'utilisation système » [Endrei *et al.*, 04]. Par conséquent, chaque sous-système (domaine fonctionnel) est modélisé en termes d'un ensemble de composants de types différents (service, fonctionnel et technique).

À l'issue de cette étape, nous obtenons une description sur la structure globale du système de maintenance sous forme de « modèle de composants ». Ce dernier met en avant tous les composants réalisant les services déjà identifiés, ainsi que les flux de dépendances entre ces composants et par conséquent, entre les sous-systèmes (voir la figure 3.16).

b) Spécification de services

Cette étape consiste à spécifier tous les éléments qui constituent un service informatique : opérations, types des messages échangés, les événements, etc. En outre, nous allons spécifier les caractéristiques non-fonctionnelles (qualité de service, post et près conditions, etc.) qui devraient faire partie du « contrat de service » entre le fournisseur et le consommateur du service. Pour cela, nous avons choisi de mener cette étape à travers les activités suivantes :

(1) spécification des objets métier : qui constituent les informations structurées de façon à être gérées dans un cadre de référentiels ou bases applicatives (SGBD, fichiers XML, etc.) [Raymond, 07]. Ces objets servent de base pour la construction des services CRUD;

(2) définition des types de données échangées (TDE) : qui forment toutes les informations véhiculées entre les participants (fournisseurs et consommateurs de service). Pour spécifier les TDE, on utilise généralement les schémas XML (au format XSD : *XML Schema Definition*);

(3) spécification des opérations : qui consiste à définir les paramètres d'entrée (*input*), type du résultat de l'opération (*output*) et éventuellement les messages d'erreur en cas d'exception (*fault*) ;

(4) spécification des interfaces : il s'agit de regrouper toutes les informations déjà citées sous un format standard (généralement WSDL) afin d'exposer les services offerts par le fournisseur pour ses clients ;

(5) spécification des aspects (contraintes) non-fonctionnels : cela consiste à définir toutes les informations concernant la qualité de services offerte tel que les performances (temps de réponse), la fiabilité (taux d'erreur), la disponibilité et la sécurité. D'autres contraintes peuvent s'articuler sur les conditions d'utilisation à travers les pré et post conditions (chapitre 2, §II.2).

La figure 3.33 illustre les éléments essentiels à considérer lors de la spécification de services.

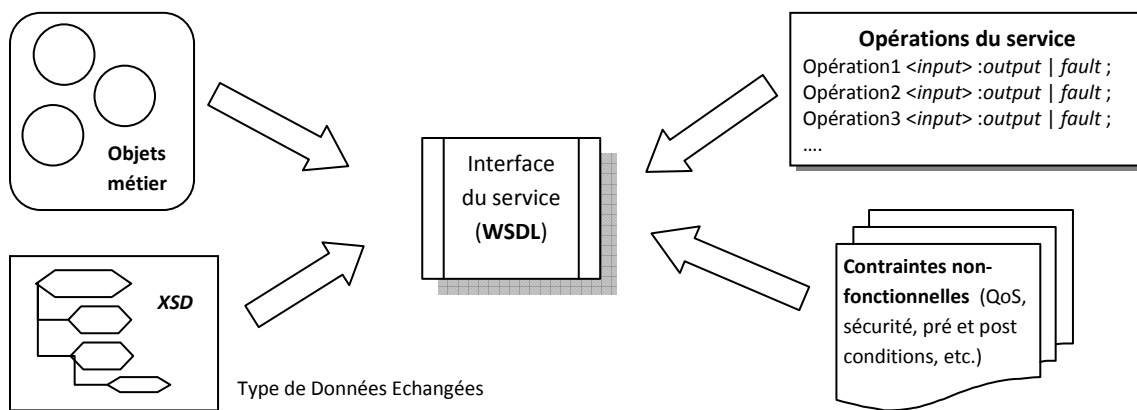


Figure 3.33 : Spécification de services

A l'issue de cette étape, nous parvenons à décrire d'une façon exhaustive les interfaces de tous les services déjà identifiés. Notons cependant, que des décisions doivent être prises au cours de cette étape, notamment sur les services à exposer aux clients (et donc à déployer leurs interfaces dans l'annuaire UDDI), la réutilisation, l'alignement du service par rapport aux objectifs métiers, etc. Il s'agit ainsi d'un second filtrage à appliquer pour les services métier comme déjà indiqué précédemment (§IV.3).

c) Spécification de composants

Après avoir identifié l'ensemble des composants (services, fonctionnels et techniques) qui constituent les briques de l'architecture orientée services (SOA) ainsi que spécifié les services à travers les interfaces WSDL, nous allons dans cette dernière étape, affiner d'avantage le modèle conceptuel de l'architecture SOA en spécifiant, cette fois-ci, les relations entre les classes du même composant, mais aussi entre les composants du même sous-système (composant d'entreprise) et éventuellement les flux d'informations échangées entre ces sous-systèmes.

Pour cela, nous utilisons les différents diagrammes offerts par la notation UML afin de décrire d'une part, les aspects dynamiques du système (interactions entre les composants, orchestration de services, etc.) à travers les diagrammes de collaborations, de séquences et

d'activités, et d'autre part pour les aspects statiques (modèle de composants, modèle d'objets, etc.) à travers les diagrammes de classes et de paquetage [Muller et Gaertner, 02].

A l'issue de cette étape, nous achevons la conception de la couche d'abstraction la plus basse du modèle SOA. Cependant, des retours vers les phases antérieures (phases 1 et 2) peuvent s'imposer afin de compléter la conception de tous les composants de l'architecture SOA.

En conclusion, nous soulignons que tous les modèles élaborés lors de cette phase (conception de services et composants) peuvent être directement exploités par les développeurs lors de l'implémentation. Par ailleurs, des décisions sur la réalisation des services peuvent s'avérer très utiles, notamment sur les technologies d'implémentation (services Web, CORBA, etc.), sur la réalisation des composants (EJB, DCOM, etc.) et aussi sur la mise en œuvre des services dédiés à la présentation (JSP, Servelets, Portlets, etc.). D'autres décisions peuvent se rapporter sur l'architecture physique du système (serveurs d'applications, pare-feux, etc.).

VI. Conclusion

Dans ce chapitre, nous avons présenté une démarche relative au domaine de l'orienté services (SOA), faisant l'objet d'un cadre méthodologique pour l'analyse et la modélisation orientée services d'un système de maintenance industrielle.

Ce cadre s'appuie sur deux approches : Spécification d'IHM à base de services Web proposée dans [Idoughi, 08] et la méthode SOMA (*Services Oriented Modeling and Architecture*) d'IBM [Arsanjani, 04].

La première approche vise à rapprocher les modèles et les techniques issues du GL et de l'IHM d'une part, des techniques du Web (services Web) et la gestion des processus métier (BPM) d'autre part. Par ailleurs, la seconde approche (SOMA) vise à exploiter plusieurs techniques d'identification et d'analyse des services métier à travers les approches descendante (*top-down*), ascendante (*bottom-up*) et intermédiaire (*middle-out*). D'autres techniques d'analyse sont proposées dans SOMA, notamment au niveau des sous-systèmes, spécification et allocation de services, etc.

Afin de mieux tirer les éléments à considérer dans notre démarche, nous avons présenté une synthèse comparative des deux approches selon plusieurs axes (identification de services, cycle de développement, etc.).

La démarche adoptée dans notre projet s'appuie essentiellement sur les phases proposées dans la première approche [Idoughi, 08] tout en complétant les phases de conception par celle définies dans SOMA. Ainsi, nous avons choisi de mener cette démarche à travers trois phases : (1) étude de l'organisation de maintenance et analyse métier ; (2) analyse et expression des besoins et (3) conception de l'architecture SOA du système de maintenance.

Dans la phase (1), une étude préliminaire de l'organisation est menée à travers deux étapes : (1) l'analyse métier qui vise à identifier tous les processus métier ainsi qu'un premier

ensemble de services métier candidats, (2) l'analyse des utilisateurs mette en évidence d'une part, les besoins métier en termes de services Web, mobilité et des tâches coopératives et une analyse des tâches de maintenance est menée afin de modéliser les aspects dynamique à travers les scénarios d'usage métier d'autre part.

Dans la phase (2), un deuxième niveau d'analyse des besoins exprimé lors de la 1^{ère} phase est effectuée à travers : (1) l'analyse du système existant, (2) identification et catégorisation des services métier et (3) identification des besoins des utilisateurs.

Enfin, la phase (3) regroupe l'ensemble d'activités liées à la conception de l'architecture SOA dédiée aux systèmes de maintenance industrielle. Cette phase est menée selon trois étapes : (1) spécification et conception des processus métier déjà identifiés, (2) conception de l'interface utilisateur (IHM) en exploitant les besoins métier des utilisateurs identifiés lors des phases 1 et 2 et (3) spécification et conception des services et composants qui vise à créer un modèle complet de services ainsi que les composants réalisant ces services, tout en spécifiant les flux d'informations entre les différents constituants de l'architecture SOA.

Dans le chapitre suivant, nous exposons la mise en œuvre de l'approche présentée en vue de démontrer ou valider les différentes phases de la démarche sur une étude de cas relative à la maintenance industrielle d'une unité de conditionnement appartenant à un groupe industriel d'agroalimentaire (*Cevital*).

CHAPITRE IV : APPLICATION ET MISE EN ŒUVRE DE L'APPROCHE PROPOSEE SUR UNE ETUDE DE CAS RELATIVE A LA MAINTENANCE INDUSTRIELLE (Unité de conditionnement - Cevital)

I. Introduction et présentation du cas d'étude

Nous avons présenté dans le chapitre précédent une démarche d'analyse et de modélisation orientée services visant une architecture SOA pour les systèmes de maintenance industrielle.

Dans ce chapitre, nous appliquons l'ensemble des concepts avancés sur une étude de cas représentative à la maintenance industrielle au niveau du groupe agro-alimentaire *Cevital* de Bejaia (Algérie). Précisément, nous focalisons sur les activités de maintenance relatives à une unité de conditionnement située au sein de la raffinerie des huiles du complexe *Cevital*. Par la suite, nous envisageons de généraliser cette étude sur l'ensemble des unités du complexe et sur l'ensemble des sites du groupe *Cevital*. L'objectif visé par cette étude, est de proposer une architecture SOA pour un système de maintenance capable de supporter les approches de maintenance présentées dans le chapitre 1 (§II.2), notamment les approches coopérative et collaborative.

Dans ce qui suit, nous présentons le contexte d'étude, en s'intéressant particulièrement aux éléments pertinents de la maintenance. Ces éléments servent à déterminer plusieurs cas d'utilisation métier relatifs à la gestion de la maintenance au niveau de l'unité de conditionnement des huiles. Pour cela, des scénarios représentatifs sont décrits afin de mieux cerner les fonctionnalités du futur système de maintenance basé sur les concepts SOA.

Dans ce chapitre, nous présentons une mise en œuvre de la démarche à travers les phases déjà énoncées (chapitre 3) sur un cas d'étude réel relatif à la maintenance industrielle.

II. Phase 1 : Etude de l'organisation de maintenance et analyse métier

Dans cette phase, nous procédons à une étude préalable de l'organisation du service de maintenance au sein de l'unité de conditionnement. Cette étude vise à délimiter le champ d'étude, identifier le problème et les objectifs métiers relatifs à la maintenance.

Nous menons cette phase à travers deux étapes d'analyse (chapitre 3, §V.1) : (1) analyse métier et (2) analyse des utilisateurs et leurs tâches. Ces étapes sont décrites ci-après.

II.1. Etape 1 : Analyse métier

Au cours de cette étape, nous considérons plusieurs éléments relatifs au métier du service de maintenance tels que, la structure du centre de maintenance et ses relations avec les autres structures de l'entreprise, les systèmes informatiques et les technologies mises en place dans lesquelles les différents acteurs humains de l'organisation évoluent.

Par conséquent, nous abordons cette étape par une description globale du champ d'étude avant de procéder aux étapes d'analyse métier (décomposition du domaine, identification des cas d'utilisation, etc.).

Description du contexte d'étude

a) *Présentation de l'Entreprise d'accueil* : Le complexe industriel agroalimentaire Cevital implanté au sein du port de Bejaia. Il composé de trois unités de production : unité d'huile, raffinerie du sucre et une unité de production de la margarine.

La figure 4.1 montre une vue globale de l'entreprise et la position de l'unité de conditionnement :

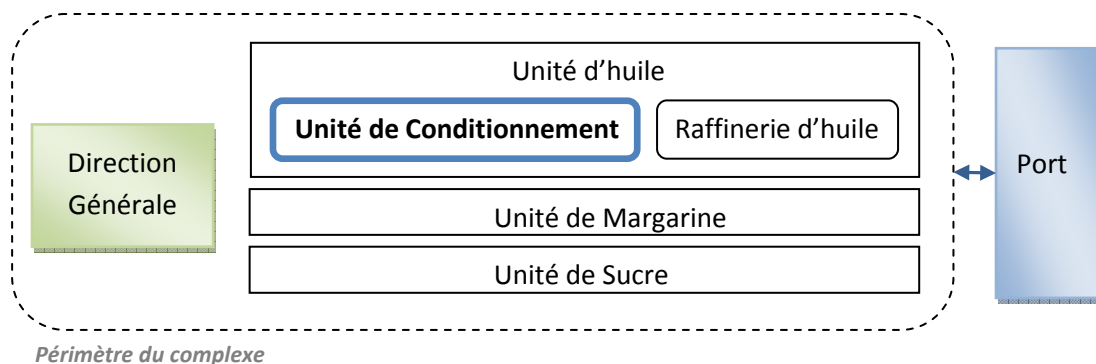


Figure 4.1 : Vue globale du complexe Cevital

b) *Description du processus métier de l'unité de conditionnement* : Le rôle principal de cette unité est le conditionnement des quantités d'huiles provenant des stations de pompage de la raffinerie d'huile. Ce conditionnement s'effectue dans des bouteilles de différents calibres, pour ensuite les emballer dans des fardeaux. Ces derniers sont acheminés directement vers le magasin de stocks.

L'unité contient essentiellement trois types de machines : (1) la souffleuse qui a pour rôle de produire des bouteilles à partir d'une matière première spéciale (PET : Polyéthylène Téréphtalate) ; (2) la remplisseuse qui permet de remplir avec l'huile provenant de la raffinerie les bouteilles produites par la souffleuse ; (3) l'étiqueteuse qui est une machine permettant de placer les étiquettes sur les bouteilles et aussi les dater avec les dates de la production et de péremption.

La figure 4.2 illustre la structure générale de l'unité.

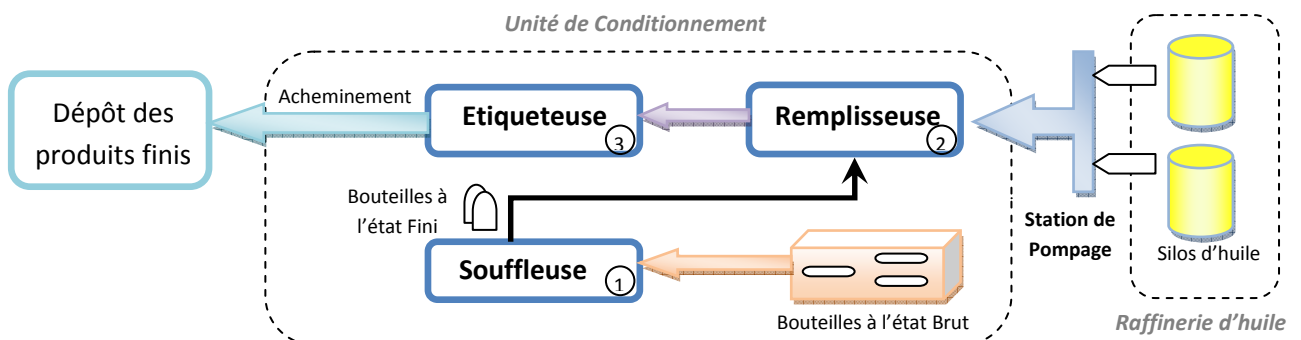


Figure 4.2 : Structure générale de l'unité de conditionnement

c) *Système de gestion maintenance existant* : La maintenance au sein de l'entreprise Cevital est gérée par un logiciel GMAO (Gestion Maintenance Assistée par Ordinateur) nommé « CosWin ». Ce logiciel est installé sur un serveur central relié par un réseau avec des terminaux installés dans les ateliers et dans le bureau de méthodes (figure 4.3). Les utilisateurs accèdent à ce système selon leurs profils (agent, technicien ou administrateur) via deux types d'interfaces : Web (Navigateur) et application Java (client lourd).

Cependant, nous constatons l'absence d'autres systèmes importants pour la gestion de la maintenance tels que : SCADA (supervision industrielle) et la gestion des ressources (ERP).

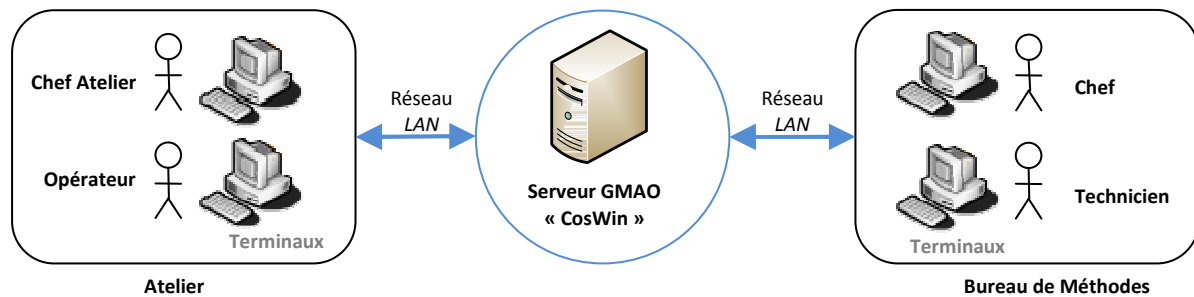


Figure 4.3 : Système de Gestion de maintenance existant

d) *Relations entre le service de maintenance et les autres services* : Le service de maintenance au niveau du complexe Cevital dépend de plusieurs autres services du complexe (figure 4.4), notamment le service de production, les achats, gestion des stocks, etc. Par ailleurs, le service de maintenance peut être en relation avec des partenaires externes à l'entreprise (fournisseurs de pièce de rechange, sous-traitant, etc.). Notons enfin, qu'il n'existe pas de système informatique permettant de gérer la communication entre ces différents services.

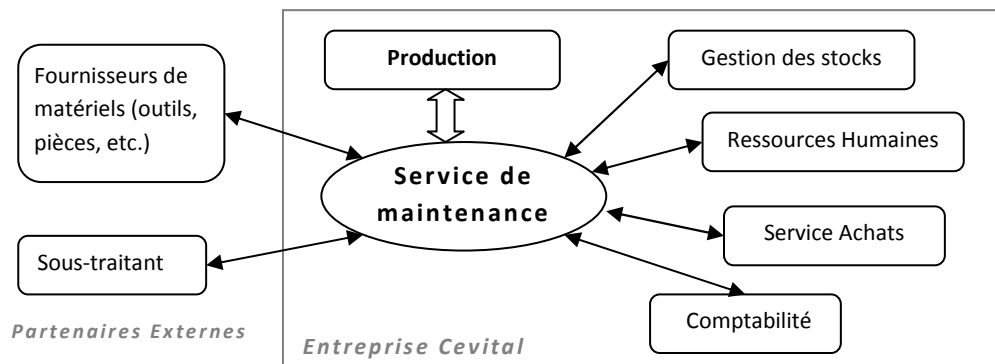


Figure 4.4 : Relations entre le service de maintenance et les autres services

Dans ce qui suit, nous présentons l'application des différentes étapes relatives à l'analyse métier.

Décomposition du domaine

Cela consiste à décomposer le domaine de la maintenance en plusieurs domaines fonctionnels en déduisant ainsi un ensemble de processus métier. La figure 4.5 montre une vue simplifiée du modèle de décomposition du domaine (ou cartographie fonctionnelle du système). Par ailleurs, nous avons mis en avant dans ce modèle, les processus métier relatifs aux opérations de maintenance correctives et les domaines qui sont en relation directe avec ces processus.

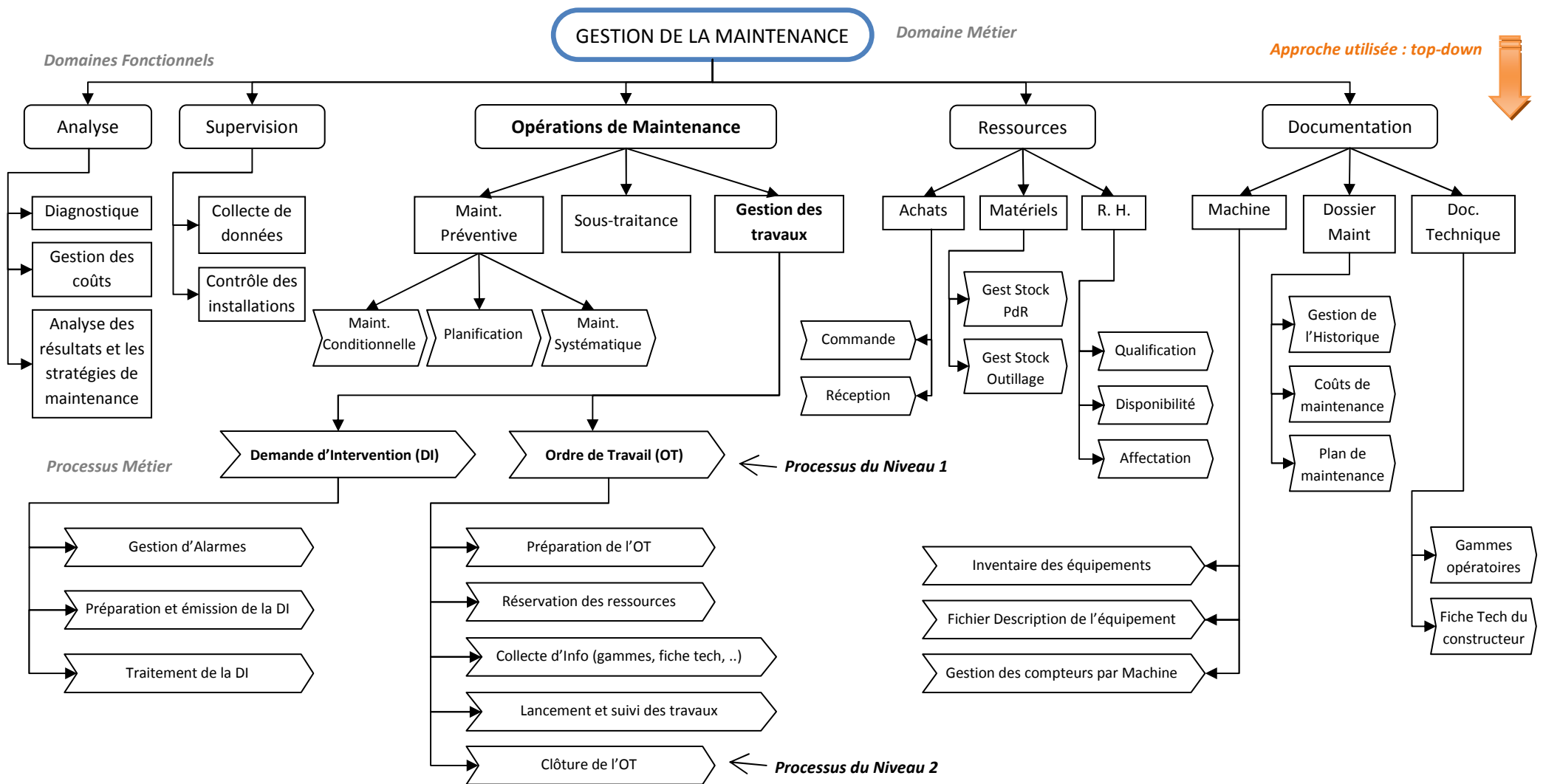


Figure 4.5 : Modèle de décomposition du domaine métier « maintenance industrielle »

Identification des processus

Suite à la décomposition du domaine métier relatif à la maintenance au niveau de l'unité de conditionnement, nous avons mis en avant un ensemble de processus métier de différents niveaux d'abstraction. En effet, les processus métier du niveau 1 (figure 4.5) représentent ceux du plus haut niveau d'abstraction pouvant être directement exposés (sous forme de services Web) aux partenaires externes (opérateurs de maintenance, applications distante, services client, etc.). En outre, les processus métier du niveau 2 (figure 4.5) correspondent aux sous-processus au sein du processus métier racine.

La différence entre ces différents niveaux de processus réside principalement dans la granularité (plus on remonte dans les niveaux, plus la granularité devient plus forte) et aussi en termes de dépendances entre les processus du même niveau d'abstraction. En effet, le processus métier « *ordre de travail* » (figure 4.5), ne dépend pas forcément du processus du même niveau « *demande d'intervention* », car on peut lancer un OT sans avoir reçu une DI, c'est le cas de la maintenance préventive. Par contre pour les sous-processus, généralement fortement dépendants entre-eux, doivent suivre un ordre d'exécution bien spécifique (selon les règles métier).

A titre illustratif, nous limitons notre champ d'étude aux deux processus relatifs aux traitements de l'OT et la DI (figure 4.5). Le tableau 4.1 montre une description de ces processus.

Processus métier racine	Sous-processus	Partenaires impliqués	Description
<i>Demande d'Intervention (DI)</i>	1- Gestion d'alarmes 2- Préparation et émission de la DI 3- Analyse de la DI	- Opérateurs de terrain (maintenance, supervision, etc.). - Opérateurs Bureau de Méthodes - Systèmes SCADA et GMAO	Englobe les activités de traitement d'alarmes (identification de la machine, du composant, etc.), édition et émission de la DI et enfin l'analyse de la DI (valider, différer, etc.) au niveau du bureau de méthodes (BM).
<i>Ordre de Travail (OT)</i>	1- Préparation OT 2- Réservation des ressources 3- Collecte d'Info 4- Lancement et suivi ; 5- Clôture de l'OT	- Opérateurs du Bureau de Méthodes et gestionnaires (magasinier, agent des RH, etc.) - Systèmes GMAO, Documentation, SCADA et ERP (gestion stock magasin, gestion des ressources humaines, etc.)	L'OT représente le noyau de la gestion de la maintenance. Ce processus consiste à initialiser l'OT, réserver les ressources (matérielles et humaines), collecter les gammes (tâches) opératoires, le lancement et le suivi des travaux et enfin, la sauvegarde de toutes les opérations réalisées.

Tableau 4.1 : Description des processus métier identifiés

Description des scénarios métier relatifs à la maintenance

La définition et la description des scénarios métier est une étape primordiale dans toute démarche de développement logiciel. Nous présentons dans ce qui suit, un scénario type

relatif à la maintenance industrielle. Précisément, nous nous intéressons à la maintenance corrective appartenant au domaine fonctionnel : « *opérations de maintenance* », (figure 4.5).

Scénario métier : « gestion de la maintenance corrective »

Ce scénario correspond aux deux processus métier déjà présentés (« *demande d'intervention* » et « *ordre de travail* »). Ce scénario décrit les tâches principales, depuis l'interception de l'alarme par le système SCADA jusqu'à la clôture de l'OT suite à la fin des travaux. Le scénario est le suivant :

- 1- Suite à l'apparition d'une anomalie au niveau d'une station de pompage (figure 4.2), une alarme est envoyée par le capteur présent sur la machine vers le système SCADA,
- 2- L'opérateur au centre de supervision prend connaissance de la notification et décide suite à l'analyse de l'alarme, d'émettre une DI vers les techniciens du bureau de méthodes (BM),
- 3- Pour préparer la DI, l'opérateur récupère les informations auprès du système SCADA sur la panne et l'équipement concerné. Ensuite, l'opérateur envoie la DI au BM (via la GMAO),
- 4- Le technicien du BM prend connaissance de la DI et lance l'outil de diagnostic de pannes. Ce dernier récupère les informations sur l'état de la machine et génère un diagnostic type,
- 5- Après avoir analysé et modifié le diagnostic, le technicien du BM décide alors de lancer un OT en priorisant la DI en question,
- 6- Le système GMAO initialise l'OT par la récupération de la documentation technique et les gammes opératoires sur le type de la machine auprès du système de documentation,
- 7- Ensuite, le technicien complète l'OT par la modification des gammes types récupérées et procède à la réservation des différentes ressources (pièces de rechanges, outils, etc.) auprès du progiciel ERP,
- 8- Une fois l'OT complété, le technicien lance son édition et l'envoi aux responsables sur le terrain (chefs d'atelier, chefs d'équipe, etc.). De leur part, ces derniers distribuent les « *bons de travaux* » (BT) aux opérateurs de maintenance en leur affectant les tâches correspondantes,
- 9- Le(s) responsable(s) de l'intervention envoient périodiquement une « *fiche de pointage* » (FP) indiquant l'état d'avancement des travaux,
- 10- Une fois la situation est dénouée, le responsable de l'intervention renseigne et envoie d'un « *rapport d'intervention* » (RI) au BM précisant tous les travaux techniques réalisés, les ressources consommées et les résultats de l'intervention (tests),
- 11- Le technicien du BM (ou chef du BM) prend connaissance du RI et décide de clôturer l'OT en sauvegardant dans l'historique (système de documentation) toutes les opérations réalisées et éventuellement pour une analyse ultérieure sur la qualité de l'intervention,
- 12- Après avoir clôturé l'OT, le technicien du BM informe l'opérateur du centre de supervision (émetteur de la DI) de la remise en marche de la machine.

La figure 4.6 montre le modèle d'interactions (sous forme de diagramme de séquences) relatif au scénario précédent.

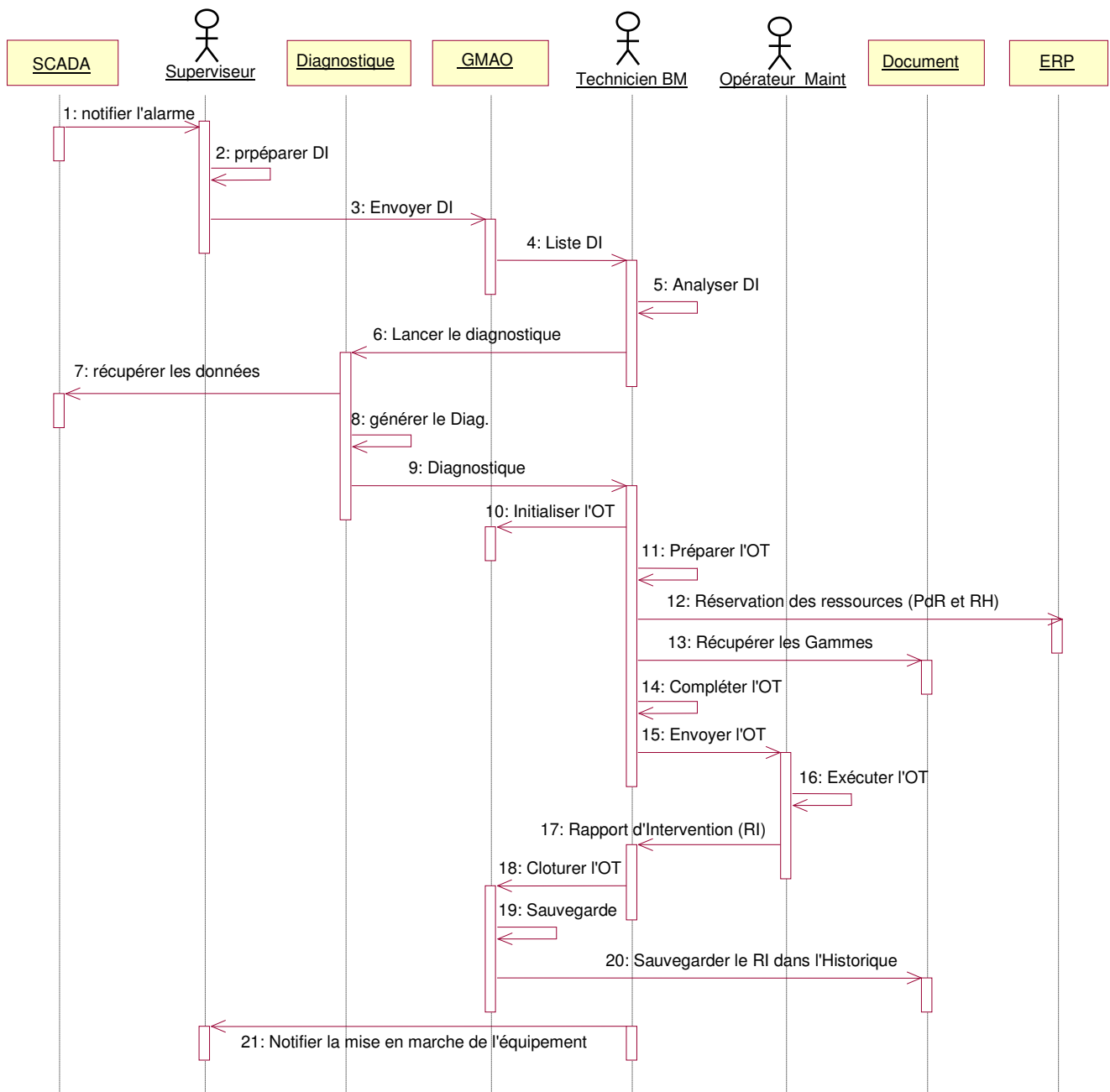


Figure 4.6 : Modèle d'interactions du scénario métier « *gestion de la maintenance corrective* ».

Après avoir décomposé le domaine métier et décrit un scénario métier relatif à la maintenance corrective, nous allons, dans l'étape suivante, déduire un ensemble de cas d'utilisation métier en se référant au modèle de cartographie du système (figure 4.5).

Identification des cas d'utilisation métier

Nous présentons, dans ce qui suit et à titre illustratif, un ensemble de cas d'utilisation métier (figure 4.7) issue des étapes d'analyse précédentes en spécifiant leurs domaines fonctionnels et processus métier dont ils sont impliqués. En effet, il s'agit de projeter les cas d'utilisation métier à identifier par rapport aux domaines fonctionnels et les processus métier (ou les phases) comme déjà définie dans l'approche SOMA d'IBM (Chapitre 3, §III.1).

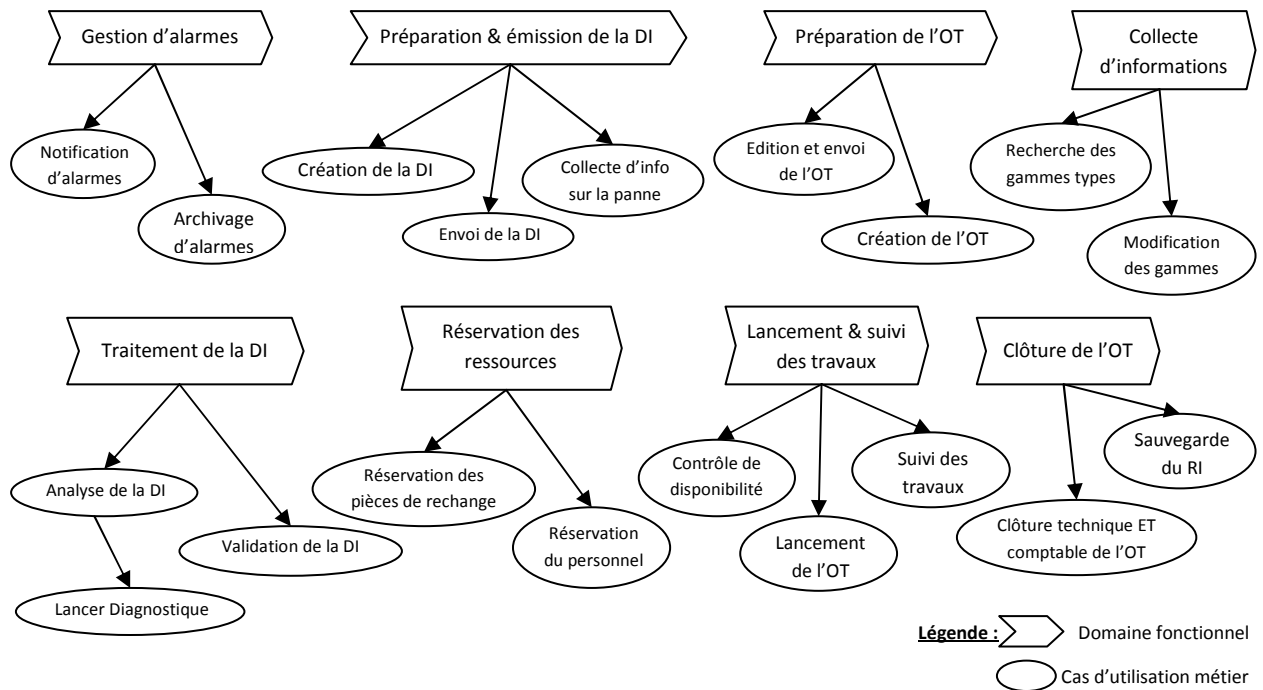


Figure 4.7 : Identification des cas d'utilisation métier

II.2. Etape 2 : Analyse des utilisateurs et leurs tâches

Nous conduisons dans cette étape à travers deux sous étapes : (1) analyse des utilisateurs et (2) modélisation des tâches. Ces activités visent à fournir les éléments importants pour la spécification orientée services de l'interface homme machine qui sera abordée lors de la phase 3. Dans ce qui suit, nous présentons ces sous étapes.

Analyse des utilisateurs

Dans cette étape, nous nous intéressons aux acteurs humains concernés par le contexte métier existant de l'entreprise, en déduisant ainsi toutes les informations concernant leurs profils et leurs besoins métier. Ces informations seront utiles lors de la spécification d'IHM relative au système de maintenance industrielle. La figure 4.8 illustre la typologie des acteurs intervenant dans la maintenance au niveau de l'unité de conditionnement. La définition des rôles ainsi que les besoins de chacun de ces acteurs est donnée en annexe (voir annexe G).

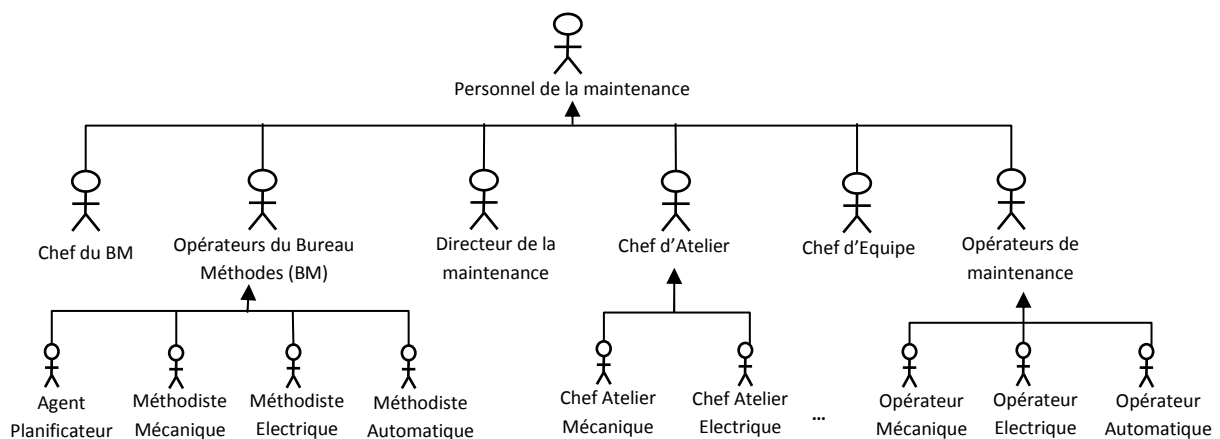


Figure 4.8 : Typologie des acteurs intervenant dans le processus métier de la maintenance

Cependant, d'autres acteurs humains appartenant aux différents départements (commercial, comptabilité, etc.) sont aussi impliqués mais d'une manière indirecte dans les processus métier relatif à la gestion de la maintenance. Néanmoins, pour des raisons de simplicité, ce type d'acteurs n'est pas traité dans ce mémoire.

A l'issue de cette analyse, nous pouvons compléter le modèle de cas d'utilisation métier (figure 4.9) en spécifiant cette fois-ci, tous les acteurs impliqués ainsi que les cas d'utilisation métier issus des scénarios relevant des activités de maintenance au niveau de l'unité.

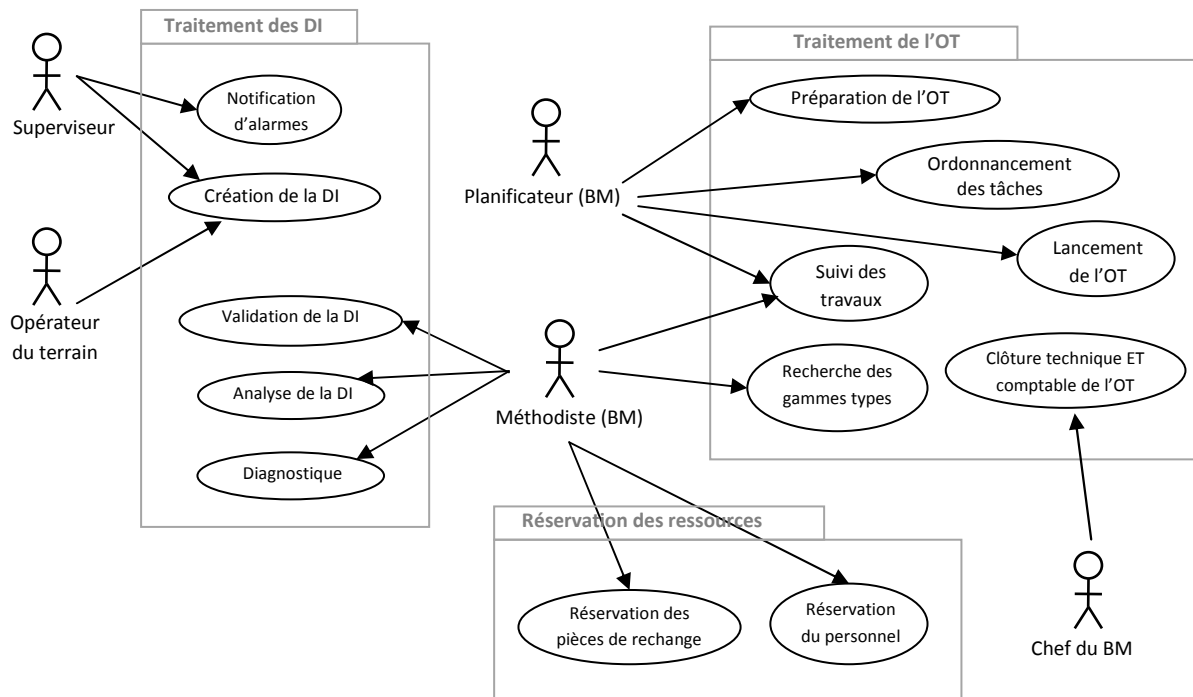


Figure 4.9 : Modèle des cas d'utilisation métier relatif au domaine « opérations de maintenance »

Modélisation des tâches

Au cours de cette étape, nous considérons l'étude du système de maintenance sous l'angle de l'analyse des tâches. Rappelons que cette analyse vise à comprendre comment les utilisateurs atteindront leurs objectifs métiers en exécutant une succession de tâches (et sous-tâches). Cette étude nous amène à élaborer un « modèle de tâches » reflétant les tâches des utilisateurs ainsi que leurs interdépendances. Il s'agit d'abord de considérer chaque sous-système (domaine fonctionnel) issu de la décomposition du domaine métier (figure 4.5) qui sera présenté sous forme de modèle de cas d'utilisation métier. Ce dernier sera analysé sous deux angles : (1) modélisation statique des activités et leurs flux, (2) modélisation dynamique des tâches des utilisateurs impliqués dans un sous-système.

Pour cela, et à titre illustratif, nous considérons un scénario métier relevant du domaine fonctionnel « maintenance conditionnel ». La figure 4.10 illustre le modèle de cas d'utilisation métier correspondant.

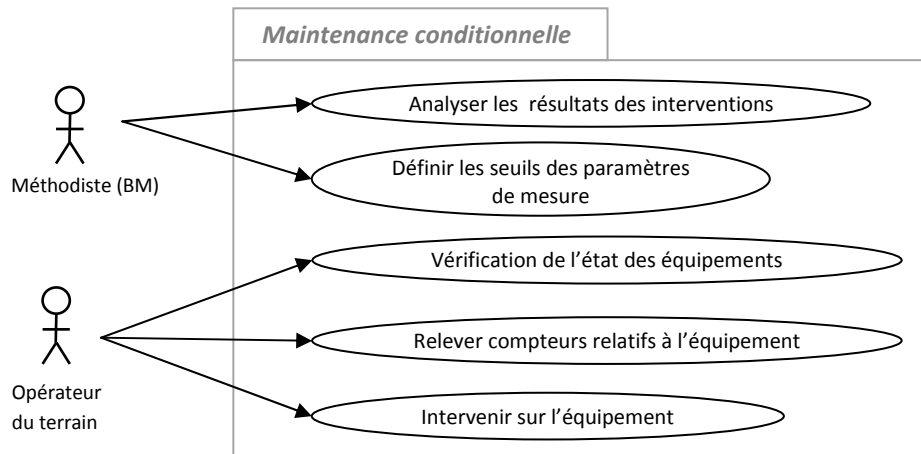


Figure 4.10 : Modèle des cas d'utilisation métier relatif au scénario « maintenance conditionnelle »

Le scénario consiste à la surveillance de l'état des machines en vérifiant les données prélevées à partir des compteurs de la machine par rapport aux seuils prédéfinis par les méthodistes (ou experts) du BM (bureau de méthodes). Suite au dépassement d'un seuil lié à un paramètre de mesure, une intervention est déclenchée sur le lieu d'incident afin de rétablir l'équipement concerné en situation normale.

Les seuils définis dans le cadre de la maintenance conditionnelle peuvent être modifiés par les méthodistes du BM suivant leurs analyses effectuées sur les résultats d'interventions (renseignés sur le RI).

1) Modélisation statique des activités (SADT)

Cela consiste à la structuration du domaine fonctionnel sous forme d'activités en utilisant la méthode de spécification fonctionnelle des systèmes complexes SADT (*Structured Analysis and Design Technique* ou Technique Structurée d'Analyse et de Modélisation de Systèmes). Dans une analyse SADT, on peut modéliser deux types d'analyse [Tajri, 05]. L'analyse par des « actigrammes » (boîtes d'action) et l'analyse par des « datagrammes » (boîtes de donnée). Sur des actigrammes, les actions sont reliées entre elles par des flux de données alors que les datagrammes se sont les données qui sont reliées entre-elles par des flux d'activité (une description sur SADT est donnée en annexe E).

Dans notre cas, nous nous intéressons à modéliser les domaines fonctionnels (ou sous-systèmes) suivant les actigrammes. La figure 4.11 illustre le modèle statique d'activités correspondant au sous-système « maintenance conditionnelle ».

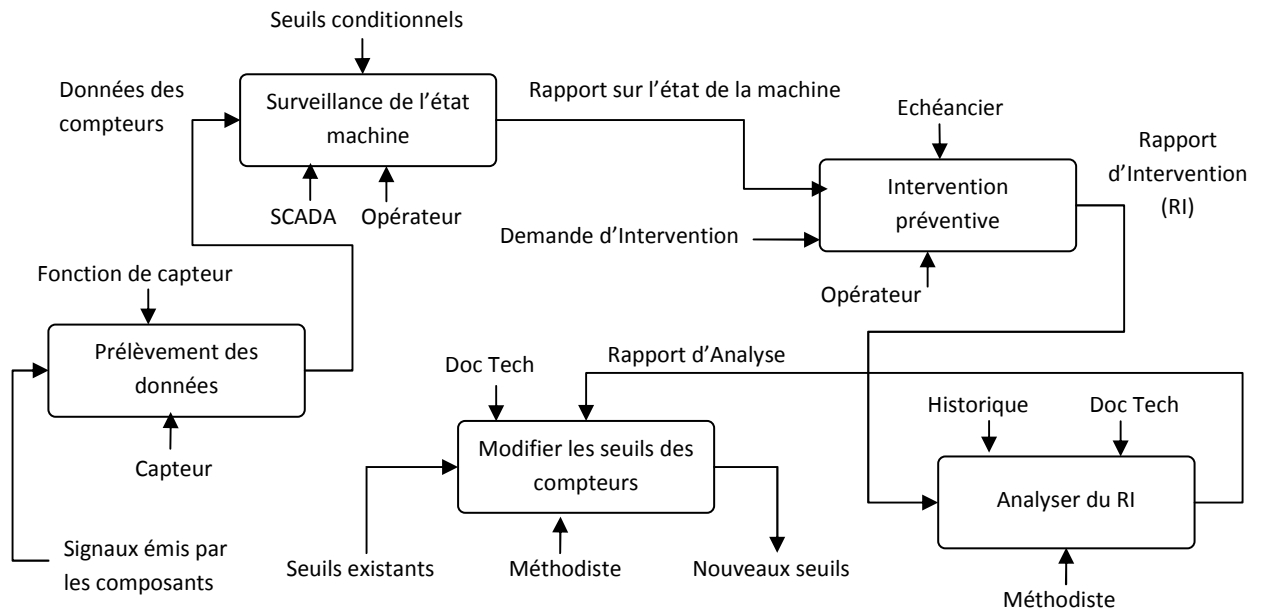


Figure 4.11 : Modèle d'activités SADT correspondant au sous-système « maintenance conditionnelle »

2) Modélisation dynamique des tâches des utilisateurs (CTT)

Après avoir découpé chaque sous-système en ensemble d'activités caractérisées par leurs différents types de flux de données (entrées, de contrôle, etc.), nous allons dans cette étape affiner notre analyse. Nous intéressons cette fois-ci à la manière dont les utilisateurs procéderont pour réaliser les objectifs correspondants aux activités déjà spécifiées par SADT.

Cette analyse s'appuie sur le formalisme CTT (*Concur Task Trees*) [Paterno, 00] qui permet de décrire plusieurs types de tâches (utilisateur, application, abstraite ou interaction) sous forme d'un « arbre de tâches ». En outre, le formalisme CTT définit des opérateurs temporels (activation, synchronisation, etc.) permettant de relier les tâches. Une description sur le formalisme CTT est donnée en annexe (annexe E). La figure 4.12 montre un extrait de l'arbre de tâches correspondant au sous-système « maintenance conditionnelle ».

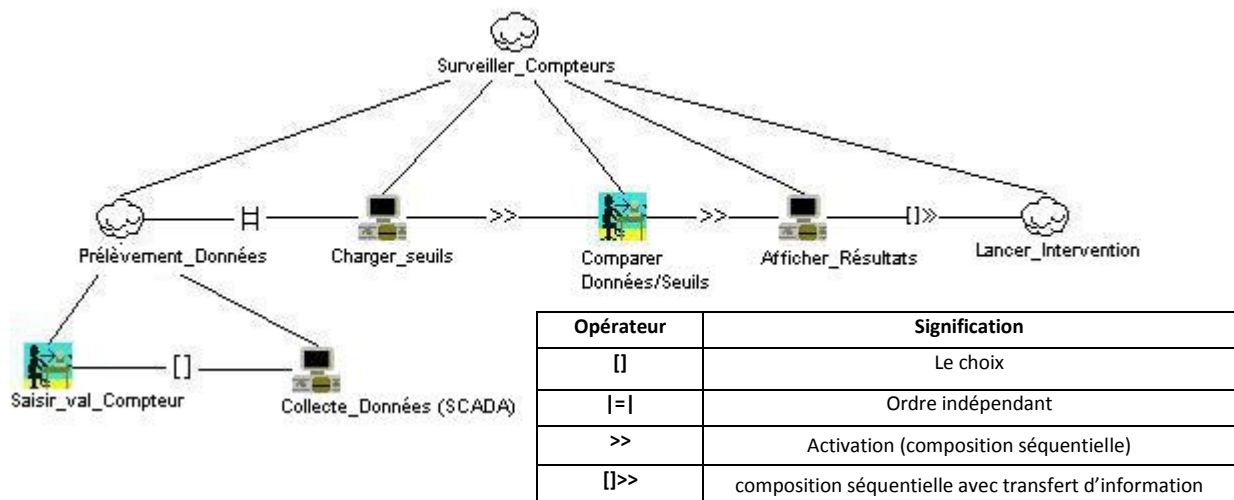


Figure 4.12 : Extrait du modèle CTT du sous-système « maintenance conditionnelle »

L'objectif de décomposition des tâches utilisateurs est de décrire et de construire des services Web de différentes granularités destinés à la réalisation de l'IHM. Dans l'exemple précédent (figure 4.12), on peut considérer la tâche abstraite « *vérifier seuils* » comme un service Web composé d'autres services d'une granularité inférieure et qui sont orientés présentation, tels que : *collecte de données, charger les seuils et comparaison*.

II.3. Conclusion sur la phase 1

Dans cette première phase, nous avons étudié les différents aspects organisationnels et architecturaux liés à l'activité de la maintenance au sein de l'unité de conditionnement des huiles (groupe *Cevital*). L'objectif majeur visé par cette phase est de comprendre la structure du système de maintenance existant afin d'extraire les fonctionnalités et les cas d'utilisation métier. Pour cela, nous avons mené cette phase à travers deux étapes d'analyse.

La première étape consiste à l'analyse du domaine métier qui vise à décomposer et à cartographier le système de maintenance existant sous forme de plusieurs domaines fonctionnels, processus métier de différents niveaux et les cas d'utilisation métier. Ces derniers sont souvent considérés comme de bons candidats pour les services métier à exposer finalement sous forme de services Web aux consommateurs (utilisateurs, applications, etc.).

La seconde étape vise à identifier tous les types d'utilisateurs à travers leurs profils et leurs rôles dans la fonction de la maintenance. Par ailleurs, nous menons dans cette étape une analyse sur les tâches des utilisateurs afin d'identifier de nouveaux besoins métier concernant les utilisateurs du futur système de la maintenance industrielle.

A la fin de cette phase, nous obtenons une liste de processus métier, de cas d'utilisation métier ainsi qu'un premier ensemble de services métier candidats. Tous ces éléments seront d'avantage exploités lors de la seconde phase pour dégager et identifier de nouveaux services métier supportant les objectifs majeurs de l'organisation de la maintenance.

III. Phase 2 : Analyse et expression des besoins

Dans cette phase, nous procédons à un second niveau d'analyse à travers lequel on identifie les différents services métier candidats et leurs catégories, les besoins des acteurs humains et enfin l'analyse de l'existant permettant d'extraire sous forme de services métier les fonctionnalités offertes par les applications existantes. Ces étapes sont décrites ci-après.

III.1. Etape 1 : Identification et catégorisation des services métier

Cette étape consiste à identifier les services métier candidats, regrouper et les classer sous forme de « catégories de services métier » reflétant la décomposition du futur système de maintenance en un ensemble de « composants d'entreprise » réalisant les différents services.

Identification des services métier

Après avoir identifié « *les cas d'utilisation métier* » lors de la première phase d'analyse, dans cette étape nous transformons ces cas d'utilisation en un ensemble de « services métier candidats » pour une éventuelle exposition sous forme de services Web. Ces services sont

identifiés en se basant sur le modèle de cas d'utilisation déjà présenté (figure 4.9) et en tenant compte du modèle d'interactions (figure 4.9) relatif au scénario métier déjà décrit.

Le tableau 4.2 décrit un ensemble de services métier candidats précisant les acteurs demandeurs du service (consommateurs) et les sous-systèmes (composants d'entreprise) fournissant la réalisation de ces services. Cependant, notons que cette liste de services n'est pas exhaustive et concerne uniquement le domaine fonctionnel « opérations de maintenance » relatif à la gestion des DI (**Demande d'Intervention**) et de l'OT (**Ordre de Travail**).

Nom du service métier candidat	Processus Métier	Description du service métier	Demandeur	Fournisseur
Notifier l'alarme	DI	Ce service a pour rôle d'identifier l'alarme envoyée par le capteur situé sur le site d'incident et d'afficher les données relative à la panne (équipement, composant, etc.).	Systèmes contrôle (capteur, automate, etc.) Opérateurs sur le terrain.	Système de contrôle et de supervision (SCADA)
Sauvegarder l'alarme	DI	Enregistrer les informations relatives à l'alarme dans l'historique de la machine concernée.	SCADA Superviseur.	Documentation (Historique)
Création de la DI	DI	Assister l'opérateur (superviseur, technicien en maintenance, etc.) dans la création et l'édition de la DI.	Superviseur. Opérateurs sur le terrain.	Gestion de la maintenance (GMAO)
Lancer le diagnostique	DI	Permet d'invoquer les utilitaires du diagnostique de pannes afin d'assister l'opérateur dans le diagnostique de la panne.	Méthodiste (BM)	Système de Diagnostique
Préparation de l'OT	OT	Permet de créer un OT en affectant la DI à l'OT. Il s'agit	Planificateur (BM)	GMAO
Ordonnancer les tâches	OT	Permet de créer et modifier le plan d'ordonnancement pour les tâches de maintenance à exécuter relatives à l'OT.	Planificateur (BM)	GMAO
Recherche des gammes type	OT	Fourni les services de recherche des gammes opératoires fournies par le constructeur ou par un expert et qui sont relatives au type de la machine ainsi que la panne détectée.	Méthodiste	Documentation (Doc Tech.)
Renseigner le BT	OT	Permet de modifier les gammes types récupérées et de préciser les travaux à réaliser suivant le diagnostique établi.	Méthodiste	GMAO
Réservation des matériels	OT	Fourni les services de réservation des pièces de rechange auprès du magasin ainsi que les outils nécessaires.	Planificateur Méthodiste	Gestion de stock (ERP)
Réservation du personnel	OT	Fourni les services de réservation des opérateurs qualifiés, selon leurs disponibilités, pour réaliser les travaux de l'OT.	Planificateur Méthodiste	Gestion des R.H. (ERP)
Lancer l'OT	OT	Permet d'assigner le début des travaux après avoir vérifier la disponibilité des ressources et autres conditions (permis, etc.)	Planificateur Chef du BM	GMAO
Suivi des travaux	OT	Consiste à collecter les Fiche de Pointage (FP) indiquant l'état d'avancement et leurs sauvegardes.	Planificateur Méthodiste	GMAO
Clôture de l'OT	OT	Permet d'assigner la fin des travaux et l'enregistrement de toutes les opérations effectuées (RI) dans l'historique.	Planificateur Chef du BM	GMAO et Documentation

Tableau 4.2 : Liste représentative des services métier (candidats) identifiés.

Catégorisation des services métier

La catégorisation des services est une étape clé dans la démarche SOA. En effet, cette catégorisation permet de spécifier avec précision les différents composants logiciels qui forment l'architecture SOA et aussi sert de base pour la découverte de nouveaux services de forte granularité notamment les services composés et les services CRUD pour la gestion des données persistantes au niveau des bases de données.

La figure 4.13 montre un modèle de catégorisation de services simplifié relatif aux services métier candidats identifiés lors de l'étape précédente (tableau 4.2).

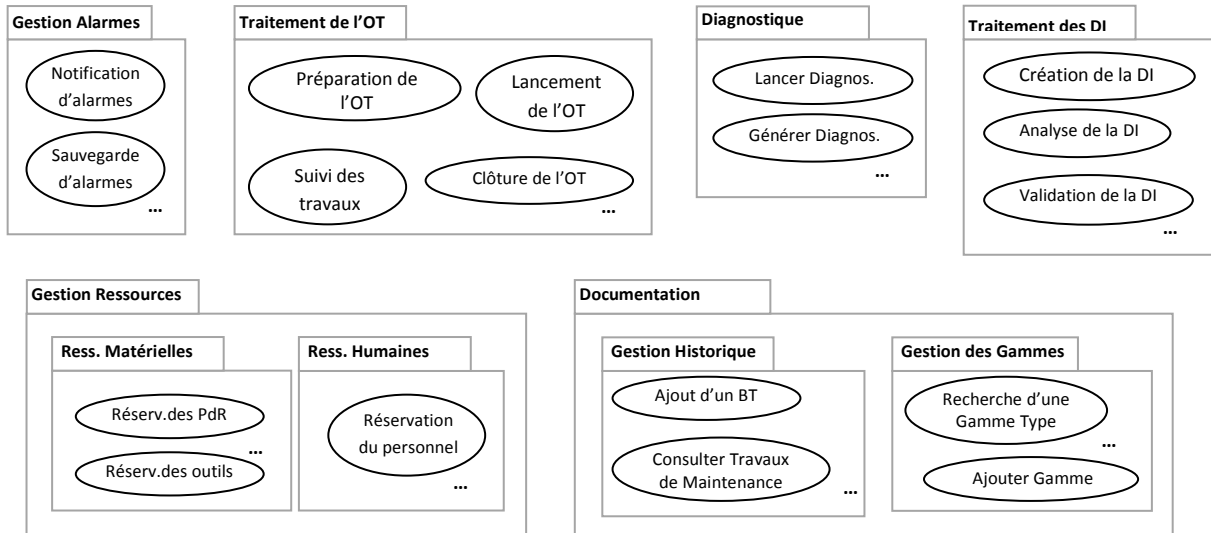


Figure 4.13 : Catégorisation des services métier

Dans ce qui suit, nous présentons quelques éléments clés issus de la catégorisation des services métier. Ces éléments reflètent d'une part, l'identification des services dédiés à la gestion des objets métier notamment les services CRUD (chapitre 2, §II.3) comme illustré sur la figure 4.14 et de l'autre part, la composition des services métier possédant une granularité moyenne ou petite pour former des services dotés d'une forte granularité (figure 4.15).

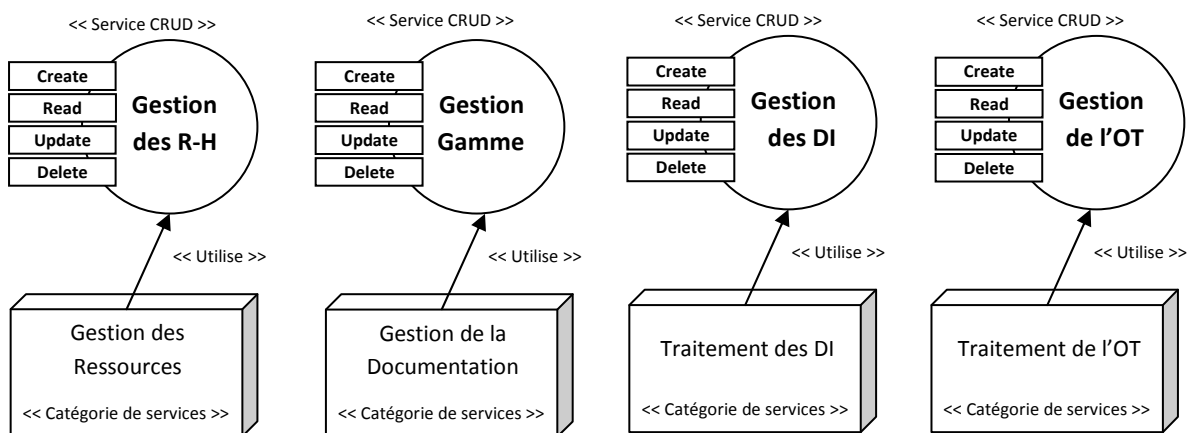


Figure 4.14 : Identification des services CRUD (*Create, Read, Update, Delete*)

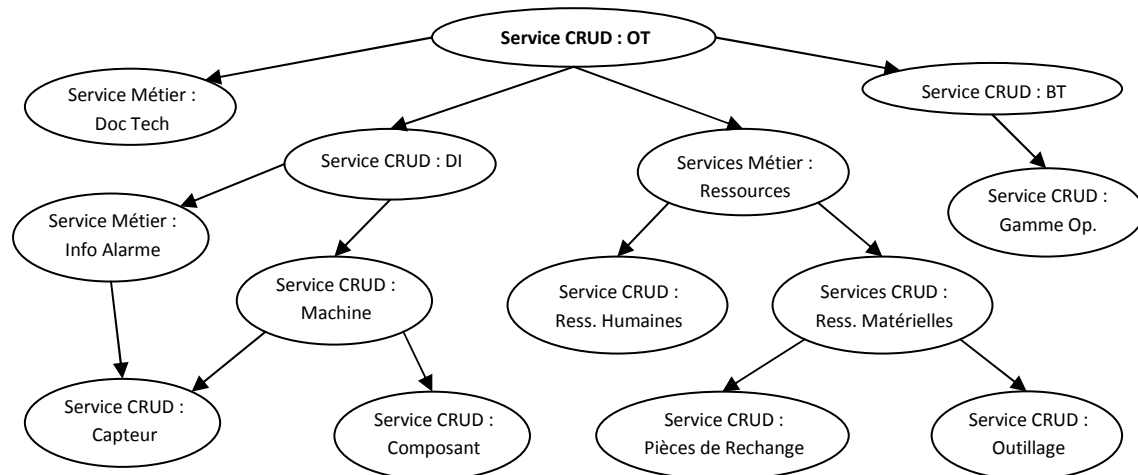


Figure 4.15 : Exemple de composition des services à base de la catégorisation

Comme illustré sur la figure 4.15, le service de gestion des données sur l'OT (Ordre de Travail) est composé de plusieurs services de type métier et CRUD (*create, read, update, delete*). De leur part, chacun de ces services peut être composé d'autres services d'une granularité inférieure. Nous obtenons ainsi un arbre de services de différentes granularités.

III.2. Etape 2 : Identification des besoins des acteurs métier

Cette étape consiste à identifier et à définir les services métier en tenant compte des interactions entre les différents acteurs métier et les processus métier déjà identifiés. Nous menons cette étape à travers trois activités d'expression des besoins : (1) en termes services Web, (2) en termes de mobilité des acteurs humains et (3) en termes de tâches coopératives.

Expression des besoins en termes de services Web

Cette étape consiste à exprimer les besoins déjà recensés à travers les cas d'utilisation métier en termes de services Web en spécifiant leurs interactions avec les utilisateurs et aussi déduire les opérations définissant le service Web. En effet, il s'agit de modéliser chaque cas d'utilisation métier sous forme d'un diagramme de séquences. Les activités peuvent définir les services Web et les échanges de messages peuvent alors correspondre aux opérations dans ces services.

Dans ce qui suit, nous considérons deux cas d'utilisation métier afin de mieux montrer l'expression des besoins métier en termes de services Web. Le premier cas d'utilisation concerne : « *préparation de gammes* » et le second est relatif à la « *réservation des ressources matérielles* ».

Ces deux cas d'utilisation métier sont décrits puis exprimés sous forme de diagramme de séquences. Les figures (4.16, 4.18) et (4.17, 4.19) montrent respectivement la description du cas d'utilisation selon le modèle défini dans [Idoughi, 08] et l'expression des services Web métier issu du cas d'utilisation métier à travers un diagramme de séquences.

Intitulé du cas d'utilisation : Préparation des gammes
Description
Suite à la création de l'OT, le méthodiste (BM) procède à la recherche des gammes opératoires qui correspondent au type de la machine et les composants renseignés sur la DI. Pour cela, il consulte les gammes définies par le constructeur et celle déjà établi par les techniciens du BM afin d'identifier les opérations à effectuer sur l'équipement, les moyens nécessaires (outils, pièces, etc.) et les profils des intervenants.
Détail du cas d'utilisation
<ul style="list-style-type: none"> - But du cas d'utilisation : renseigner le bon de travaux (BT ou OT) à envoyer aux opérateurs sur le terrain. - Pré-conditions : réception et validation de la DI. - Post-conditions (succès) : les opérations ainsi que les ressources nécessaires sont identifiées. - Post-conditions (échec) : créer ou mettre à jour une gamme opératoire relative au type d'équipement. - Événement déclencheur : réception d'une DI prioritaire.
Scénario nominal
<p>Enchainements :</p> <ol style="list-style-type: none"> 1. En fonction de l'analyse de la DI, le méthodiste (BM) décide de créer un OT relatif à la DI reçue, 2. Il lance la recherche de la gamme en spécifiant l'identifiant de l'équipement (et/ou du composant) en question, 3. Après la réception de la gamme type, le système crée automatiquement un BT correspondant aux travaux et aux ressources nécessaires mentionnées sur la gamme type. 4. Afin de compléter ou modifier la gamme type obtenue, le méthodiste consulte l'historique des interventions effectuée sur l'équipement désigné et analyse les résultats de l'historique, 5. Après avoir modifié et complété le BT, il procède à sa sauvegarde et à son édition afin de le distribuer sur les responsables (chef d'atelier, chef d'équipe, etc.) de la maintenance présentent sur le terrain. <p>Acteurs impliqués : méthodiste du BM.</p>
Scénario alternatif

Figure 4.16 : Description du cas d'utilisation « préparation des gammes »

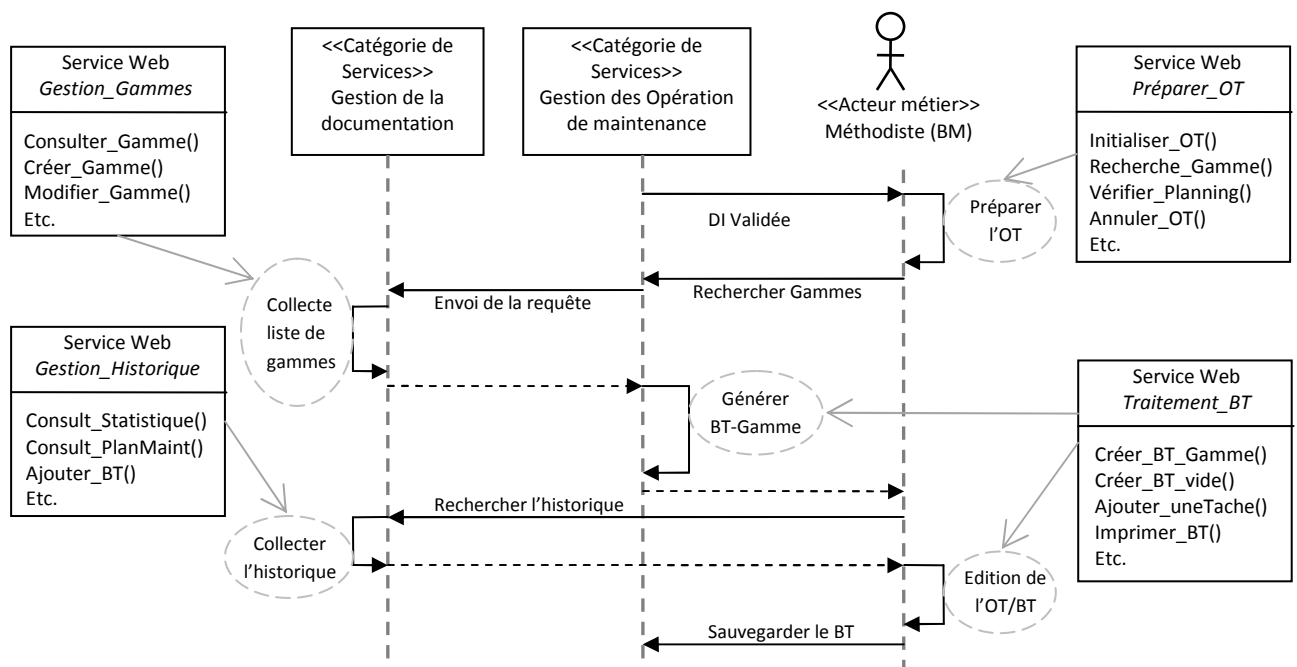


Figure 4.17 : Expression en terme de services Web du cas d'utilisation : « préparation des gammes »

Intitulé du cas d'utilisation : Réservation des ressources matérielles
Description
Le planificateur lance une demande de fourniture (DF) afin de réserver les ressources nécessaires à l'intervention en termes de pièces de rechange. A la réception de la DF, le magasinier vérifie le stock et procède à l'approvisionnement. A la fin de cette opération, le magasinier confirme la disponibilité des ressources auprès du BM.
Détail du cas d'utilisation
<ul style="list-style-type: none"> - But du cas d'utilisation : réservation des ressources matérielles nécessaires pour l'intervention. - Pré-conditions : création et validation de l'OT. - Post-conditions (succès) : ressources matérielles demandées sont disponibles. - Post-conditions (échec) : ressources matérielles non disponibles, le magasinier lance alors une DA. - Événement déclencheur : réception d'une DI prioritaire.
Scénario nominal
<p>Enchaînements :</p> <ol style="list-style-type: none"> 1. Le planificateur du BM crée puis envoie un DF au service de gestion stock. 2. Le magasinier vérifie la disponibilité des ressources (PdR) demandées et constate un manque. Alors il décide de lancer une DA (Demande d'Achat) auprès du service achats. 3. L'agent du service achats procède au traitement de la DA et crée un bon de commande (BC) qu'il envoie à un fournisseur. 4. Le fournisseur traite le BC, livre les pièces demandées et procède à la création et l'émission de la facture. 5. A la réception des pièces, le service achats informe le service de gestion stock de la réception des ressources demandées. Ce dernier établit alors un bon de réception et mis à jour le stock. 6. A la fin de l'opération d'approvisionnement, le magasinier informe le planificateur (BM) en validant la DF reçue. <p>Acteurs impliqués : planificateur ou methodiste du BM, magasinier, agent du service achat et le fournisseur de pièces.</p>
Scénario alternatif

Figure 4.18 : Description du cas d'utilisation « réservation des ressources matérielles »

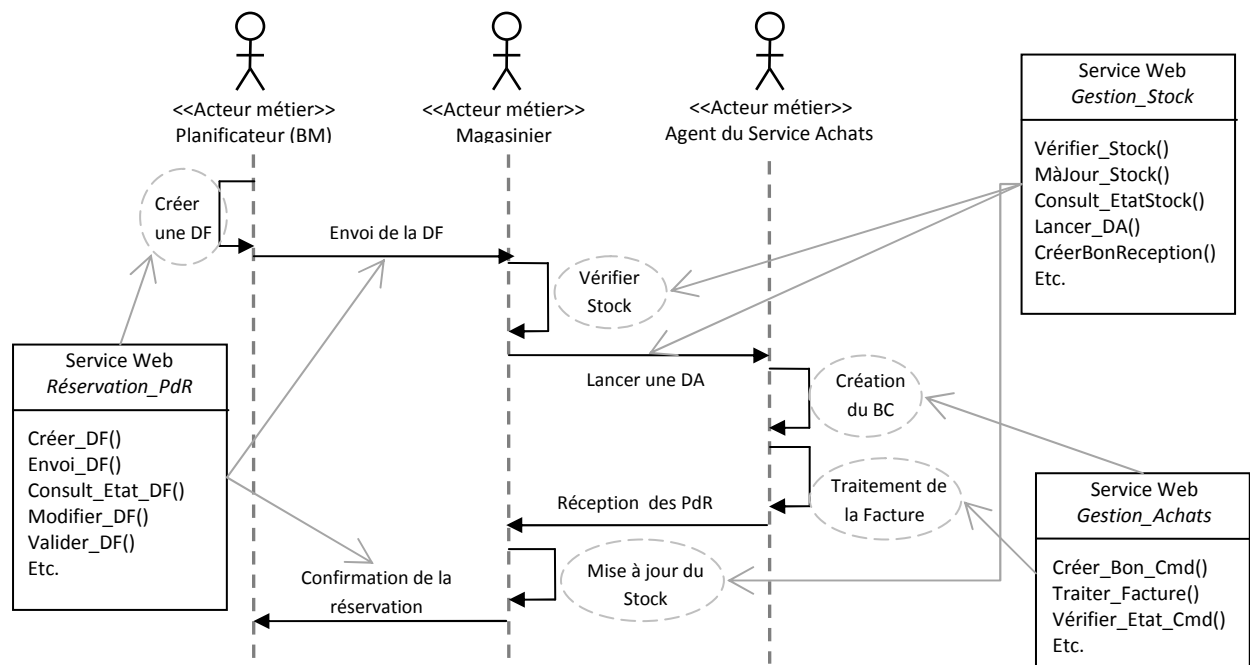


Figure 4.19 : Expression en terme de services Web du cas d'utilisation : « réservation des ressources matérielles ».

A l'issue de cette étape, nous obtenons un modèle (ou catalogue) de services métier exprimés sous forme d'un ensemble d'opérations. La figure 4.20 illustre un extrait du catalogue de services métier identifiés lors des étapes précédentes en spécifiant cette fois-ci leurs opérations. Notons qu'à ce stade d'analyse, la description des opérations n'inclue pas la spécification des paramètres (*input, output et fault*) ainsi que les types des résultats relatifs à ces opérations. Cette spécification sera traitée lors de la troisième phase (spécification de l'architecture orientée services) de la démarche.

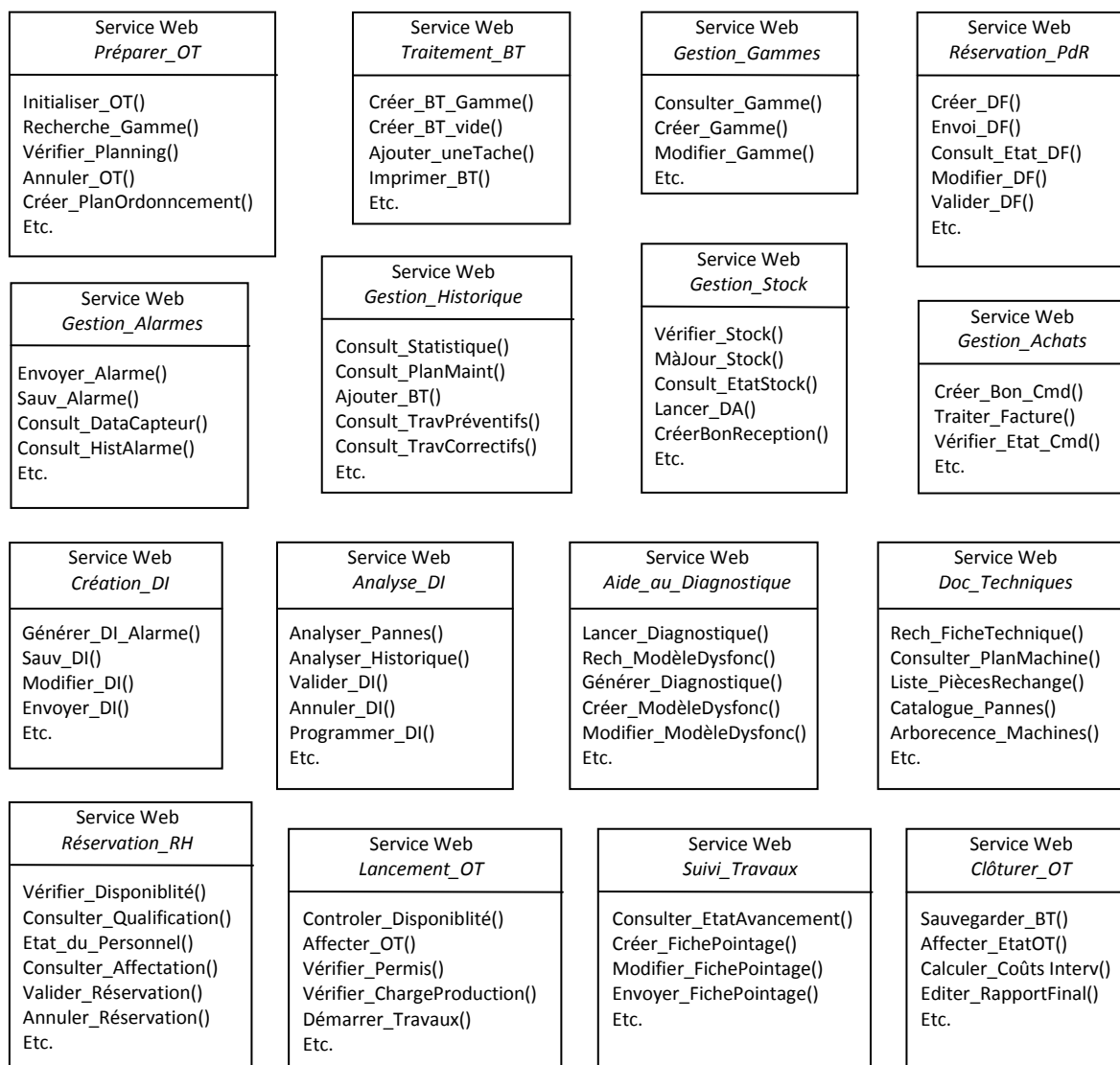


Figure 4.20 : Extrait du Catalogue des Services Métier identifiés

Expression des besoins en termes de mobilité des acteurs humains

La mobilité des acteurs humains constitue un élément important dans la fonction de maintenance industrielle. En effet, la présence des techniciens (ou experts) au niveau BM n'est pas souvent garantie, car certaines activités notamment celles liées au diagnostic, l'assistance technique dans les travaux et rondes (inspections) exigent à ces opérateurs d'être présent sur le site des installations. Par conséquent, la plateforme de la gestion de la

maintenance doit prendre en compte cet aspect de mobilité des acteurs en leur permettant d'exercer leurs tâches à distance selon leurs contexte d'usage (profils utilisateurs, leurs localisation, le type de dispositif de communication utilisé, etc.).

Dans cette étape, nous allons appliquer une démarche similaire à celle évoquée dans [Idoughi, 08]. Cette démarche consiste à identifier les scénarios de la maintenance relatifs à la mobilité des acteurs humains, leurs description sous forme de diagrammes d'interactions et en fin, déterminer les nouveaux services métier dédiés à la mobilité avec leurs opérations.

Dans ce qui suit, nous présentons un scénario représentatif (figure 4.21) dans le cadre de mobilité des acteurs humains (méthodiste BM et opérateur du terrain).

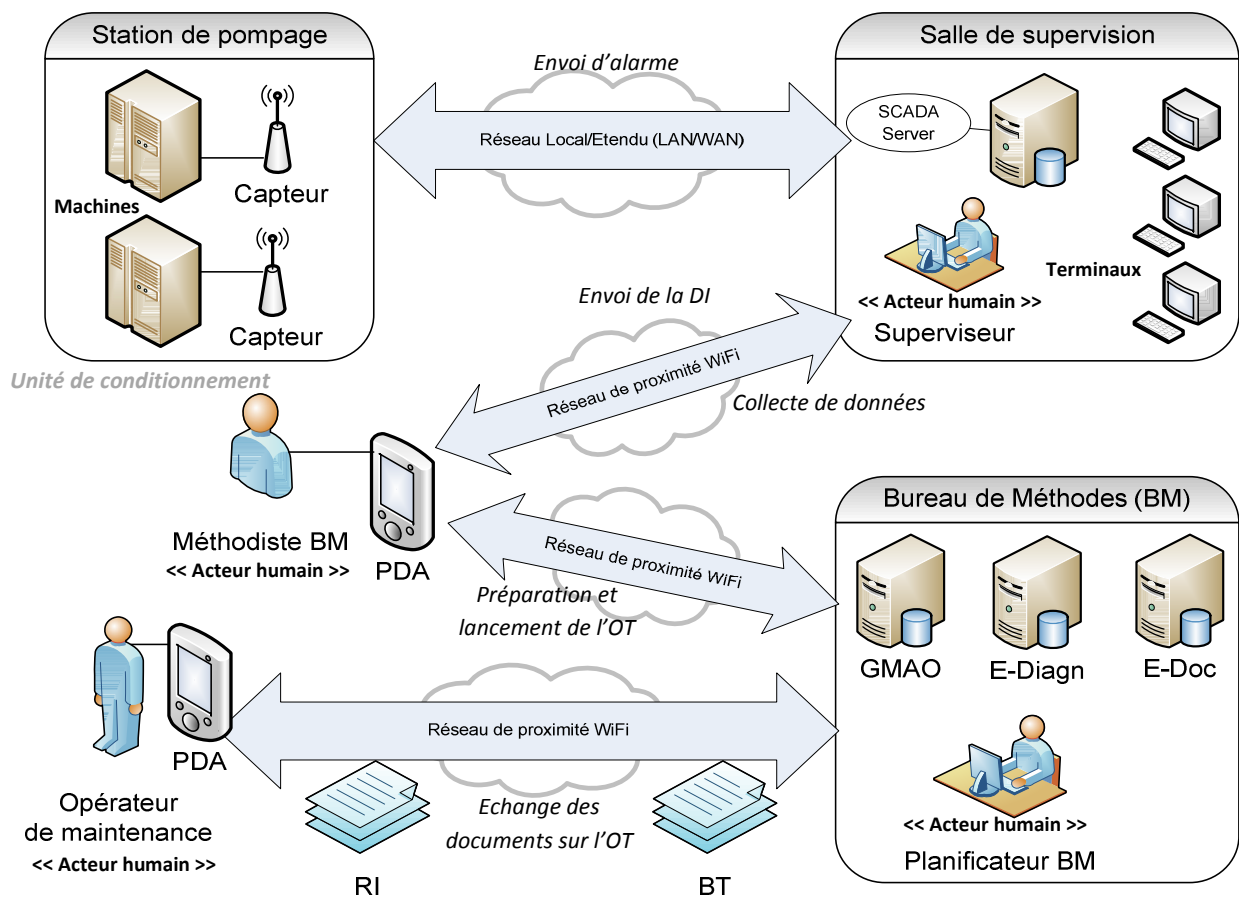


Figure 4.21 : Scénario de maintenance relatif à la mobilité des acteurs humains

La première phase de ce scénario est la détection d'anomalie et l'envoi d'alarme par les capteurs présents sur la station de pompage vers la salle de supervision. Le superviseur en chef prend connaissance de cette notification et décide de créer puis envoyer une DI vers le méthodiste du BM via le réseau de proximité Wifi. Le méthodiste reçoit la DI sur son PDA et déclenche l'outil de diagnostic (e-Diagn¹⁴) ainsi que la collecte de données au niveau du SCADA et documentation technique au niveau du serveur e-Doc¹⁵ (figure 4.21).

¹⁴ e-Diag (electronic-Diagnostic): Système d'aide au diagnostic de pannes basé sur les technologies du Web.

¹⁵ e-Doc (electronic-Documentation): Système de gestion des documents techniques basé sur les technologies du Web.

Après avoir collecté les informations nécessaires (travaux à effectuer, les ressources matérielles, etc.) il envoie une demande de création de l'OT vers le planificateur présent dans le BM (bureau de méthodes). Ce dernier procède donc à la réservation des ressources et à l'édition d'un bon de travaux (BT) qu'il envoie à l'opérateur de maintenance en chef (chef d'atelier par exemple) via son PDA. L'opérateur en chef dirige les opérations de maintenance et à la fin, il renseigne tous les travaux réalisés sur un rapport d'intervention (RI) qu'il envoie par son PDA vers le serveur de la GMAO.

Le diagramme de séquences simplifié correspondant à ce scénario est décrit sur la figure 4.22.

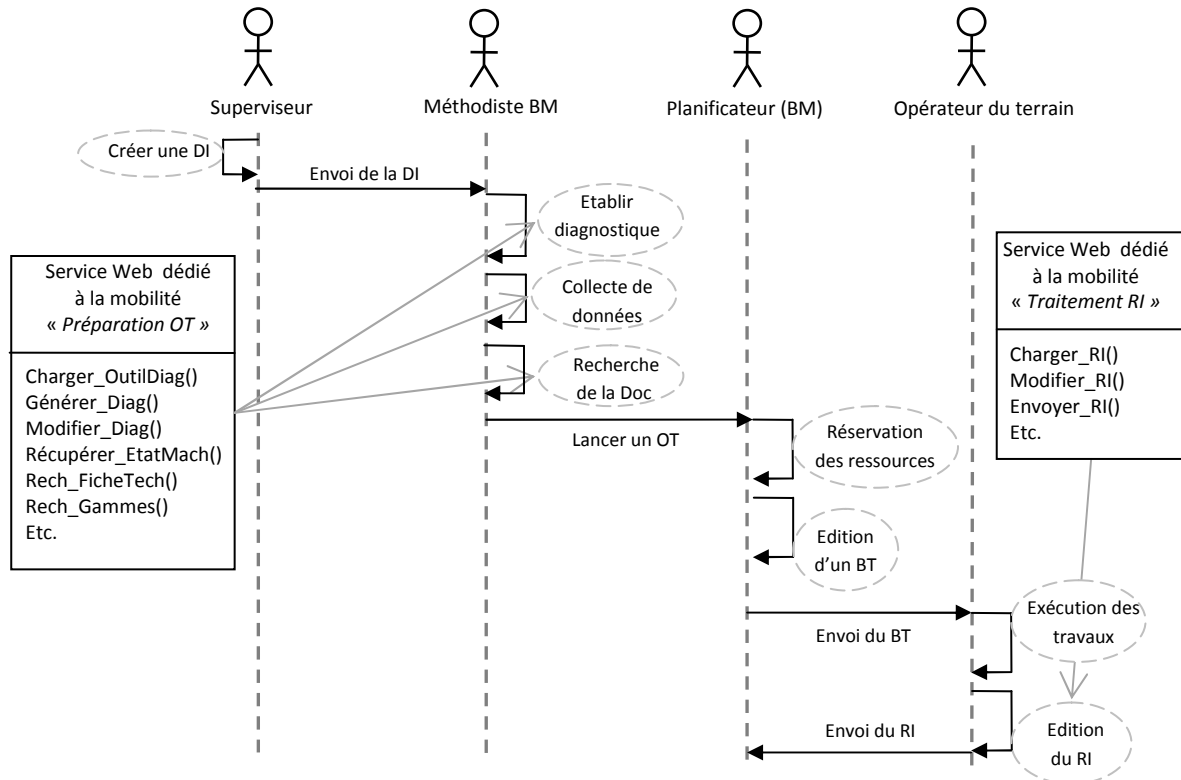


Figure 4.22 : Expression en terme de services Web du scénario de mobilité : « gestion des interventions ».

À l'issue de cette étape, nous obtenons de nouveaux services métier exprimant les besoins en termes de mobilité des acteurs humains. Le tableau 4.3 donne un aperçu de la description des services Web métier dédiés à la mobilité.

Classe de scénarios	Services Web dédié à la mobilité	Profils acteurs nomades	Tâches de maintenance visées	Profils des dispositifs de communication
Traitement de la DI	Préparation de l'OT	Méthodiste du BM	Diagnostiquer la panne, collecter les données sur l'état de la machine et préparation de la documentation technique.	PDA (<i>Personal Digital Assitant</i>) à travers le réseau de proximité WiFi
Gestion de l'OT	Traitement du RI	Opérateur de maintenance	Exécuter les tâches mentionnées sur le BT, renseigner et envoyer au BM un rapport d'intervention (RI)	PDA à travers le réseau de proximité WiFi

Tableau 4.3 : Extrait de la description des services métier dédiés à la mobilité des acteurs humains

Expression des besoins en termes des tâches coopératives entre les acteurs métier

Dans cette étape, on identifie d'abord les scénarios potentiels relatifs à l'accomplissement des tâches de maintenance d'une façon coopérative. Par la suite, nous appliquons le même principe que celui des étapes précédentes afin d'identifier les services métier dédiés aux tâches coopératives. Ces services sont en effet composés de plusieurs autres services d'une granularité inférieure, et peuvent être décrits sous forme de processus métier, c'est-à-dire une orchestration de services afin d'accomplir un objectif métier commun pour les utilisateurs.

Dans ce qui suit, nous considérons un scénario de maintenance relatif à « la réservation des ressources matérielles » nécessaires pour mener l'intervention. En effet, la réservation des ressources peut impliquer plusieurs acteurs humains non seulement au sein du centre de maintenance, mais aussi d'autres acteurs appartenant à d'autres structures (services approvisionnement et achat), voir à l'extérieur de l'entreprise notamment avec les fournisseurs de pièces de rechange et fournitures.

La figure 4.23 montre un scénario métier (tâche coopérative) à travers une succession de tâches élémentaires accomplies par différents acteurs afin de réaliser la réservation des ressources matérielles pour l'intervention.

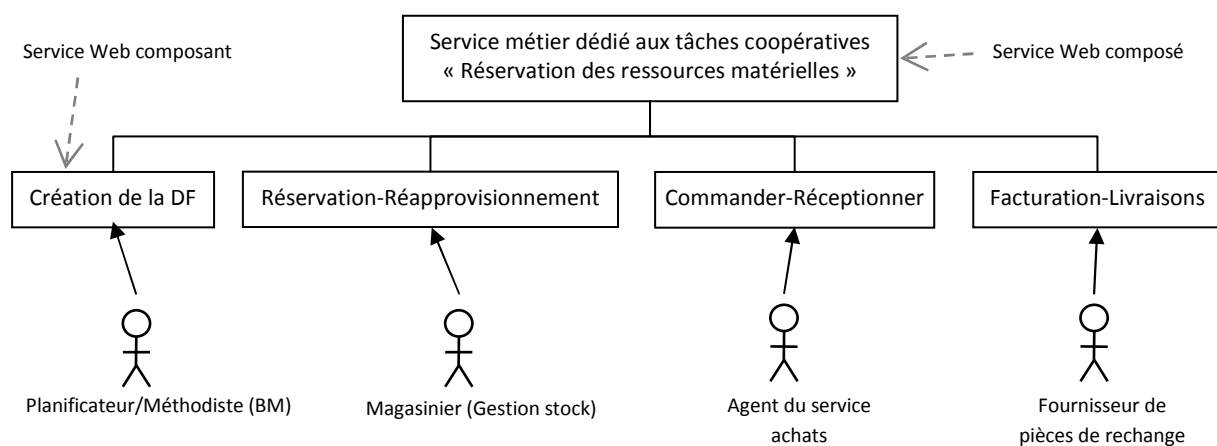


Figure 4.23 : Expression d'une tâche coopérative « réservation des ressources matérielles » à l'aide de services Web composants

Après avoir décomposé les tâches coopératives en un ensemble de tâches élémentaires, nous exprimons ces tâches sous forme de diagramme de séquences l'enchaînement de ces tâches dans le scénario métier « la réservation des ressources matérielles » comme illustré sur la figure 4.24.

Le scénario commence par la création et l'envoi de la demande de fournitures (DF) par l'opérateur du BM à destination du gestionnaire de stocks de pièces. Ce dernier vérifie le stock et constate le manque des fournitures demandé, et lance donc une demande d'achat (DA) auprès du service d'achats. Ce dernier procède à la création et l'envoi d'un bon de commande (BC) au fournisseur sélectionné. Ce dernier, livre la fourniture demandée et crée une facture pour le paiement.

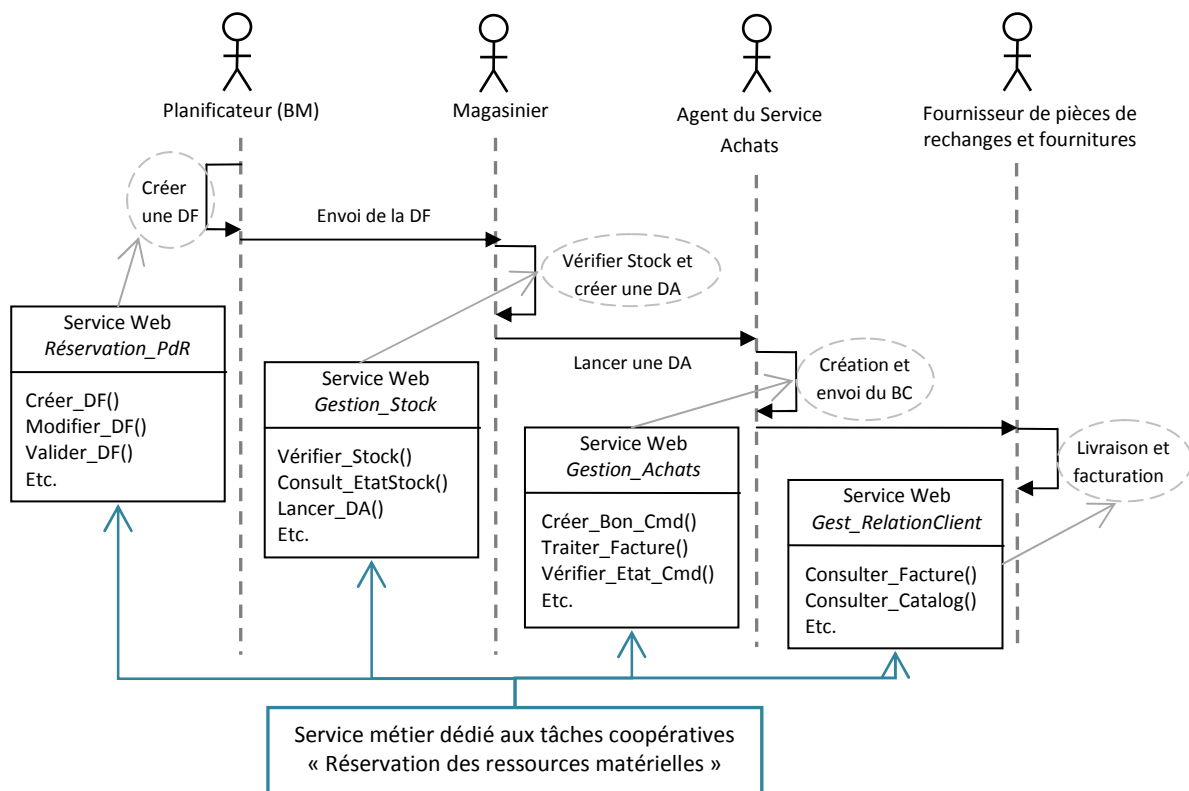


Figure 4.24 : Expression en terme de services Web composants la tâche coopérative relative au scénario métier : « réservation des ressources matérielles ».

III.3. Etape 3 : Analyse du système existant

Lors de notre stage au niveau de l'unité de conditionnement du complexe *Cevital*, nous avons décelé un ensemble d'applications logicielles qui sont impliquées d'une façon directe ou indirecte dans le processus global de la maintenance industrielle. Cependant, ces applications telles que, GMAO (*Coswin*) ou le progiciel ERP (*SAP*), fonctionnent d'une manière indépendante. En effet, il n'existe aucune coopération entre ces systèmes. Par ailleurs, la communication entre les différentes structures de l'entreprise s'effectue d'une façon non-informatisée (sous forme de documents papier ou verbalement) ce qui peut engendrer des retards et des incohérences au niveau des traitements d'informations et par conséquent, dans la réalisation des tâches relatives à la maintenance telles que la réservation des ressources (matérielles et humaines) et la collecte de données nécessaires (état des machines, fiche technique, etc.) pour l'intervention.

De ce constat, nous proposons d'analyser chacun des systèmes existants (GMAO, SCADA, ERP, etc.) en identifiant les fonctionnalités potentielles susceptibles d'être transformées en services Web métier. Cette activité nécessite cependant une réflexion profonde notamment sur le mode de fonctionnement de ces applications (transactionnelles, en temps réel, etc.), leurs technologies d'implémentation (Java, C++, etc.) et aussi leurs typologies de déploiement (client/serveur, standalone, etc.).

Dans cette étape, nous considérons à titre représentatif deux types d'applications pouvant « fournir » des composants logiciels capables de réaliser des services métier à exposer. Ces systèmes sont : (1) ERP (gestion stocks et gestion des ressources humaines), et (2) SCADA. La figure 4.25 illustre l'analyse de l'existant en appliquant l'approche ascendante (*bottom-up*)

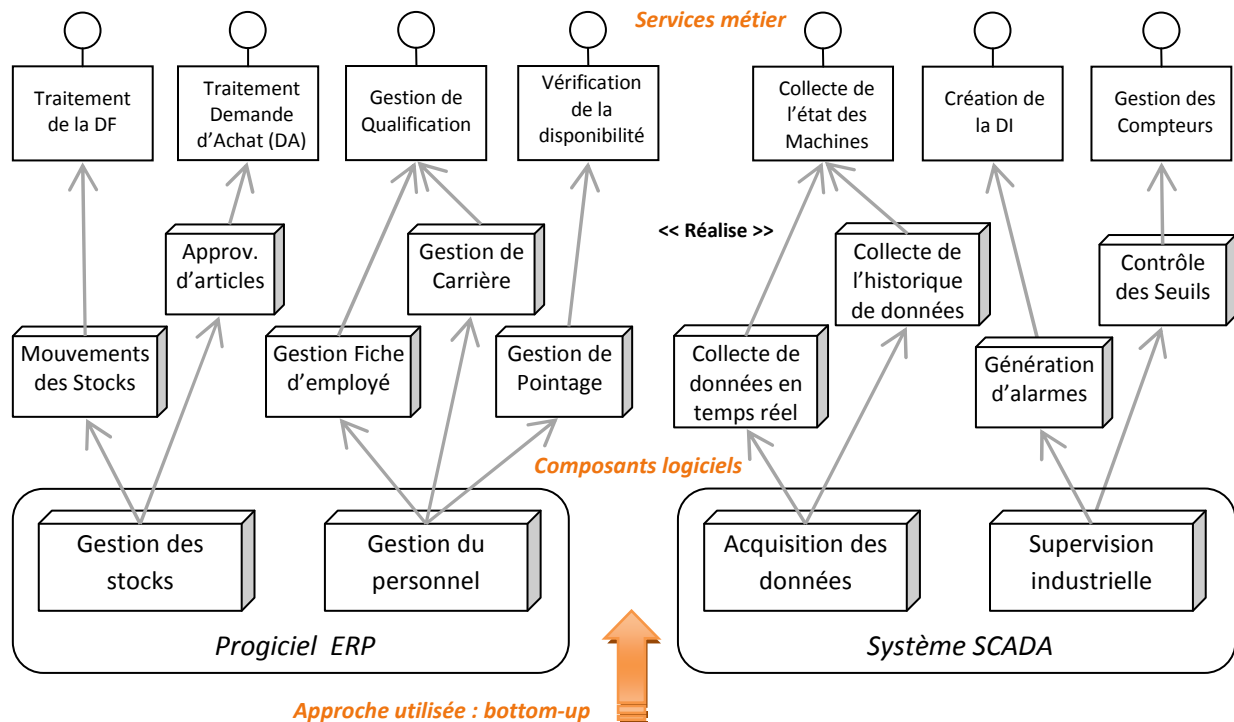


Figure 4.25 : Décomposition des deux systèmes : ERP et SCADA.

A travers cette décomposition du système existant, nous pouvons déduire non seulement les composants logiciels capables de fournir une réalisation pour les services métier déjà identifiés, mais aussi de découvrir de nouvelles fonctionnalités pouvant être dissimilées dans les applications existantes. A titre d'exemple, le système SCADA peut nous fournir des fonctionnalités permettant de gérer les compteurs d'une machine à travers le module « contrôle des seuils » (figure 4.25). En effet, la gestion des compteurs constitue un élément indispensable dans les processus de maintenance préventive (prévisionnelle et conditionnelle).

III.4. Conclusion sur la phase 2

Au cours de cette phase, nous avons procédé à un deuxième niveau d'analyse afin d'affiner l'identification des services métier candidats. Pour cela, nous avons appliqué trois étapes d'analyse.

La première étape consiste à la catégorisation des services, qui vise à regrouper les services métier autour d'une même fonctionnalité métier liée à la maintenance. Cela nous permet d'identifier de nouveaux services notamment les services composés qui sont dotés d'une granularité importante et aussi les services CRUD permettant la gestion des objets métier.

Dans l'étape 2, nous avons exprimé les besoins identifiés en termes de services Web d'une part et d'autre part vis-à-vis de la mobilité et des tâches coopératives des acteurs humains.

Enfin, l'étape 3 vise à identifier les composants logiciels dissimilé dans les systèmes existants (ERP, SCADA, etc.) afin de fournir une réalisation pour les services déjà identifiés ou encore découvrir de nouvelles fonctionnalités pouvant être transformer en services métier. Il s'agit en effet de décomposer le système existant sous forme de *modules logiciels* afin de réutiliser ces différents composants dans l'architecture SOA.

A l'issue de cette phase, nous aurons une vue plus élaborée sur les différents types de services notamment les services métier, fonctionnels (par exemple issus des systèmes existants) et techniques (tels que les services CRUD). Tous ces services seront par la suite spécifier en détails lors de la phase 3, objet de la section ci-après.

IV. Phase 3 : Spécification de l'architecture orientée services

Dans la phase 3 de la démarche, nous procédons à la spécification de tous les constituants de l'architecture orientée services (SOA) dédiée aux systèmes de maintenance industrielle.

Pour cela, nous conduisons cette phase à travers trois grandes étapes : (1) spécification des processus métier ; (2) spécification de l'interface utilisateur ; (3) spécification des services et composants. Chacune de ces étapes est décomposée en sous-étapes permettant d'affiner progressivement la spécification de l'architecture orientée services du système de maintenance industrielle.

IV.1 Etape 1 : Spécification des processus métier

L'objectif visé par cette étape est de définir d'une manière succincte la structure interne des processus métier à travers les tâches et leurs enchainements dans le processus et les interactions entre les partenaires externes (humains ou automatiques) avec le processus métier de la maintenance industrielle. Pour cela, nous procédons suivant deux sous étapes : (1) spécification des interactions Homme-Machine et (2) spécification des processus métier à l'aide de BPMN.

IV.1.1. Spécification des interactions Homme-Machine

Afin de montrer la procédure à suivre dans la spécification des interactions Homme-Machine, nous avons choisi le processus métier « *traitement de la DI* » identifié lors de la première phase (§II.1). Ce processus décrit un enchainement d'invocations de plusieurs services métier et fait intervenir des acteurs métier de différents profils.

Nous commençons par : (1) définir les participants à travers les « *liens partenaires* » de type humain et automatique ; Ensuite, (2) nous spécifions leurs interactions avec le processus métier à travers les « *variables d'échanges* ».

1) Définition des participants

La figure 4.26 exprime globalement les interactions entre les différents partenaires externes avec le processus métier « traitement de la DI ».

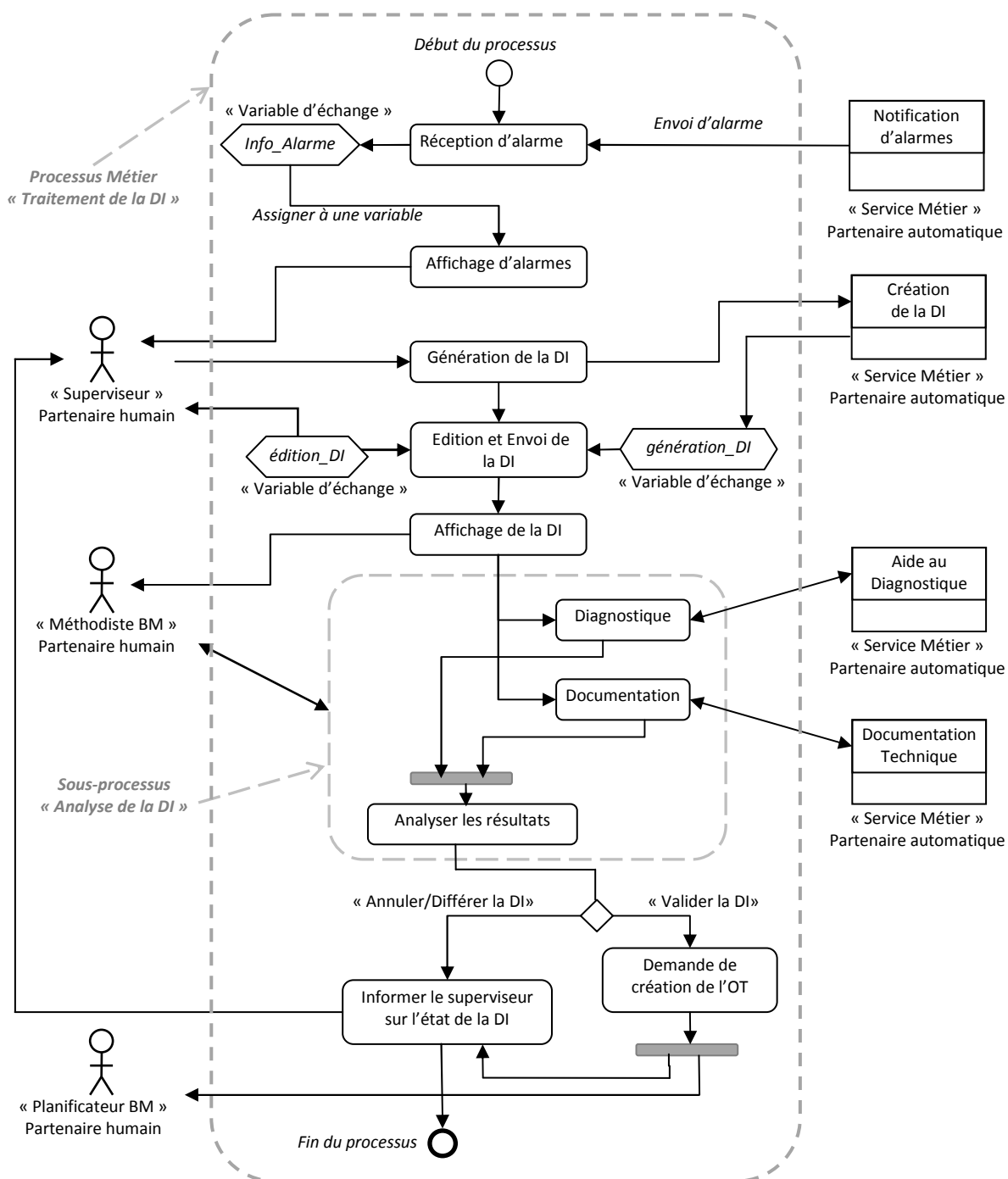


Figure 4.26 : Expression des interactions entre les partenaires humains et automatiques avec le processus métier « traitement de la DI ».

Les participants humains mis en évidence dans l'enchaînement indiqué sur la figure 4.26 sont : (1) le superviseur, (2) le méthodiste du BM et (3) le planificateur du BM.

Les services métier participants mis en évidence dans l'enchaînement (figure 4.26) sont : (1) *notification d'alarmes*, (2) *création de la DI*, (3) *aide au diagnostique* et (4) *documentation technique*.

Nous pouvons exprimer tout ces liens partenaires dans le processus métier à l'aide de la spécification montrée sur la figure 4.27 :

```
<Liens_Partenaies>
<Acteurs_humains>
  <Acteur_humain nom='Superviseur' rôle = 'Etablir et envoyer la DI au BM' />
  <Acteur_humain nom='Méthodiste BM' rôle = 'Analyse de la DI et création de l'OT' />
  <Acteur_humain nom='Planificateur BM' rôle = 'Préparation de l'OT' />
</Acteurs_humains>
<Services_Métier>
  <Service nom='Notification_Alarmes' rôle = 'Déetecte et envoie les alarmes' />
  <Service nom='Création_DI' rôle = 'Déetecte et envoie les alarmes' />
  <Service nom='Aide_au_Diagnostique' rôle = 'Généérer le diagnostique' />
  <Service nom='Documentation_Technique' rôle = 'Collecte de la Documentation Technique sur la machine' />
</Services_Métier>
</ Liens_Partenaies >
```

Figure 4.27 : Extrait du fichier relatif à la spécification des liens partenaires dans le processus métier « *traitement de la DI* ».

2) Spécification des interactions à travers les variables d'échanges

Dans ce qui suit, nous nous intéressons à la spécification des interactions à travers les variables d'échanges et les messages échangés (messages d'entrée, messages de sortie et messages d'exception).

Dans notre cas, nous considérons à titre représentatif les interactions entre d'une part le *superviseur* et le processus et d'autre part entre le processus et le service métier *création de la DI*.

La figure 4.28 exprime les éléments suivants : (1) le participant humain correspondant au superviseur invoque l'opération *générer_DI* défini par le processus métier « *traitement de la DI* ».

L'interaction se fait à travers la variable *génération_DI* en utilisant le message *informations_alarme* en entrée, le message *nouvelle_DI* en sortie et le message *erreur_création* en cas d'erreur; (2) le processus métier invoque le service métier participant « *création_DI* » afin de créer la DI à partir des informations issues de l'alarme reçue (code de la panne, date d'alarme, identifiant de la machine, etc.); (3) à la réception de la DI, le superviseur peut éditer en complétant la DI avec des observations et les informations personnelles (nom d'opérateur, contact, etc.); (4) à la fin de l'édition, le processus procède à l'envoi de la DI vers le système GMAO, précisément via le service d'*affichage_des_DI*.

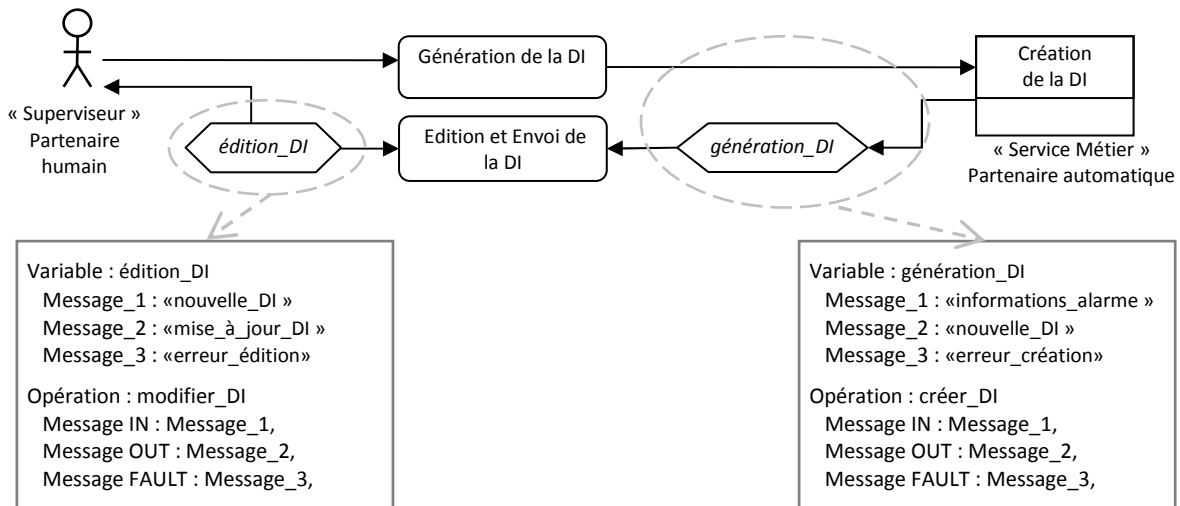


Figure 4.28 : Exemple de spécification des interactions lors de la génération de la DI à travers des variables et messages d'échange.

A l'issue de cette étape, nous avons spécifié, à travers un exemple illustratif relatif à la gestion des DI, les interactions entre les différents partenaires (humains ou automatiques) et le processus métier « *traitement de la DI* ». Par conséquent, nous appliquons la même démarche pour le reste des processus métier en respectant ainsi le cycle de développement itératif et incrémental. Dans l'étape suivante, nous nous intéressons à la définition de la structure interne des processus métier à travers la notation BPMN.

IV.1.2. Spécification des processus métier à l'aide de BPMN

Après avoir exprimé les interactions Homme-Machine dans les processus métier sous forme de spécifications générale, nous allons dans cette étape traduire ces spécifications sous forme de modèles de processus métier en utilisant le formalisme BPMN (*Business Process Modeling Notation*. Chapitre 3, §III.3).

Rappelons que dans SOA, l'utilisation de la notation BPMN permet d'une part de modéliser les processus métier indépendamment des technologies sous-jacentes d'implémentation des services et d'autre part, permet à des utilisateurs non informaticiens de participer à la modélisation des processus métier selon leurs besoins.

Dans cette partie, nous présentons un modèle BPMN correspondant au processus précédent « *traitement de la DI* » (figure 4.29). Ce modèle met en avant un ensemble d'éléments tels que : les activités internes au processus (invocation d'un service partenaire, renvoi des résultats, etc.), tâches utilisateurs (par exemple: analyse de la DI), les branchements (exclusifs, parallèles, etc.) et les sous-processus (ou tâches composées) qui permettent de définir des abstractions et de choisir la granularité de l'information.

La figure 4.29 illustre le modèle BPMN relatif au processus métier « *traitement de la DI* ». Ce modèle a été réalisé à l'aide de l'outil de modélisation *BizAgi Process Modeling* (www.bizagi.com).

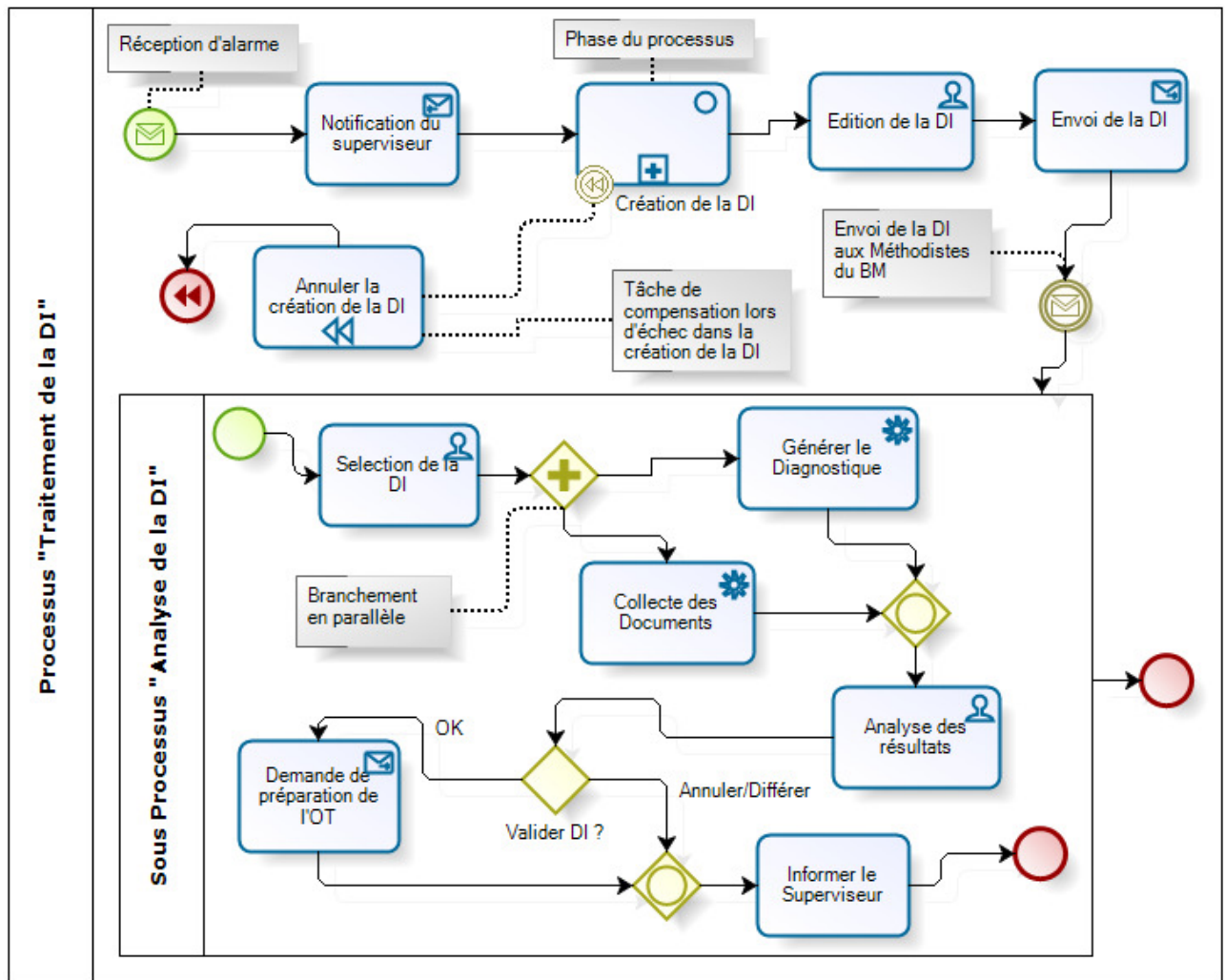


Figure 4.29 : Modélisation du processus métier « traitement de la DI » par la notation BPMN.

Le processus métier *traitement de la DI* est déclenché suite à la réception d'une alarme de la part du système SCADA.

- Le superviseur prend connaissance de la notification et décide de créer automatiquement la DI suivant les informations véhiculées par le message d'alarme. La phase de « création de la DI » motionnée sur le modèle BPMN est une *tâche abstraite*, caractérisée par le symbole « + » qui signifie une abstraction de plusieurs tâches effectuées lors de la création de la DI (récupération des informations sur la panne, sauvegarde de la DI, etc.).
- Cependant, le superviseur peut annuler cette tâche en choisissant de créer la DI manuellement. Cette opération est représentée par une « tâche de compensation » sur le modèle BPMN (figure 4.24) en la spécifiant par le symbole contenant deux flèches.
- Suite à la création, l'édition et l'envoi de la DI, le processus actualise la liste des DI en notifiant la nouvelle DI auprès du *methodiste BM*. Ce dernier procède du sous processus « analyse de la DI » en déclenchant l'outil d'aide au diagnostique et aussi à la collecte des informations sur l'équipement concerné par la DI. Le sous processus est montré sur le

modèle BPMN sous forme d'un rectangle englobant les tâches relative à l'analyse de la DI. Les tâches de diagnostic et de récupération des documents techniques peuvent être effectuées en parallèle, ce qui est indiqué par l'objet de *branchement parallèle* sur le modèle BPMN (figure 4.29). À la réception du diagnostic et de la documentation nécessaire, le méthodiste procède à l'analyse des résultats obtenus. Cette tâche est de type utilisateur.

- À la suite de cette analyse, le méthodiste prend une décision sur la validation ou d'annulation de la DI. Dans les deux cas, le superviseur émetteur de la DI, est informé sur l'état de traitement de la DI (tâche *informer superviseur sur l'état de la DI*, figure 4.29). Dans le cas de validation de la DI, le méthodiste lance une demande de préparation d'un OT (ordre de travail) auprès du *planificateur BM* afin de préparer l'intervention sur le site d'incident.

Translation du modèle BPMN vers un programme BPEL

Nous présentons dans ce qui suit, le programme d'exécution BPEL relatif au processus métier « traitement de la DI ». Rappelons que la notation BPMN définit sous forme graphique les tâches métier et leur relations. Le modèle BPMN est en effet exprimé via les outils de modélisation sous le format XML, ce qui facilite sa conversion vers un programme BPEL.

Par ailleurs, le BPEL définit l'enchaînement et la logique des actions qui seront exécutées par le moteur BPM. Il constitue donc le code source de l'application relative au processus métier.

Le tableau suivant illustre quelques objets BPMN et les éléments du langage BPEL correspondant.




Objet BPMN	Notation	Élément BPEL	Description
<p>Objet : Start Event (EventType: Message)</p> <p>Attributs : Name = "Receive1" Instantiate = "True" Participant = "Notification_Alarme" Operation = "notfier_Alarme" Interface = "tns: Notif_Alarme_PortType"</p>		<pre><receive name="Receive1" createInstance="yes" partnerLink="Notification_Alarme " operation="notfier_Alarme" portType="tns: Notif_Alarme_PortType"/></pre>	<p>Evènements Début : permettent de signaler l'arrivée ou l'attente d'un message.</p>
<p>Objet : Task (TaskType: Service)</p> <p>Attributs : Name = "Invoke_Diag" InMessage = " Demande_Diag " OutMessage = " Diagnostique " Participant = " Diagnostique " Operation = " Générer_Diag " Interface = "tns: Diagnostique _PortType"</p>		<pre><invoke name="Invoke_Diag" inputVariable="Demande_Diag" outputVariable="Diagnostique" partnerLink="Diagnostique" operation="Générer_Diag" portType="tns: Diagnostique_PortType"/></pre>	<p>Une tâche : est un élément indivisible. Elle représente une action. Chaque tâche a un début et une fin et donc une tâche ne peut débuter que si la tâche précédente est terminée.</p>
<p>Objet : End Event_(EventType: Message)</p> <p>Attributs : Name = "Invoke_Diag" Message = " Etat_DI " Participant = " Ap_Supervision " Operation = " DemandeIntervOperation " Interface = "tns: Diagnostique _PortType"</p>		<pre><reply name="Reply1" variable="Etat_DI" partnerLink="Ap_Supervision" operation="DemandeIntervOperation" portType="tns: superv_PortType "/></pre>	<p>Evènements Fin : permettent de signaler la fin d'un enchaînement de tâche par l'envoi d'un message de réponse.</p>

Tableau 4.4 : Translation de quelques objets BPMN vers les éléments du langage BPEL

Afin de traduire le modèle BPMN vers un programme exécutable BPEL, plusieurs outils de modélisation des processus métier existent tels que *Intalio BPMN Process Designer* (<http://bpms.intalio.com/>), *Eclipse STP Modeler* (<http://download.eclipse.org/stp/downloads/>), etc.

La figure 4.30 montre un extrait du programme BPEL généré à l'aide de l'outil *Intalio BPMN Process Designer*.

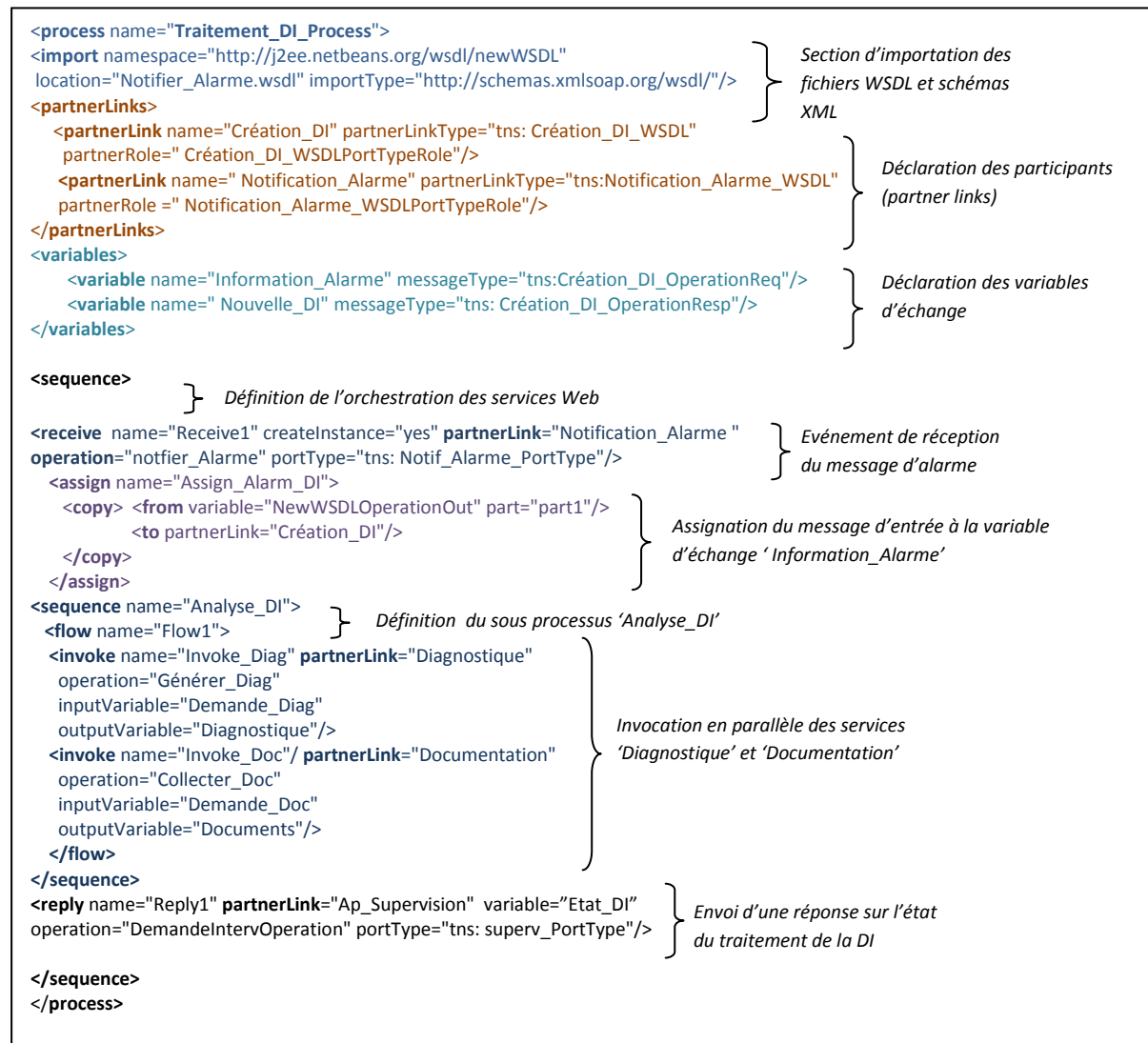


Figure 4.30 : Extrait du code BPEL relatif au processus métier « traitement de la DI ».

IV.2. Etape 2 : Spécification de l'interface utilisateur

L'objectif visé par cette étape est de spécifier et d'élaborer les différents constituants de l'interface utilisateur d'un système de maintenance. La spécification concerne d'une part, les services Web orientés présentation et d'autre part la définition des interfaces permettant d'accéder à ces services de présentation. Par ailleurs, nous présentons l'application de la démarche relative à la conception des maquettes et les prototypes dédiés aux systèmes de la

maintenance industrielle. Pour cela, nous procédons suivant deux sous étapes : (1) spécification des services présentation et (2) maquettage et prototypage de l'IHM.

IV.2.1. Spécification des services de présentation

Dans cette partie, nous focalisons notre étude sur la conception des services Web orientés présentation capables de générer dynamiquement les aspects liés à l'interface graphique d'un système de maintenance. Cela peut être réalisé sans avoir recours à développer du code spécifique pour la génération de l'interface dédiée à la maintenance.

En revanche, la conception des services orientés présentation nécessite de définir succinctement les besoins des acteurs métier en termes d'IHM. Par conséquent, cette étape doit être réalisée en parallèle avec l'étape de maquettage/prototypage (§IV.2.2) tout en respectant le principe d'une démarche itérative et incrémentale.

Rappelons que la mise en œuvre basée sur les services Web de la couche présentation du modèle SOA nécessite l'adoption de plusieurs normes de spécifications. Ces spécifications concernent d'une part, la définition des interfaces dites communes (dans notre projet, la norme WSRP a été retenue) pour établir des communications transparentes entre le portail Web et les services Web de présentation résidant sur les serveurs d'applications et d'autre part, une spécification définissant la façon de réaliser les *portlets* (par exemple la spécification JSR) qui fournissent à la fois une logique applicatives (ou de données) et celle de la présentation.

Afin d'appliquer les aspects cités précédemment, nous limitons notre champ d'étude sur une partie constituant le portail Web dédié à la maintenance industrielle. Précisément, nous considérons le domaine fonctionnel « analyse des résultats » (décrit sur la figure 4.5). Ce dernier regroupe l'ensemble des services métier fournissant les informations pertinentes pour assister les experts et les méthodistes du BM dans l'évaluation des méthodes de maintenance adoptées, analyse des coûts engendrés par les interventions, etc.

Par conséquent, nous focalisons notre étude sur un ensemble représentatif de services Web amenés à générer des contenus pour les différentes vues du « tableau de bord ». Ce dernier représente un outil indispensable pour les opérateurs de maintenance en leur fournissant des vues de diverses natures (tableaux, graphes, synoptiques sur les chaînes de production, etc.) permettant ainsi d'avoir une vision globale sur le déroulement des activités de la maintenance.

La figure 4.31 suivante illustre une architecture simplifiée de deux services de présentation (*Stat_taux_arrets* et *Stat_état_OT*) basés sur la norme WSRP.

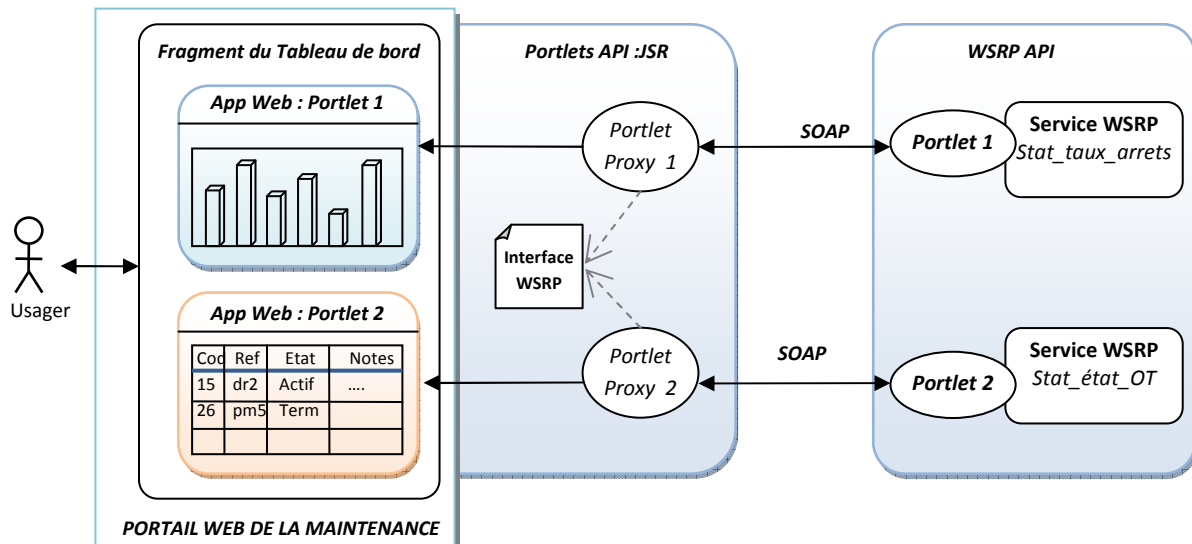


Figure 4.31 : Architecture simplifiée de la couche présentation basée sur les services WSRP

Les services *Stat_taux_arrets* et *Stat_état_OT* permettent respectivement : (1) de visualiser sous forme de graphes les statistiques sur les taux d'arrêts liés aux pannes et (2) d'afficher l'état de déroulement des différents OT (ordre de travail).

Ces services permettent de produire dynamiquement des « fragments d'interface » pouvant être directement intégrés dans le portail Web dédié à la maintenance industrielle. En effet, les services WSRP¹⁶ distants présentés sur le schéma de la figure 4.31, peuvent être invoqués à travers une interface WSRP commune en utilisant des *portlets* génériques d'accès (appelées aussi *portlets proxy*) qui sont totalement indépendantes des technologies utilisées par les services WSRP présents sur le serveur d'applications.

Par ailleurs, le fait d'intégrer dynamiquement ces fragments de présentation (portlets spécifiées par JSR-168¹⁷) au sein du portail Web de la maintenance, cela permet de réaliser une « coopération » entre plusieurs portails Web dédiés à la maintenance appartenant à des organisations différentes. Ainsi, le partage des informations sur la maintenance est devenu possible grâce à l'exposition du contenu (données et leurs présentations) sous forme de services WSRP.

En conclusion, l'utilisation des services orientés présentation (services WSRP) incluant à la fois la logique applicatives et de présentation au sein du système de la maintenance, contribue considérablement dans la mise en œuvre d'une « maintenance collaborative » déjà énoncée dans la partie liée aux motivations de ce mémoire (chapitre 1, § IV.5).

A l'issue de cette étape, nous avons spécifié l'IHM à l'aide des services Web orientés présentation. Par ailleurs, nous avons insisté sur le rôle de ce type de services dans la *médiation* entre les services Web métier résidants sur les serveurs d'application (par exemple

¹⁶ WSRP : Web Services for Remote Portlets. Specification d'OASIS : www.oasis-open.org/committees/wsrp/

¹⁷ JSR - 168 : Java Specification Requests N° 168. Spécification des portlets: jcp.org/aboutJava/communityprocess/final/jsr168/

J2EE) et les différentes applications Web (basées sur JSP, PHP, etc.) déployées sur les serveurs Web et qui sont directement exploitables par les utilisateurs finaux via un navigateur Web.

IV.2.2. Maquettage et prototypage de l'IHM

Dans cette partie, nous intéressons à la dernière couche du modèle SOA, notamment celle de l'interface graphique de l'utilisateur (ou *GUI: Graphical User Interface*). Nous avons présenté dans le chapitre 3 la démarche globale pour la conception des IHM. Nous avons souligné que cette démarche est basée essentiellement sur des scénarios métier ciblés. Ces derniers sont traités afin de dégager un ensemble de *maquettes* représentant des vues assez complètes sur les différents aspects de l'IHM caractérisant les scénarios relatifs à la maintenance industrielle.

Ces maquettes vont par la suite évoluer vers des *prototypes d'interface fonctionnels* qui peuvent être modifiés facilement selon les besoins des utilisateurs finaux du système.

Illustration d'un scénario de la gestion de la maintenance sous l'angle d'IHM

Nous présentons dans ce qui suit, un scénario représentatif sur l'utilisation du futur système de maintenance industrielle. Précisément, nous considérons le cas d'utilisation relatif à « la réservation des ressources matérielles et humaines » impliqué dans le processus métier de *préparation de l'OT*.

La figure 4.32 suivante montre les différentes vues des services offerts par le système de maintenance afin de gérer la réservation des ressources nécessaires à l'OT de bout en bout.

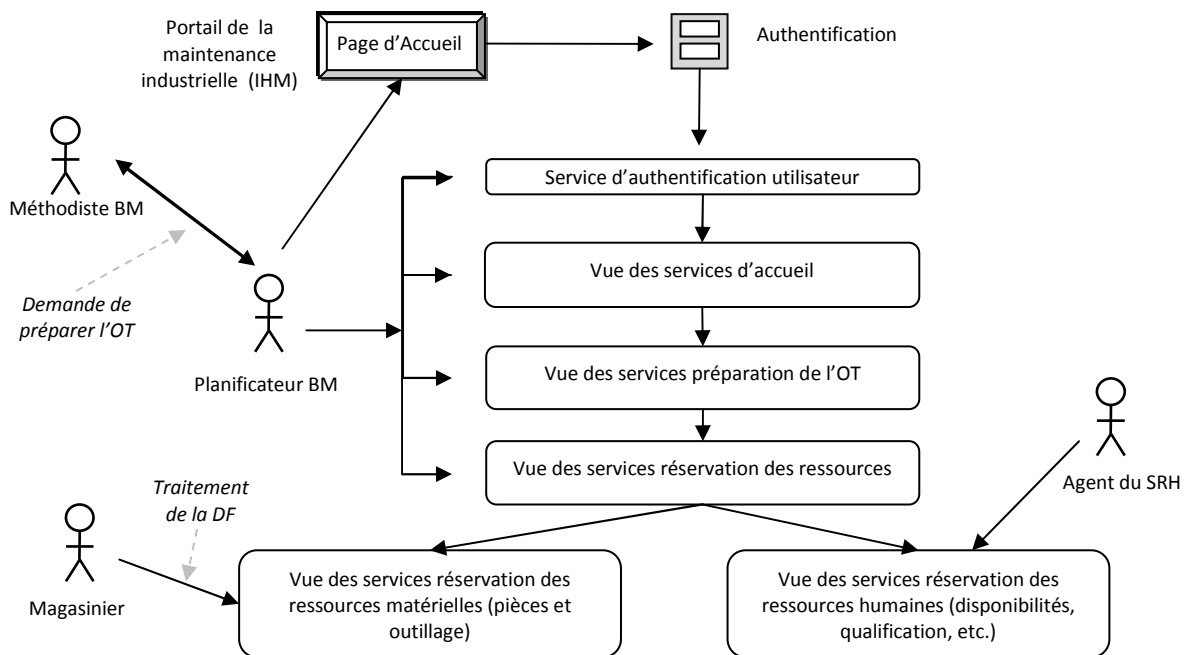


Figure 4.32 : Illustration du scénario *réservation des ressources* sous l'angle d'IHM

Dans ce qui suit, nous allons illustrer d'une façon succincte le scénario associé à la spécification présentée sur la figure 4.32 à l'aide de quelques maquettes représentatives qui pouvant évoluer ensuite vers des prototypes plus élaborés. La séquence d'enchaînement relatif à ce scénario est la suivante :

- 1) Suite à la demande de préparation de l'OT du methodiste BM, le planificateur BM décide d'accéder au système de maintenance de l'entreprise à travers son portail. Ce dernier intègre tous les services nécessaires à la gestion de la maintenance.
- 2) Le système demande l'identification de l'opérateur à travers le service d'authentification.
- 3) L'opérateur (planificateur BM) s'identifie.
- 4) Le système invoque le service d'identification utilisateur afin de vérifier le droit d'accès (zone 1, figure 4.33).
- 5) En fonction du profil de l'utilisateur, le système affiche une vue personnalisée de l'ensemble des services dédiés à la maintenance dont les services relatifs à « la réservation des ressources » (zone 3, figure 4.33) et une liste de tâches autorisées pour l'opérateur (zone 5).

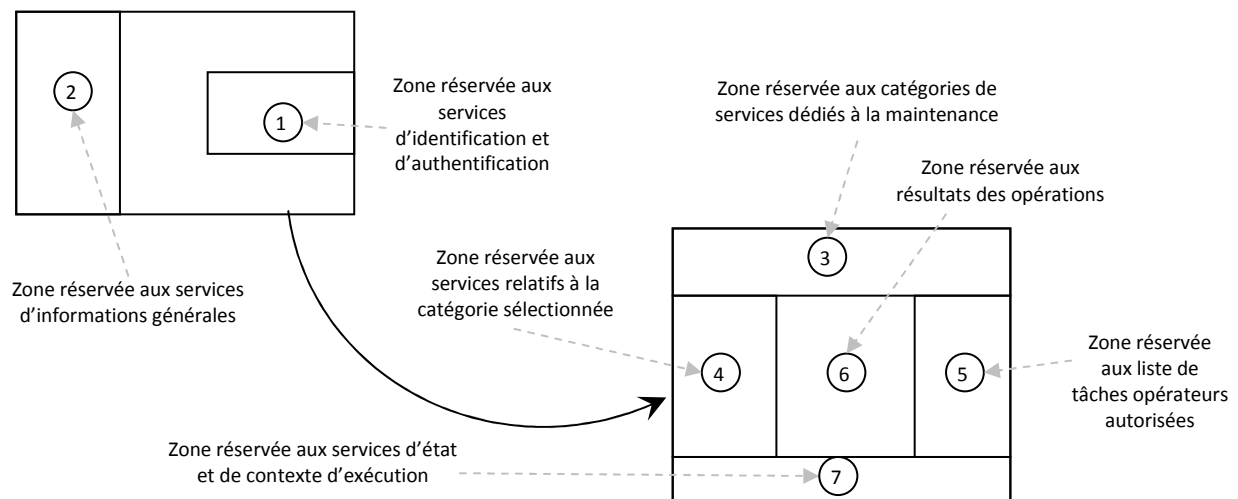


Figure 4.33 : Maquette d'authentification et accès de l'opérateur aux différents services dédiés à la maintenance

Par la suite,

- 6) Le planificateur choisit la rubrique relative aux services de réservation des ressources en sélectionnant le volet « ressources ».
- 7) Le système affiche alors les différentes tâches associées à deux catégories : (1) réservation du matériel et (2) réservation du personnel.
- 8) L'opérateur choisit la catégorie relative à la réservation du matériel et invoque le service création de la demande de fourniture (DF) (zone 8, figure 4.34).
- 9) Le système affiche alors un formulaire relatif à la création d'une DF.
- 10) Après avoir saisi la référence de l'OT en cours et les différentes informations relatives aux pièces et fournitures (code, quantité, etc.) requises pour l'intervention, le planificateur valide le formulaire (zone 8, figure 4.34).

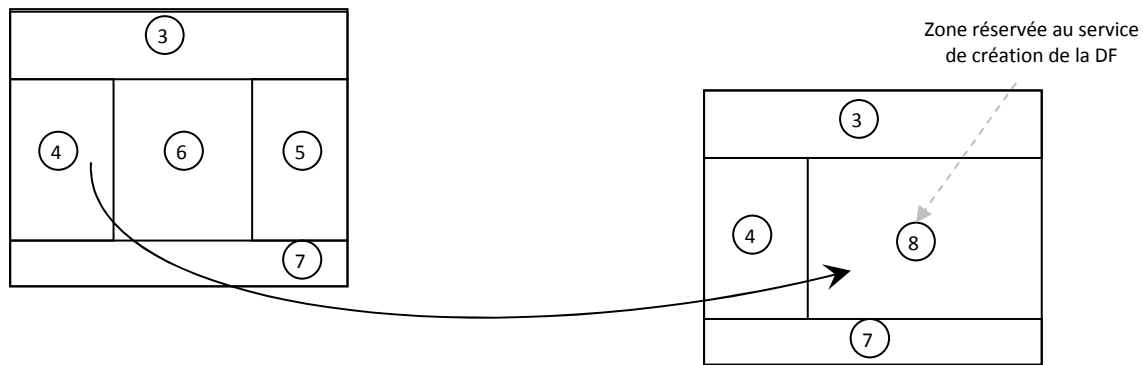


Figure 4.34 : Maquette relative à la réservation des ressources matérielles

Ensuite,

11) L'opérateur procède à la réservation du personnel qualifié pour effectuer les travaux de maintenance en invoquant le service réservation des ressources humaines (zone 4, figure 4.35).

12) Le système affiche alors la liste du personnel de la maintenance en précisant les informations sur chaque opérateur telles que : la disponibilité, l'affectation et la qualification (zone 9, figure 4.35).

13) Le planificateur sélectionne sur la liste affichée, les opérateurs habilités à exécuter les opérations prescrites sur l'OT.

14) A la fin de l'opération de réservation, le système rajoute les ressources sélectionnées à l'OT déjà créé et affiche une vue en mode d'édition sur l'OT pour une éventuelle modification ou impression par le planificateur (zone 10, figure 4.35).

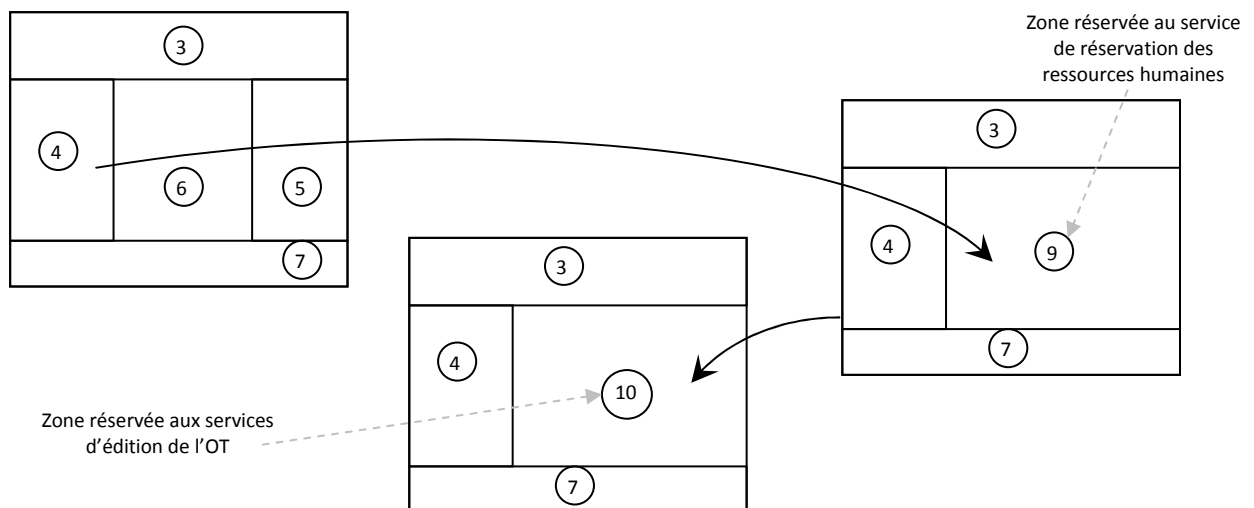


Figure 4.35 : Maquettes relatives à la réservation des ressources matérielles et l'édition de l'OT

Après avoir élaboré les différentes maquettes relatives aux scénarios métier de la maintenance, l'étape suivante à faire évoluer ces maquettes vers des prototypes d'interface pouvant être directement validés par les utilisateurs finaux.

La figure 4.36 illustre à titre représentatif, un ensemble de prototypes liés au portail de la maintenance et qui représentent les vues principales fournies par le système lors du processus de réservation des ressources.

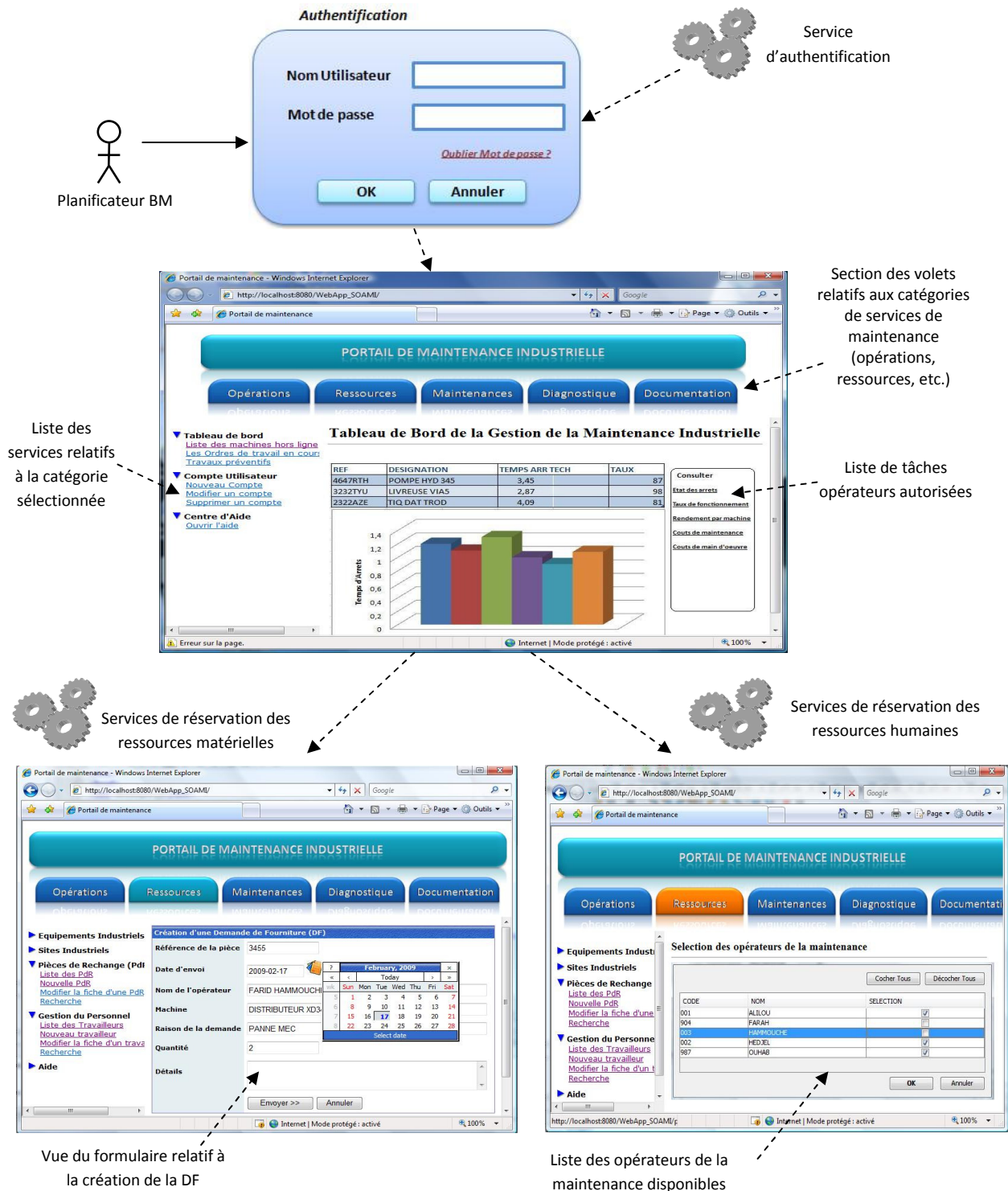


Figure 4.36 : Exemples de prototypes d'interface relatifs au scénario de réservation de ressources

IV.3 Etape 3 : Spécification des services et composants

Les phases précédentes nous ont permis de recenser plusieurs éléments conceptuels, notamment les domaines fonctionnels, les processus métier (orchestration de services) et les différents types de services (métier, CRUD et présentation). Tous ces éléments devront être assemblés et structurés afin de construire les différentes couches de l'architecture SOA. Cette construction est basée essentiellement sur les différents modèles conceptuels permettant de spécifier la structure interne des composants de l'architecture SOA et les relations de dépendance entre les différents constituants de l'architecture (composants, services, classes, etc.).

Ainsi, l'objectif de cette dernière étape est de représenter l'architecture SOA dédiée aux systèmes de maintenance sous forme de modèles conceptuels de divers niveaux d'abstraction et reflétant la vue globale du système de maintenance à développer lors de la phase d'implémentation.

Nous abordons cette étape suivant trois types d'activités : (1) analyse des sous-systèmes, (2) spécification des services et (3) spécification des composants.

IV.3.1. Analyse des sous-systèmes

Nous rappelons que cette étape consiste à élaborer les modèles de composants relatifs à chaque sous-système (domaine fonctionnel) déjà identifié lors de la phase 1. Ces modèles représentent une vue d'abstraction de haut niveau pour l'ensemble des composants formant l'architecture SOA.

Nous considérons à titre représentatif le domaine fonctionnel *ordre de travail*. Ce dernier est analysé en définissant les différents types de composants et leurs flux de dépendance au sein du sous-système *ordre de travail*.

Par ailleurs, il est important de souligner que nous respectons, au cours de cette étape, le processus de développement *itératif et incrémental*. En effet, on considère chaque sous-système pour la modélisation des différents composants, et on procède de la même manière pour les autres sous-systèmes.

La figure 4.37 illustre un modèle de composants simplifié relatif au sous-système *ordre de travail*.

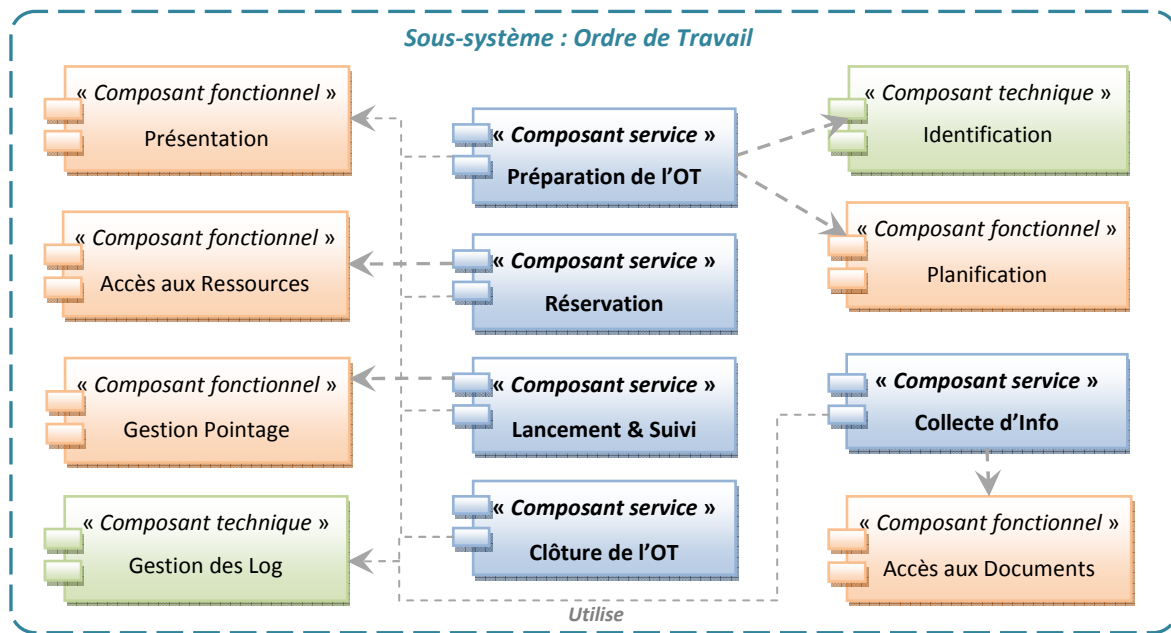


Figure 4.37 : Modèle de composants relatif au sous-système *Ordre de travail*.

Le modèle présenté sur la figure 4.37 définit trois types de composants : (1) des composants services réalisant les services métier tels que *préparation de l'OT*, *réservation*, etc. (2) des composants fonctionnels qui sont utilisés pour des besoins fonctionnels par les composants services afin de réaliser certains traitements tels que *accès aux ressources*, *planification*, etc. (3) des composants techniques qui traitent les besoins non-fonctionnels tels que la sécurité à travers l'*identification* des utilisateurs et la *gestion du fichier log* pour le suivi du déroulement des différentes phases liées au processus métier « ordre de travail ».

IV.3.2. Spécification des services

Dans ce qui suit, nous présentons l'application des étapes définies par l'approche proposée dans la conception des services. Pour cela, nous reprenons le domaine fonctionnel « ordre de travail » en tenant compte d'un ensemble représentatif de services déjà identifiés. Ensuite, nous spécifions les interfaces de chaque service en définissant les types de données échangées (TDE), les opérations et leurs assemblage dans le fichier WSDL et enfin, spécifier les contraintes non-fonctionnelles telles que les performances, la sécurité, etc.

En revanche, avant de spécifier tous ces éléments, nous commençons d'abord à modéliser les objets métier manipulés par les services SOA en élaborant le *modèle de classes d'objets*.

Modélisation des objets métier

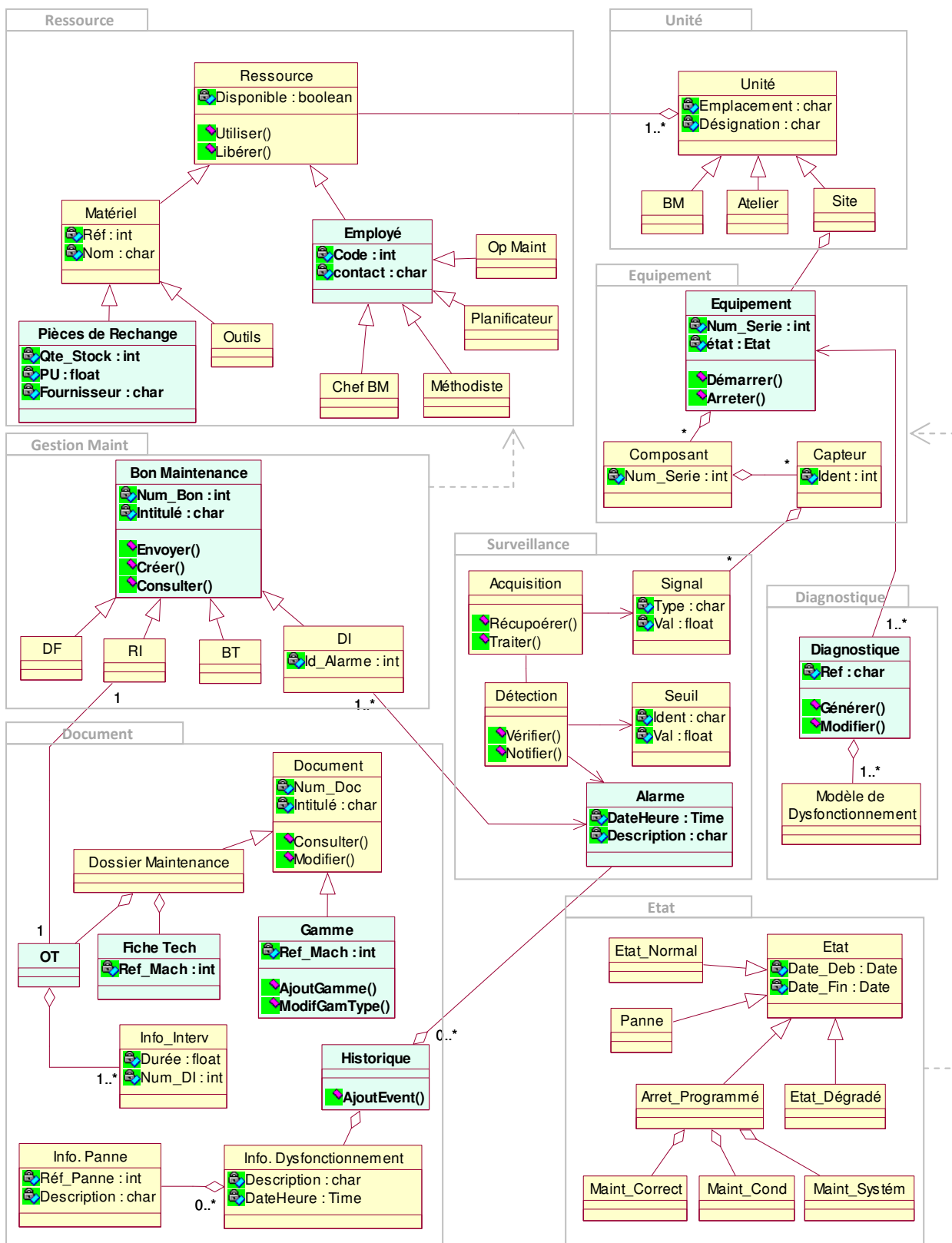
La figure 4.38 illustre le diagramme de classes UML correspondant aux principaux objets métier impliqués dans le système de la maintenance industrielle.

Ce modèle de classes décrit la structure statique du système de la maintenance en termes d'objets métier. Dans ce qui suit, nous décrivons sommairement le rôle de quelques objets métier présentés sur le modèle (figure 4.38).

Le package *gestion maintenance* constitue l'élément central dans le système de maintenance. Il permet de gérer toutes les informations sur l'intervention sous forme de *bon de maintenance*. Par ailleurs, d'autres packages servent de base pour assister les opérateurs lors de l'exécution de leurs tâches de maintenance, notamment les modules suivants : (1) *Ressources* qui englobe tous les objets métier fournissant les informations sur le personnel intervenant et le matériel utilisé en termes de pièces de rechange et l'outillage d'intervention, (2) *Document* constitue un module indispensable dans les activités de maintenance telles que, le diagnostique, préparation de l'OT, renseignement du bon de travaux (BT), etc.

En outre, les modules *Equipement* et *Unité* permettent de gérer l'ensemble des installations à maintenir, les ateliers, les sites de production, etc.

Enfin, les modules relevant de la surveillance et du diagnostique permettent d'une part, de contrôler l'état des équipements industriels, notifier les alarmes, détection des anomalies et d'autres part fournissent les outils nécessaires pour l'identification de l'origine et la cause des dysfonctionnements à travers la fonction du diagnostique.



DF : Demande de fourniture OT : Ordre de Travail DI : Demande d'intervention
 BT : Bon de Travaux RI : Rapport d'Intervention BM : Bureau de Méthodes

Figure 4.38 : Modèle de classes d'objets métier relatif au système de maintenance

Spécification des types de données échangées (TDE)

Les données d'échange sont les informations véhiculées entre les participants (consommateurs ou fournisseurs de service) à travers l'invocation des opérations de service. Cependant, les types de données d'échange (appelés TDE) établissent la sémantique, la structure et le format de ces données.

La figure 4.39 montre le modèle correspondant au « TDE Alarme », ses relations avec d'autres TDE et aussi avec les objets métier clé du système.

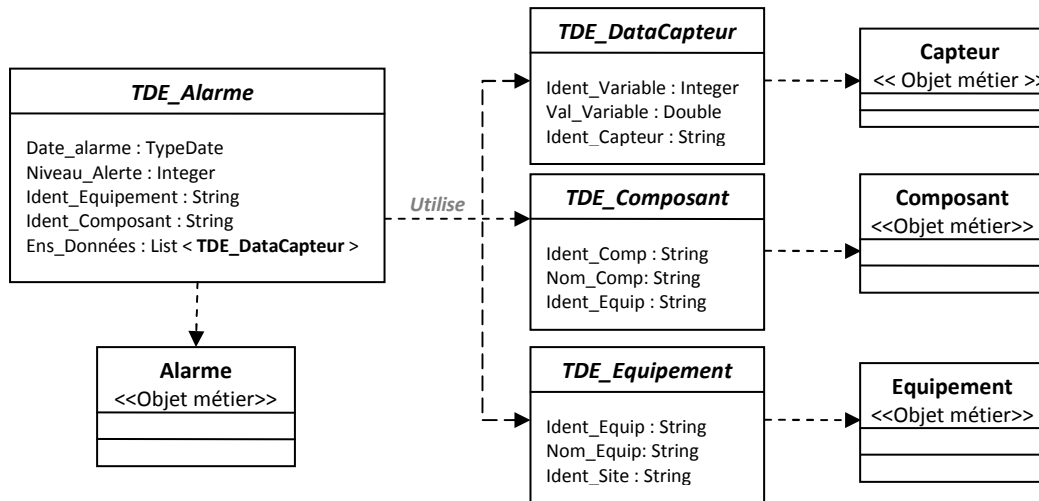


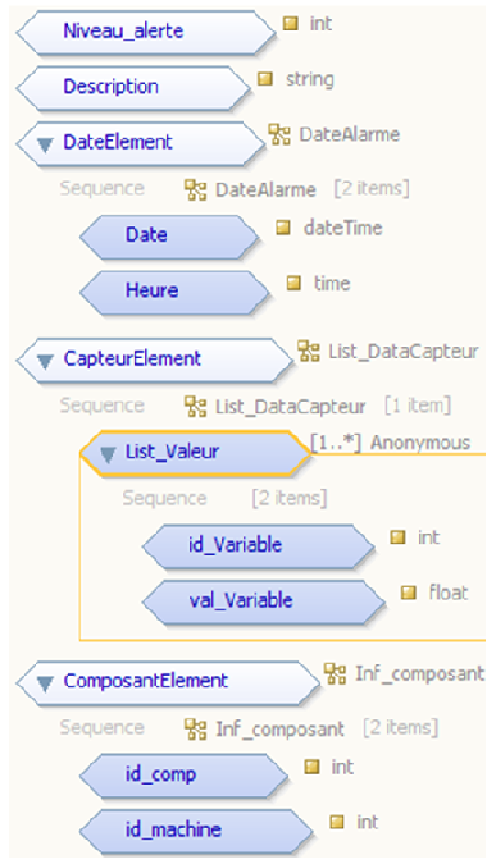
Figure 4.39 : Modélisation du « TDE Alarme ».

Comme illustré sur la figure 4.39, le type de données relatif à l'objet *alarme* dépend d'autres TDE, notamment ceux relatifs aux données issues des capteurs, d'équipements industriels et des composants de chaque machine.

De leur côté, les TDE *TDE_DataCapteur*, *TDE_Composant* et *TDE_Equipement* sont en relations directes avec les objets métier (*Capteur*, *Composant* et *Equipement*) gérés par les services CRUD au niveau de la base de données.

Après avoir modélisé les différents TDE nécessaires pour les échanges entre les services au sein du système de maintenance, l'étape suivante sera de formaliser ces TDE sous un format standard afin d'être utilisés par les services Web via leurs interface WSDL. Pour cela, le format *XML Schema* (fichiers de type XSD : *XML Schema Definition*) a été retenu pour la spécification des TDE.

La figure 4.40 illustre une représentation graphique de la structure du *TDE_Alarme* (partie gauche de la figure 4.35) et la spécification formelle correspondante (partie droite de la figure 4.40) sous format *XML Schema*. L'utilisation des représentations graphiques lors de la spécification des TDE est largement adoptée par de nombreux environnements de développement SOA tels que *SUN-NetBeans* (www.netbeans.org), *IBM-WebSphere* (www.ibm-websphere.com), *Oracle-JDeveloper* (www.oracle-jdev.com), etc. En effet, l'exemple montré sur la figure 4.40 a été généré en utilisant l'IDE *NetBeans*.



TDE_Alarme sous forme graphique

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:complexType name="DateAlarme">
    <xsd:sequence>
      <xsd:element name="Date" type="xsd:dateTime"/>
      <xsd:element name="Heure" type="xsd:time"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="List_DataCapteur">
    <xsd:sequence>
      <xsd:element name="List_Valeur" maxOccurs="unbounded">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="id_Variable" type="xsd:int"/>
            <xsd:element name="val_Variable" type="xsd:float"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="Inf_composant">
    <xsd:sequence>
      <xsd:element name="id_comp" type="xsd:int"/>
      <xsd:element name="id_machine" type="xsd:int"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:element name="Niveau_alerte" type="xsd:int"/>
  <xsd:element name="Description" type="xsd:string"/>
  <xsd:element name="DateElement" type="tns:DateAlarme"/>
  <xsd:element name="CapteurElement" type="tns:List_DataCapteur"/>
  <xsd:element name="ComposantElement" type="tns:Inf_composant"/>
</xsd:schema>
```

Fichier XML Schema (XSD) relatif à TDE_Alarme

Figure 4.40 : Spécification du TDE_Alarme à l'aide de XML Schema (XSD)

Spécification des interfaces

Les interfaces sont un mécanisme permettant de spécifier les traitements offerts par une classe d'objets ou un composant du système. Concrètement, l'interface est définie à travers une classe qui ne contient que la description des opérations visibles à l'extérieur, permettant ainsi de séparer la spécification des classes (ou composants de services) de leurs implémentations.

Le schéma de la figure 4.41 illustre un exemple d'interfaces des services métier déjà identifiés lors de la phase 1. Chaque interface possède une ou plusieurs opérations et chaque opération est définie par un ensemble de paramètres d'entrés ainsi que leurs types de sortie en concordance avec la spécification des TDE établie lors de l'étape précédente.

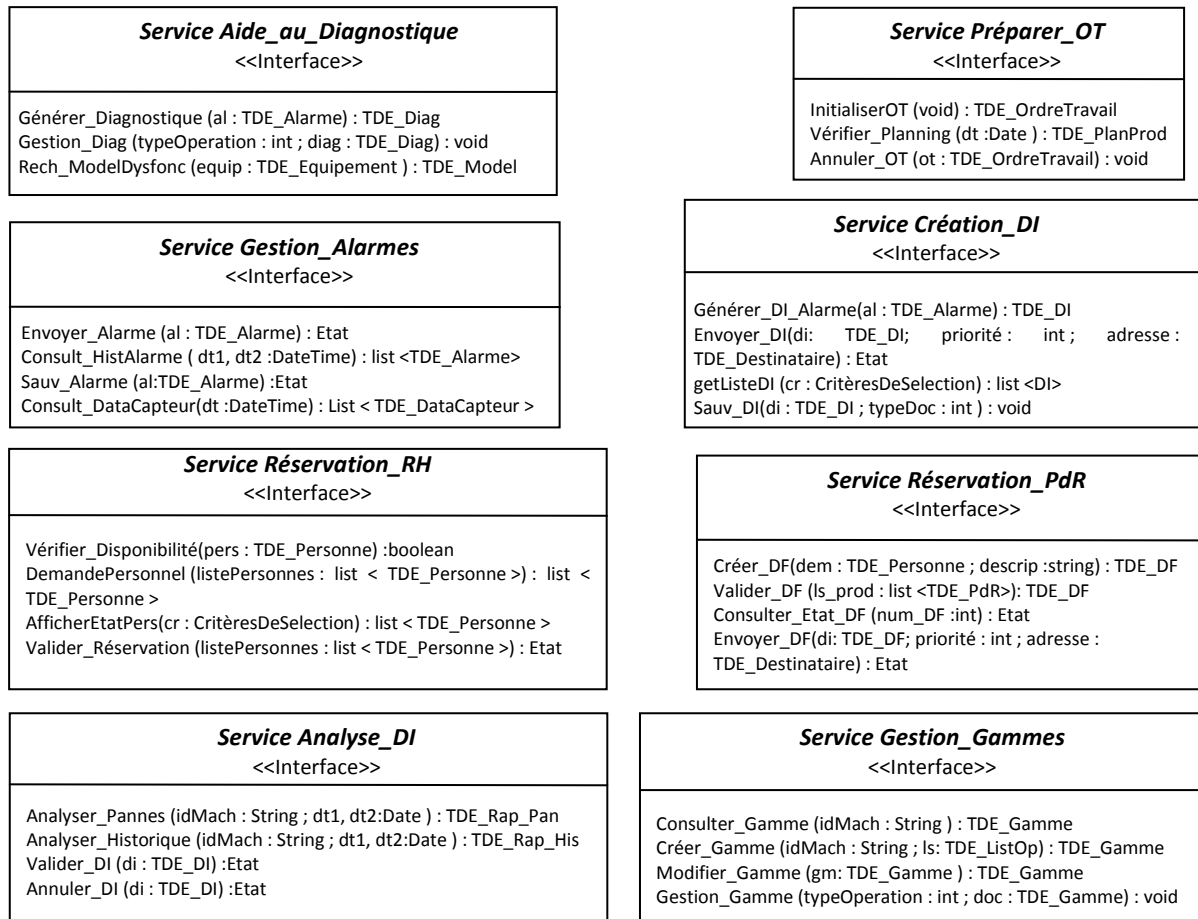


Figure 4.41 : Extrait du modèle d'interfaces des services métier

L'étape suivante sera de transformer ce modèle en un ensemble de fichiers de description WSDL, pour ensuite les déployer sur le serveur d'application ou dans le registre de services (UDDI). Un exemple de fichiers WSDL est donné en annexe (annexe H).

La dernière phase de spécification de services est relative à la définition des contraintes non-fonctionnelles qui complèteront le contrat de service entre le consommateur et le fournisseur du service Web.

Le tableau suivant dresse un ensemble de contraintes non-fonctionnelles liées à la qualité de service (*QoS : Quality of Service*), notamment en termes de performances, disponibilité et la sécurité.

Nom du paramètre QoS	Description	Exemple de valeur
Fiabilité	Mesure la capacité d'un service d'exécuter ses fonctions dans des conditions indiquées dans une période de temps spécifiée. Elle peut être mesurée par: MTBF: "Mean time between failure" - Temps moyen entre pannes. MTF: "Mean Time to Failure" - Temps moyen par panne. MTTT: "Mean Time To Transition" – Temps moyen pour la transition	MTBF=6 Mois
Disponibilité	Elle est la probabilité que le système soit actif. Elle est liée à la fiabilité.	0.9%

Capacité	Mesure la limite de demandes concourantes pour une performance garantie.	10 dem/s
Authentification	Comment le service authentifie-t-il des utilisateurs ou d'autres services qui peuvent avoir accès au service et aux données ?	Par : Nom d'utilisateur et mot de passe
Traçabilité et vérifiabilité	Est-il possible de tracer l'histoire d'un service lorsqu'une demande a été entretenue.	Oui

Tableau 4.5 : Exemple de contraintes non-fonctionnelles pour le contrat de service

La figure 4.42 montre un exemple de message SOAP définissant les caractéristiques du contrat de service en termes de qualité de service.

```

<?xml version="1.0" encoding="UTF-8" ?>
<envelope xmlns="http://schemas.xmlsoap.org/soap/envelope/">
<body>
<find_service businessKey="*" generic="1.0" xmlns="urn:uddi-org:api" maxRows="100">
<findQualifiers></findQualifiers>
<name>Préparer_OT</name> ← Nom du service Web
<qualityInformation>
<availability>0.9%</availability>
<fiability>
<mtbf>6 mois</mtbf>
<mtf>2 heures</mtf>
</fiability>
<capacity>10 requêtes par s</ capacity>
< Authentification> nom utilisateur & mot de passe </Authentification>
<Traceability>oui</Traceability>
</qualityInformation>
</find_service>
</body> </envelope>
    
```

} Définition des contraintes non-fonctionnelles

Figure 4.42 : Message SOAP définissant les contraintes QoS dans un contrat de service

IV.3.3. Spécification des composants

Après avoir défini les différents types de composants (service, fonctionnel et technique) contenus dans chaque sous-système et spécifié les services constituant l'architecture SOA dédiée aux systèmes de la maintenance, nous affinons dans cette dernière étape les modèles obtenus lors des étapes précédentes, notamment le modèle de composants. Cela consiste à concevoir chaque composant et éventuellement chaque sous-système sous forme de diagramme de classes UML. Le modèle obtenu décrit la structure interne des services Web à exposer et d'autres services internes à chaque sous-système.

Nous présentons à titre représentatif, la spécification des composants relatifs au domaine fonctionnel «*Ordre de Travail*». La figure 4.43 montre le diagramme de classes UML correspondant au modèle de composants déjà présenté lors de l'analyse des sous-systèmes (§IV3.1, figure 4.37).

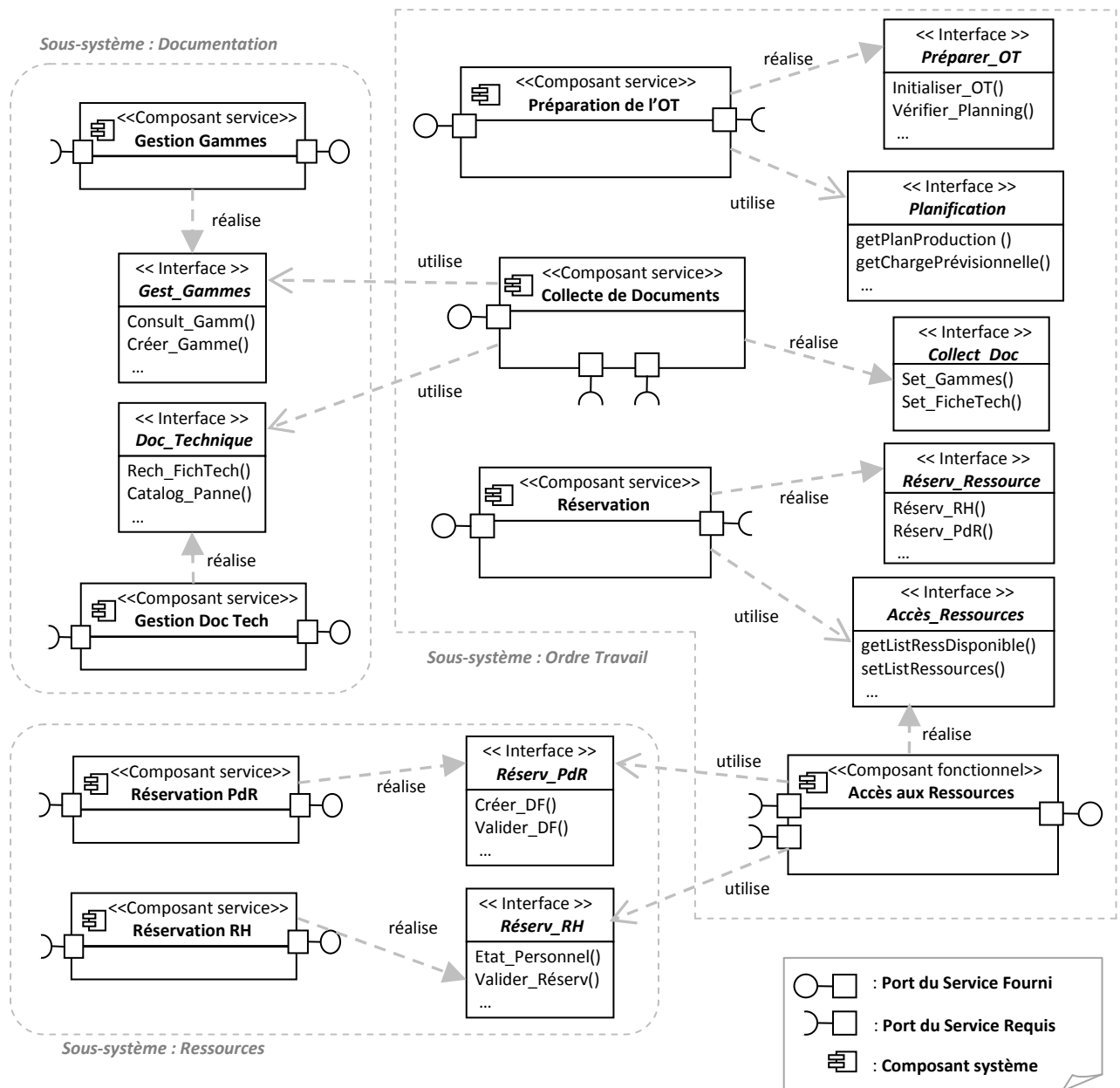


Figure 4.43 : Modèle conceptuel de classes des composants

L'extrait du modèle de classes relatif aux composants de services illustré sur la figure 4.43 décrit avec précision *les interfaces* pour les services offerts et leurs relations avec les autres services demandeurs. Il s'agit en effet d'un modèle décrivant la structure interne de chaque sous-système selon *une vue orientée services*.

Par ailleurs, l'orchestration des services dédiés au traitement de l'OT (*processus métier OT*) est réalisée via l'invocation des services métier composés (*Préparer_OT*, *Collect_Doc*, etc.) à travers leur port d'entrée/sortie. Ce dernier constitue le point d'interactions du service avec les autres partenaires (services Web, processus métier, etc.).

Cependant, pour des raisons de lisibilité, nous n'avons pas détaillé le modèle de classes d'implémentation contenues dans chaque composant de service.

Après avoir terminé la représentation conceptuelle des différents composants, nous arrivons dans ce qui suit à l'élaboration du « modèle de services ». Ce dernier offre une vue assez exhaustive sur les relations entre les différents services métier (implémentés sous forme de services Web) issus des sous-systèmes distincts.

La figure 4.44 illustre un extrait du modèle de services qui met en avant les services métier interagissant au sein du processus métier « traitement de l'OT ».

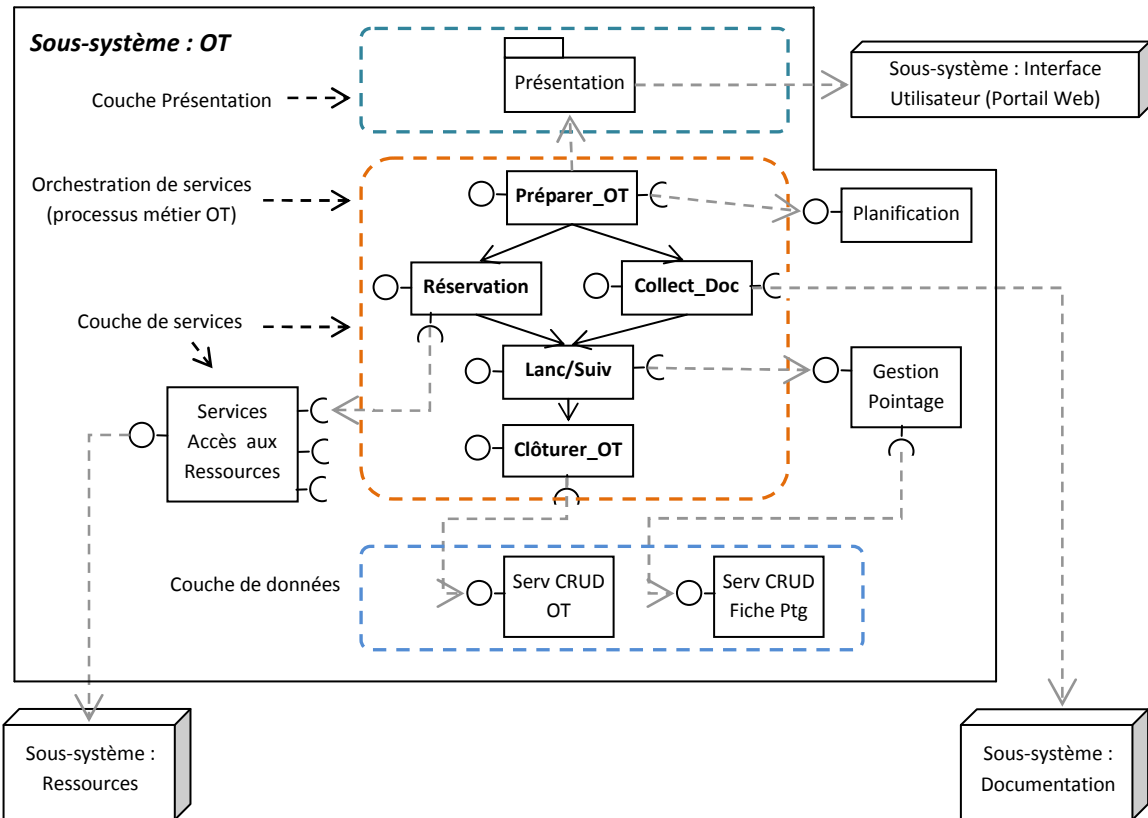


Figure 4.44 : Extrait du modèle de services

Le modèle de services présenté sur la figure 4.44 facilite le processus d'allocation de services vers les différents niveaux de l'architecture SOA comme déjà évoqué dans la méthode SOMA (chapitre 3, §III.3).

La dernière étape de notre démarche consiste à présenter l'architecture globale du système de la maintenance (figure 4.45). Cette architecture consolide l'ensemble des éléments conceptuels déjà définis (domaines fonctionnels, services métier, processus, composants, etc.) tout en se référant au modèle architectural de SOA.

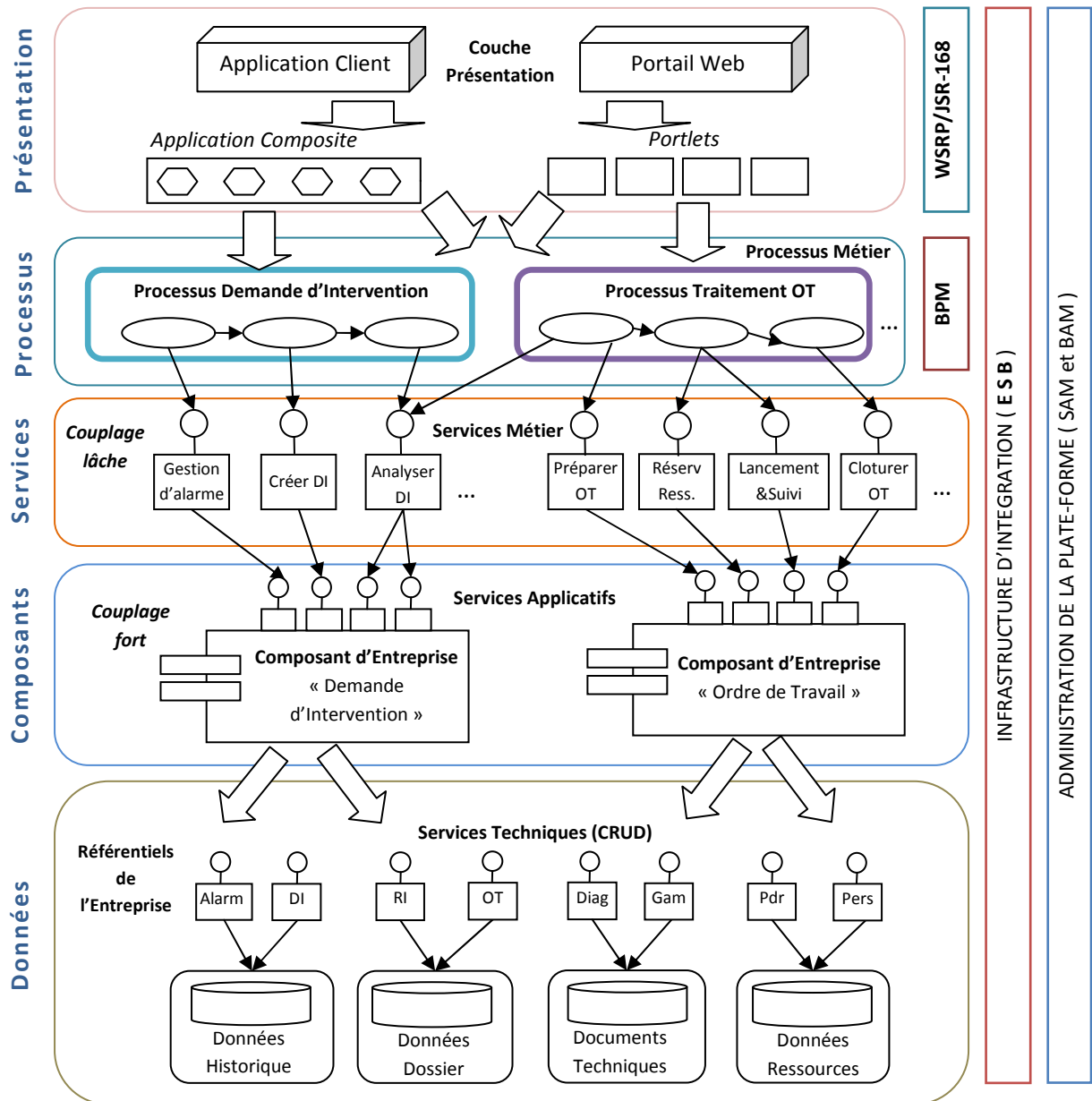


Figure 4.45 : Architecture Orientée Services pour un système de gestion de la Maintenance Industrielle

A travers le schéma de la figure 4.45 nous pouvons dégager un ensemble d'éléments architecturaux liés à SOA. Le premier élément est celui de découpage vertical du système en termes de domaine fonctionnels (représentés par les composants d'entreprise sur la figure 4.45). Chaque domaine fonctionnel représente un bloc applicatif pouvant interagir facilement avec d'autres blocs en utilisant les services Web. En outre, nous remarquons que ces domaines couvrent globalement toutes les couches du modèle SOA, notamment les couches : présentation, orchestration de services (processus métier), services métier, composants d'entreprise (ou sous-système) et la couche de données (services CRUD).

Par ailleurs, le schéma de la figure 4.45 montre le découpage horizontal du système à travers les couches dédiées aux processus métier et les services métier. Ces derniers peuvent être

composés dynamiquement afin de fournir d'autres services d'une plus grande granularité et offrant ainsi une nouvelle fonctionnalité pour le processus métier global de la maintenance.

IV.4. Conclusion sur la phase 3

Dans cette phase, nous avons illustré la conception des différents éléments constituant l'architecture SOA dédiée aux systèmes de la maintenance industrielle. Nous avons conduit cette phase suivant trois étapes essentielles :

Dans l'étape 1, nous avons décrit et illustré les différents modèles conceptuels des processus métier en se basant sur des scénarios métier relatifs à la maintenance. Nous avons abordé cette étape par une spécification des interactions Homme-Machine en mettant en avant les flux des différentes tâches (humaines et automatique). Cette spécification permet de définir d'une façon succincte la logique métier du processus métier à travers l'enchaînement d'invocation des services et les variables d'échanges. Par ailleurs, nous avons utilisé cette spécification dans la modélisation des processus métier en utilisant la notation BPMN et le langage BPEL.

L'étape 2 a été consacrée pour la conception de l'interface utilisateur. Nous avons présenté une architecture relative à la couche présentation de SOA en se basant sur les services orientés présentation spécifiés à l'aide de la norme WSRP. En outre, nous avons montré à l'aide des maquettes et des prototypes les différentes vues du système via l'interface utilisateur.

Enfin, dans l'étape 3, on a procédé à la représentation conceptuelle des différents types de composants et services. Cette étape est la plus importante dans toute la démarche car c'est à travers celle-ci qu'on doit réaliser l'assemblage des différents artefacts (élément méthodologiques) déjà définis au cours des étapes ultérieures de la démarche. Les modèles conceptuels présentés dans cette étape visent à fournir des vues logiques du système selon plusieurs niveaux d'abstraction (sous-système, composant et classes d'objets). Par ailleurs, nous avons achevé cette étape par la présentation de l'architecture SOA globale dédiée au système de maintenance. Cette architecture englobe l'ensemble des aspects déjà étudiés dans le cadre de SOA, notamment ceux relatifs aux niveaux d'abstraction, processus transverse, services composés, etc.

V. Conclusion

L'étude de cas présentée dans ce chapitre, est relative à la maintenance industrielle impliquant l'utilisation des nouvelles technologies d'information et de communication et aussi impliquant de nouveaux moyens de communication et les tâches coopératives de la maintenance.

Nous avons appliqué la démarche proposée dans le chapitre 3 à travers ses principales phases d'analyse et de modélisation orientée services. Cette application de la démarche sur un cas d'étude réel relevant de la maintenance nous permet d'évaluer et de valider en partie les différentes phases et étapes énoncées dans le chapitre 3.

Par ailleurs, l'étude de cas réalisée ne couvre pas tous les domaines fonctionnels d'un système de maintenance (analyse des résultats, service achat, etc.) pour des raisons de simplification dans la présentation de la validation de la démarche. En revanche, les domaines et les services choisis constituent des éléments fondamentaux dans tout système de maintenance et qui sont en relation avec la plupart des autres domaines. Néanmoins, l'application et la mise en œuvre de la démarche SOA proposée vise à couvrir au maximum le processus de développement d'une architecture orientée services à travers les phases d'analyse et de conception.

Les résultats obtenus de cette application peuvent être directement exploités par les développeurs lors de la phase d'implémentation. Notons cependant, pour des raisons liées à la taille du mémoire, cette mise en œuvre ne traite pas les aspects liés au filtrage progressif des services à exposer et les décisions à entreprendre sur le déploiement des services et les technologies de réalisation du système (langages, réseaux, serveurs, etc.).

Enfin, nous insistons sur le fait que la démarche proposée est une consolidation de deux méthodes issues de l'orienté services : (1) méthode de spécification d'IHM à base des services Web [Idoughi, 08] et (2) méthode SOMA d'IBM qui couvre les phases d'identification et de spécification des services. Notons cependant, que cette fusion des deux méthodes utilisant des approches et des outils différents n'étant pas évidente. Néanmoins, nous pensons que cette contribution peut constituer un atout pour les concepteurs des systèmes orientés services, précisément dans la cadre des systèmes de gestion de la maintenance industrielle.

En conclusion, la démarche proposée nécessite d'être complétée et améliorée en se focalisant par exemple, sur des domaines spécifiques de l'approche orientée services tels que l'orchestration et la chorégraphie des services, composition des services, etc. Cela peut constituer des perspectives de recherche dont les principales sont exposées dans le chapitre suivant.

CHAPITRE V : CONCLUSION GENERALE ET PERSPECTIVES DE RECHERCHE

I. Conclusion générale

Notre travail s'inscrit dans la tendance actuelle de fournir une plateforme cohérente basée sur les technologies du Web et intégrant l'ensemble des systèmes informatiques impliqués dans la fonction de la maintenance industrielle. L'objectif principal visé dans ce mémoire, est de montrer l'intérêt d'utiliser les services Web en se référant au modèle d'architecture orientée services (SOA) dans les systèmes de maintenance d'une part et d'autre part, la nécessité de conduire le projet SOA selon une démarche d'analyse et de modélisation rationnelle.

Dans le chapitre 1, nous avons décrit les spécificités de la maintenance industrielle en termes de concepts, de méthodes et de stratégies de collaboration adoptées par les acteurs métier. Par ailleurs, nous avons focalisé notre étude sur les architectures des systèmes de maintenance et l'apport des nouvelles technologies de l'information et de la communication dans l'évolution de ces systèmes vers d'autres formes de maintenance notamment la e-maintenance qui est basée sur le réseau Internet.

Par conséquent, de nouvelles motivations sont apparues suite à ces évolutions, notamment sur les problèmes d'intégration tels que le fonctionnement en silos applicatifs, l'interfaçage des applications et les aspects liés aux IHM dédiées aux systèmes de maintenance industrielle. Ajoutant à cela, les nouveaux besoins des entreprises en termes d'externalisation de la maintenance et du travail coopératif, voir collaboratif, entre les unités de maintenance appartenant à des organisations différentes sont aussi d'autres exigences.

Dans le chapitre 2, nous avons présenté les éléments et concepts clé de l'approche orientée services et les technologies liées aux services Web. Nous avons expliqué que cette nouvelle approche, baptisée SOA (*Services Oriented Architecture*), vise à organiser les applications informatiques en se basant sur *l'émergence d'une couche de services*. Chaque service de cette couche permet d'encapsuler les traitements et les données en masquant ainsi l'hétérogénéité des différents systèmes communicants.

Par ailleurs, nous avons insisté sur la vocation de SOA dans le développement des applications autour des briques fonctionnelles (services) autonomes, faiblement couplées et qui peuvent être composées dynamiquement pour fournir de nouvelles fonctionnalités pour les utilisateurs. Nous avons aussi souligné la différence entre SOA, qui constitue une *approche* de développement informatique et les services Web qui constituent la *technologie* la plus répondue pour l'implémentation d'une architecture SOA.

Dans le chapitre 3, nous avons étudié deux approches de modélisation orientée services fondées sur des visions différentes mais complémentaires. La première proposée dans

[Idoughi, 08] se focalise sur la spécification des IHM à base des services Web et la seconde, proposée par IBM [Arsanjani, 04], basée sur la décomposition progressif du système en termes de services de différentes granularités. En effet, nous avons confronté ces deux types d'approches en vue de ressortir de nouveaux éléments à prendre en compte dans notre démarche d'analyse et de modélisation d'un système de maintenance. Pour cela, nous avons présenté une synthèse sur les deux approches étudiées tout en se référant aux problématiques déjà énoncées lors du chapitre 1.

Suite à cette étude, nous avons proposé une démarche globale d'analyse et de modélisation orientée services en vue de développement d'un système de maintenance industrielle. Cette démarche a pour vocation de fédérer les techniques liées à la spécification des interactions Homme-Machine, qui caractérise les systèmes de maintenance, et les approches d'analyse et de conception des différents constituants d'un domaine métier donné. Par ailleurs, nous avons souligné que la démarche proposée vise à couvrir au maximum les phases d'analyse et de conception dans le processus de développement des applications orientées services.

Cette démarche s'articule sur trois phases : (1) la phase relative à l'étude de l'organisation de la maintenance et l'analyse métier, (2) la phase d'analyse et expression des besoins des utilisateurs et (3) la phase de conception de l'architecture orientée services. Chacune de ces phases a été succinctement détaillée.

Enfin, le chapitre 4 traite de la mise en œuvre et de la validation de l'approche proposée sur une étude de cas réel relative à la maintenance industrielle dans une unité de conditionnement des huiles d'un groupe agro-alimentaire. L'étude de cas a été menée suivant les trois phases globales de la démarche orientée services proposée.

II. Perspectives

Notre travail est avant tout un travail d'analyse et de modélisation d'une architecture orientée services dédiée aux systèmes de maintenance rentrant dans le cadre de travaux de recherche antérieurs, notamment celui [Idoughi, 08] dans le domaine de la supervision industrielle et ceux liés au projet européen *Proteus* [Bangemann *et al.*, 05 ; Rebeuf *et al.*, 05] dans le cadre de la proposition d'une plateforme générique pour la e-maintenance.

Cependant, ce travail peut faire l'objet de plusieurs perspectives de recherche dont les principales sont décrites ci-après.

1) Validation de l'approche

L'étude de cas traitée dans ce mémoire a été relative à la maintenance industrielle à travers ses fonctions et les processus métier impliqués. Par ailleurs, nous envisageons d'appliquer la démarche proposée sur d'autres domaines métier tels que, *la gestion d'une organisation hospitalière complexe*, où plusieurs systèmes informatiques de gestion, affiliés à des hôpitaux différents, peuvent communiquer sur le réseau Internet pour l'échange et le partage sécurisé des différentes ressources informationnelles sur leurs patients, leurs compétences, etc.

Par ailleurs, nous envisageons de l'appliquer aussi sur un domaine relatif au système d'information national de la *e-gouvernance* dans le cadre du programme « e-Algérie 2013 »¹⁸.

2) Amélioration de l'approche

Afin d'améliorer l'approche proposée, nous envisageons aussi de traiter d'une façon succincte les aspects liés à l'interaction entre les services, notamment dans l'utilisation des langages de spécification standards dont WS CDL¹⁹ pour réaliser des communications *ad-hoc* entre les différents services Web appartenant à des fournisseurs de services distincts. Par ailleurs, d'autres améliorations peuvent être apportées au niveau de la composition des services Web à travers le modèle SCA²⁰ et aussi dans la recherche dynamique des services à travers les référentiels de services UDDI.

3) Extension de la démarche proposée

Afin d'étendre notre démarche d'analyse et de modélisation orientée services, nous envisageons d'inclure d'autres phase potentiellement importantes pour la mise en œuvre plus ou moins complète d'une architecture SOA. Précisément, nous pensons à définir tous les aspects liés à l'infrastructure logicielle et matérielle permettant le déploiement de la solution SOA pour les systèmes de maintenance, notamment les serveurs d'application et de base de données, les pare-feux, etc. De plus, les décisions sur les technologies à entreprendre lors de la phase d'implémentation est primordiale, notamment sur la plateforme applicative à utiliser (J2EE, DotNet, etc.), les technologies des pages Web (JSP, JSF, Strut, etc.), etc.

4) Utilisation de SoaML (Service oriented architecture Modeling Language)

SoaML²¹ ou langage de modélisation des architectures orientée services, est une nouvelle spécification de l'organisation OMG destinée aux développements en environnements SOA basée sur UML [Berre, 08]. L'objectif de SoaML est de fournir aux utilisateurs d'UML un moyen de modéliser les concepts de base d'une architecture orientée services, c'est-à-dire les notions de *consommateur* de services, *fournisseur* de services, et *contrat de services*.

Cette spécification n'a pas été prise dans ce mémoire vu son apparition récente et sa maturité (OMG a prévu de finaliser les travaux sur SoaML en mars 2009). Néanmoins, l'utilisation de cette spécification dans la phase 3 de notre démarche, précisément lors de la conception du modèle de services, constituerait un atout considérable dans l'affinement de la conception de l'architecture SOA pour les systèmes de la maintenance industrielle.

En conclusion, les approches orientées services reste un domaine très vaste et complexe, qui laisse envisager plusieurs travaux de recherche en perspectives et qui nécessite des applications sur d'autres domaines métier.

¹⁸ **e-Algérie** : Projet lancé en mars 2009 par le ministère de la poste et des technologies de l'information et de la communication (MPTIC) d'Algérie visant à renforcer les performances de l'économie nationale, des entreprises et de l'administration à travers le développement et l'accélération de l'usage des nouvelles technologies de l'information et de la communication (NTIC). Source : <http://www.mptic.dz>.

¹⁹ **WS-CDL** (Web Services Choreography Description Language) : Normalisé par W3C, WS-CDL est un langage de description de la collaboration paire-à-paire entre les participants (services Web). Source : www.w3.org/TR/2004/WD-ws-cdl-10-20041217/

²⁰ **SCA** (Service Component Architecture) : Spécifications d'OASIS. Source : <http://www.oasis-open.org/committees>.

²¹ **SoaML** (Service oriented architecture Modeling Language) : Spécification d'OMG. Source: <http://www.omg.org/docs/ad/08-08-04.pdf>

Références bibliographiques

- [Abbou, 03] Rosa Abbou. Thèse Doctorat de l'UJF « *Contribution à la mise en œuvre d'une Maintenance Centralisée : Conception et Optimisation d'un Atelier de Maintenance* » Université Joseph Fourier - GRENOBLE 1, Octobre 2003.
- [AFNOR, 01] AFNOR : Association Française de Normalisation, « *Terminologie de la maintenance, Norme européenne NF EN 13306* ». <http://www.afnor.org/portail.asp>, 2001.
- [Alami, 05] Ahmed ALAMI « *L'Architecture Orientée Services* », SOA-Architect.Net 2006. source : <http://www.soa-architect.net/publications/soa/soa/index.php>.
- [Arsanjani, 04] Ali Arsanjani. « *Service-oriented Modeling and Architecture : How to identify, specify, and realize services for your SOA* », IBM Novembre 2004.
- [Auberville, 04] Jean-Marie Auberville. « *Maintenance industrielle : de l'entretien de base à l'optimisation de la sûreté* ». Paris : Ellipses, 2004.
- [Bangemann *et al.*, 05] Thomas Bangemann, Xavier Rebeuf, Denis Reboul, Andreas Schulze, Jacek Szymanski, Jean-Pierre Thomesse, Mario Thron, Noureddine Zerhouni. « *PROTEUS – Creating Distributed Maintenance Systems through an Integration Platform* » CEGELEC , LORIA 2005.
- [Barre et Fonash, 04] Dalle Barre et Peter Fonash « *Supervisory Control and Data Acquisition (SCADA) Systems* » office of the manager national communications system, Octobre 2004.
- [Berre, 08] Arne J. Berre. « *Service oriented architecture Modeling Language (SoaML) - Specification for the UML Profile and Metamodel for Services (UPMS)* » Revised Submission. Document de l'OMG ajouté le : 04/08/2008 source (<http://www.omg.org/docs/ad/08-08-04.pdf>)
- [Bonnet, 05 a] Pierre BONNET. « *Cadre de référence SOA* ». Orchestra Network - Février 2005.
- [Bonnet, 05 b] Pierre BONNET. « *Cadre de référence Web Service* ». Orchestra Network - Octobre 2005.
- [Chauvet, 02] Jean-Marie Chauvet « *Services Web avec SOAP, WSDL, UDDI, ebXML...* ». EYROLLES 2002.
- [Dennergy, 07] Olivier Dennergy « *Méthodologie SOMA : retour d'expérience dans la spécification des services* ». IBM SOA Summit (www.ibm.com), 2007.
- [Endrei *et al.*, 04] Mark Endrei, Jenny Ang, Ali Arsanjani, Sook Chua, Philippe Comte, Pål Krogdahl, Min Luo, Tony Newling. « *Patterns: Service-Oriented Architecture and Web Services* ». IBM RedBooks (www.rebook.ibm.com) 2004.

- [Ferrari, 07] Michaël Ferrari, « *BPMN, la norme du BPM Le point sur les véritables possibilités de BPMN* ». BPMS.info 2007.
- [Fournier-Morel *et al.*, 06] Xavier Fournier-Morel, Pascal Grojean, Guillaum Plouin, Cyril Rognon. « *SOA : Guide des Architectes* ». DUNOD 2006.
- [Frédéric, 03] Marc Frédéric. « *Mettre en œuvre une GMAO : maintenance industrielle, service après-vente, maintenance immobilière* ». DUNOD, 2003.
- [Ganci *et al.*, 06] John Ganci, Amit Acharya, Jonathan Adams, Paula Diaz de Eusebio, Gurdeep Rahi, Diane Strachan, Kanako Utsumi, Noritoshi Washio. « *Patterns: SOA Foundation Service Creation Scenario* ». IBM RedBooks (www.rebook.ibm.com) Septembre 2006.
- [Gustienne et Gustas, 07] Prima Gustiene, Remigijus Gustas. « *Introducing service-orientation into system analysis and design* ». Department of Information Systems, Karlstad University, Sweden 2007.
- [Idoughi *et al.*, 08] Djilali Idoughi, Moussa Kerkar, Christophe Kolski. « *Towards new web services based supervisory systems in complex industrial organizations: basic principles and study case* ». Juillet 2008.
- [Idoughi et Kolski, 06 a] Djilali Idoughi, Christophe Kolski. « *Approches orientées services Web de l'IHM de supervision : nouvelles solutions technologiques pour les ingénieurs et nouvelles problématiques pour les ergonomes ?* ». ErgoIA, 2006.
- [Idoughi et Kolski, 06 b] Djilali Idoughi, Christophe Kolski. « *Towards new Web Services based Supervisory Systems dedicated to Nomadic Operators* ». EAM, 2006.
- [Idoughi, 08] Djilali Idoughi. Thèse Doctorat : « *Contribution à un cadre de spécification et conception d'IHM de supervision à base de services web dans les systèmes industriels complexes : Application à une raffinerie de sucre* ». Université de Béjaïa, Septembre 2008.
- [Joffroy *et al.*, 07] Cédric Joffroy, Sébastien Mosser, Mireille Blay-Fornarino, Clémentine Nemo, « *Des Orchestrations de Services Web aux Aspects* ». Laboratoire I3S (CNRS – UNSA), 2007.
- [Kadima et Monfort, 03] Hubert Kadima et Valérie Monfort. « *Les services Web : techniques, démarches et outils, XML, WSDL, SOAP, UDDI, Rosetta, UML* ». Edition Dunod, Paris, 2003.
- [Kaffel, 01] Hédi Kaffel. Thèse Doctorat : « *La maintenance distribuée : concepts, évaluation et mise en œuvre* ». Université Laval, Québec, Canada, Octobre 2001.

- [Kolski et Millot, 93] Christophe Kolski, Patrick Millot « *Problems in telemaintenance and decision aid criteria for telemaintenance system design* » International Journal of Industrial Ergonomics, 1993.
- [Manouvrier et Ménard, 07] Bernard Manouvrier, Laurent Ménard « Intégration applicative EAI, B2B, BPM et SOA ». Paris : Lavoisier, 2007.
- [Muller et Gaertner, 02] Pierre-Alain Muller, Nathalie Gaertner. « *Modélisation Objet avec UML* », Eyrolles, Deuxième édition 2002.
- [Paterno, 00] Paternò Fabio. « *Model Based Design and Evaluation of Interactive applications* », Springer-Verlag, London, 2000.
- [Porter-Roth, 05] Bud Porter-Roth Associates « *Introduction to Service-Oriented Architectures for Electronic Document Management Systems* », Westbrook Technologies Inc, Septembre 2005.
- [Portier, 07] Bertrand Portier « *SOA terminology overview, Part 3 Analysis and design* » disponible à (<http://www.ibm.com/developerworks/webservices/library/ws-soa-term3/index.html>). IBM, mai 2007.
- [Rasovska et al., 07] Ivana Rasovska, Brigitte Chebel-Morello, Noureddine Zerhouni. « *Classification des différentes architectures en maintenance* ». Laboratoire d'Automatique de Besançon UMR CNRS, juin 2007.
- [Raymond, 07] Gilbert Raymond. « *SOA : Architecture Logique. Principes, structures et bonnes pratiques* » SOFTEAM (www.softeam.fr) février 2007.
- [Rebeuf et al., 05] X. Rebeuf, N. Blanc, F. Charpillet, D. Cheve, A. Dutech, C. Lang, L. Pélissier, J.P. Thomesse. « *Proteus, des web services pour les systèmes de maintenance* » Cegelec, Nanterre, France. LIFC, Besançon, France. LORIA, Nancy, France 2005.
- [Rivard et al., 05] Rivard F., Brendel C., Buche N., Delayre S., Mocaër A., Nevers J. « *SOA et urbanisme, Le rôle des Architectures Orientées Services dans l'alignement métier des Systèmes d'Information*. Unilog Management, 2005.
- [Rodriguez et al., 07] Juan R. Rodriguez, Alex Barbosa Coqueiro, Belen Gonzalez Agudo, Sunil Patel. « *Building Composite Applications* ». International Technical Support Organization. IBM Redbooks (www.redbooks.ibm.com) Juillet 2007.
- [Salatgé, 06] Nicolas Salatgé. Thèse Doctorat : « *Conception et mise en œuvre d'une plate-forme pour la sûreté de fonctionnement des Services Web* ». Institut National Polytechnique de Toulouse, Décembre 2006.
- [Schaeck et Thompson, 03] Thomas Schaeck et Richard Thompson. « *Web Services for Remote Portlets (WSRP) Whitepaper* » (source : www.oasis-open.org/committees/download.php/2634) OASIS, May 2003.
- [Stephen, 04] Stephen A. Livre blanc « *Introduction to BPMN* », IBM Corporation, BPTrends Juillet, 2004.

- [St-Marseille et Lapointe, 97] Marc St-Marseille, Jean-Brune Lapointe. « *La gestion des équipements vers l'entretien préventif* ». ASP Métal Electrique 1997.
- [Tajri, 05] I.Tajri, H. EL Aoufir, D. Bouami. « *Modélisation des activités de la maintenance par SADT. Démarche et application aux installations de signalisation ferroviaire à l'ONCF* » CPI'2005 – Casablanca, Maroc.
- [Tararykine, 05] Viatcheslav Tararykine. Thèse Doctorat : « *Modélisation des Flux d'Information dans un Système de E-maintenance* ». UFR des Sciences et Techniques de l'Université de Franche-Comté, Novembre 2005.
- [Vasiliev, 07] Yuli Vasiliev. « *SOA and WS-BPEL: composing service-oriented solutions with PHP and ActiveBPEL* ». Packt Publishing, Septembre 2007.
- [Vauquier et Bonnet, 06] Dominique Vauquier et Pierre Bonnet. « *Méthode Praxeme et SOA* » Unilog Management-Orchestra Networks, Juin 2006.
- [Way, 08] Mike Way. Livre Blanc : « *Service-Oriented Architecture (SOA) : From Introduction through Implementation* ». Learning Tree International 2008.
- [Zimmerman *et al.*, 04] Olaf Zimmermann, Pal Krogdahl, Clive Gee. « *Elements of Service-Oriented Analysis and Design : An interdisciplinary modeling approach for SOA projects* », IBM Juin 2004.

ANNEXE A : Description sommaire des couches de base des services Web

1. SOAP (Simple Object Access Protocol)

SOAP est un protocole de RPC orienté objet bâti sur XML. Il permet la transmission de messages entre les objets distants, ce qui veut dire qu'il autorise un objet à invoquer des méthodes d'objets physiquement situés sur un autre serveur.

Le protocole SOAP est composé de deux parties :

- Une enveloppe, contenant des informations sur le message lui-même afin de permettre son acheminement et son traitement,
- Un modèle de données, définissant le format du message, c'est-à-dire les informations à transmettre.

Un message SOAP se compose de trois éléments : un élément racine appelé Enveloppe et deux autres éléments intégrés à l'enveloppe, appelés En-tête (optionnel) et Corps (obligatoire).

L'Enveloppe doit obligatoirement contenir un élément Corps. S'il contient également un élément En-tête, non obligatoire, il ne doit contenir que ce seul En-tête. L'En-tête doit obligatoirement être placé avant le Corps.

L'En-tête utilise un code XML valide, bien formé (*well-formed*) et qualifié par un espace de noms (*namespace*) pour décrire le contenu et le mode de traitement du message. Chaque élément contenu dans l'En-tête est appelé Bloc d'en-tête. Ces blocs d'informations fournissent des renseignements sur la forme des structures de données incluses, sur l'application de sécurité, sur l'identité de l'expéditeur et sur le routage souhaité.

La partie corps (Body) porte les données propres au message, et matérialise la requête ou la réponse SOAP. La figure A.1 montre les éléments d'un message SOAP.

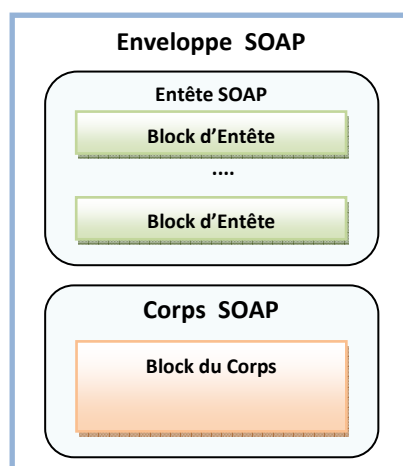


Figure A.1 : Structure générale d'un message SOAP [Kadima et Monfort, 03]

2. WSDL (Web Services Description Language)

Le langage WSDL est une syntaxe XML utilisée pour définir l'interface générale des services web. Cette définition intègre les composants fondamentaux suivants :

- Informations sur toutes les fonctions disponibles publiquement ;
- Définitions abstraites des données à transmettre ;
- Informations sur le type de données pour tous les messages XML ;

- Informations obligatoires sur le protocole de transfert à utiliser spécifiquement ;
- Informations de type Adresse pour localiser le service à spécifier.

La figure A.2 présente un schéma décrivant la structure globale d'un document WSDL.

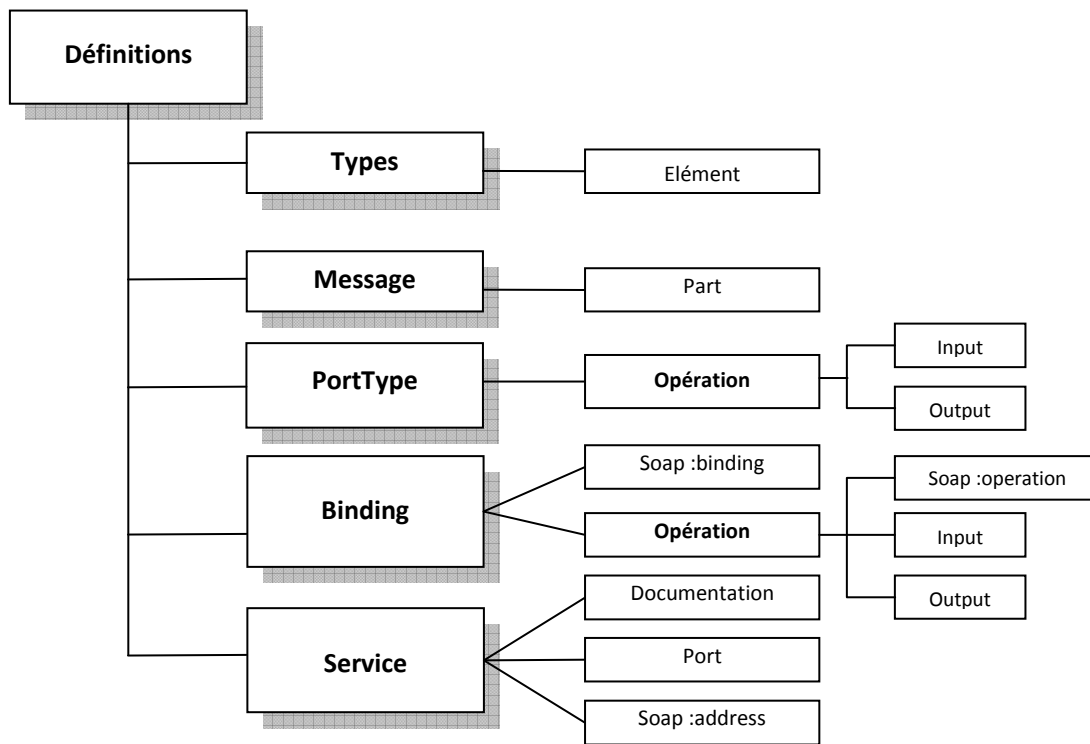


Figure A.2 : Structure d'un document WSDL [Kadima et Monfort, 03]

- **Définitions** : cette partie contient le nom du service décrit et les espaces de noms (*namespaces*) faisant référence aux types utilisés dans le document ;
- **Types** : Permet la définition des éléments abstraits présents dans le document WSDL ;
- **Message** (name, (part())) : Ce bloc est caractérisé par les éléments suivants :
 - Deux types de message IN et OUT. Si aucun argument n'est nécessaire, il n'y a pas de messages IN.
 - Définition abstraite des messages échangés entre deux nœuds ;
 - Peut être composé de plusieurs parties (Parts) correspondant aux paramètres passés aux fonctions.
- **Part** (nom, type ou element) : Cet élément peut être défini comme un type (simple ou complexe) ou un élément, et l'ordre des « parts » dépend de la signature de la méthode ;
- **portType** (name, operation(name,(input msg, output msg))) : Il correspond à une interface IDL en CORBA. Il contient les classes accessibles. Si le service propose différentes classes, leur nom doit être différent. A chaque portType sont associées des opérations, correspondant aux méthodes. Pour chaque méthode on définit le message d'entrée et de sortie. Les opérations peuvent être de natures différentes : unidirectionnelle, requête/réponse, sollicitation/réponse et notification ;
- **Binding** : Permet de spécifier quel protocole d'invocation utiliser HTTP GET/POST, SOAP, SMTP, FTP, etc. ;
- **Service** (name, port, documentation()) : Cet élément permet de spécifier le nom (*name*), où se trouve la documentation concernant le service mais aussi où envoyer les messages d'invocation. Le Port spécifie l'adresse (URI) à laquelle un service peut être invoqué. En fin, le paramètre Documentation permet aux développeurs d'apporter plus de précisions sur le service.

3. UDDI (Universal Description, Discovery and Integration)

UDDI est un annuaire de services basé sur XML et plus particulièrement destiné aux services Web. UDDI se divise en deux parties : un registre contenant toutes les métadonnées spécifiques à un service web (y compris une référence à la description WSDL du service) et une série de définitions WSDL de type Port permettant de modifier et de parcourir le registre.

La figure A.3 illustre le principe de fonctionnement de l'annuaire UDDI [Chauvet, 02]:

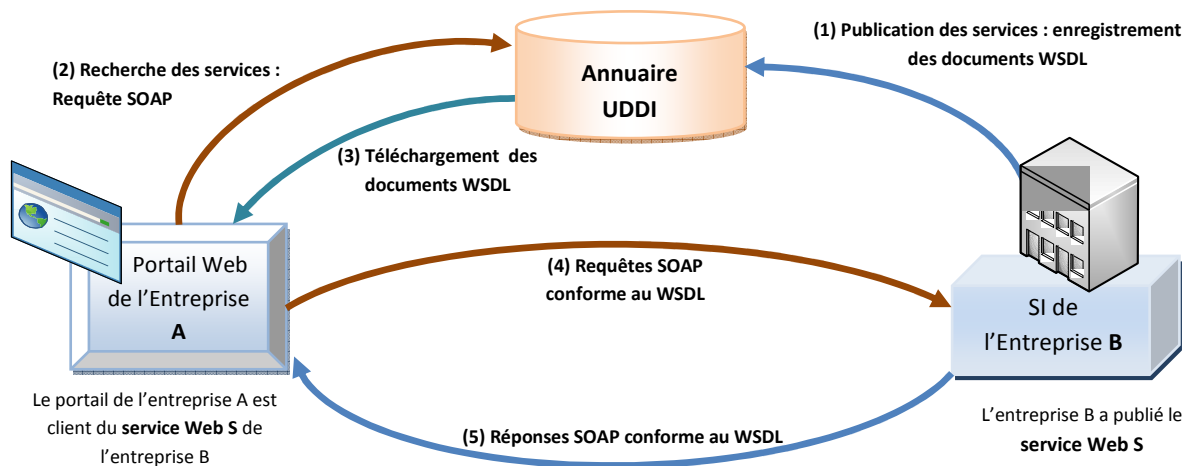


Figure A.3 : Principe de fonctionnement de l'annuaire UDDI [Chauvet, 02]

Comme le montre la figure A.3, la mise en œuvre d'un service Web, en se basant sur l'annuaire UDDI, passe par cinq étapes : (1) la publication des services où le fournisseur publie (enregistre) son service Web auprès d'un distributeur (sur un annuaire UDDI). Cette opération se fait en envoyant directement à l'annuaire un message UDDI (encapsulé dans une enveloppe SOAP) via un protocole de transport. Les informations fournies regroupent la localisation du service, la méthode d'invocation (et les paramètres associés) ainsi que le format de réponse. Toutes ces informations seront formalisées ensuite à l'aide de WSDL. (2) la recherche : le client effectue une recherche de service Web auprès du distributeur. Cette recherche se fait en envoyant un message UDDI (requête) encapsulé dans une enveloppe SOAP via un protocole de transport. (3) récupération le résultat de la recherche : le client reçoit en retour (via un protocole de transport) une réponse de l'annuaire. Cette réponse est un message WSDL encapsulé dans une enveloppe SOAP. (4) Invocation du service : à l'aide de la réponse reçue, le client va pouvoir accéder directement au service Web chez le fournisseur. La demande de service s'effectue à l'aide d'un message SOAP via un protocole de transport. (5) réponse à l'invocation : le client reçoit une réponse (via un protocole de transport) du service Web sous la forme d'un message SOAP. Il ne reste alors qu'à exploiter cette réponse.

ANNEXE B : Description sommaire du langage BPEL (Business Process Execution Language)

Introduction

Le BPEL (Business Process Execution Language) est un langage de programmation destiné à l'exécution des procédures d'entreprise. Le BPEL est issu des langages WSLF (Web Services Flow Language) et XLANG, et est dérivé du XML.

BPEL est un langage de modélisation/orchestration des processus qui se base sur XML et qui soutient la technologie des services web, tels que SOAP, WSDL, UDDI, WS-Reliable Messaging, WS-Addressing, WS-Coordination et WS-Transaction. Il permet ainsi l'intégration des processus métier avec des applications de services web automatisés.

La spécification initiale BPEL4WS (Business Process Execution Language for Web Services) date de 2002 ; elle est devenue WS-BPEL.

WS-BPEL ou BPEL dépend particulièrement des spécifications basées XML suivantes : WSDL - Web Services Description Language (<http://www.w3.org/TR/wsdl>), XPath expression language (<http://www.w3.org/TR/xpath>), Schema XML (<http://www.w3.org/2001/XMLSchema>).

Eléments d'un fichier BPEL

Le fichier BPEL définit le processus, ou l'enchaînement et la logique des actions qui seront exécutées par le moteur d'orchestration. La structure du fichier BPEL est la même que celle du processus. Ce fichier est véritablement le code source de l'application que constitue le processus, le moteur d'orchestration agissant comme une machine virtuelle capable d'exécuter le code BPEL.

Dans ce qui suit, nous présentons les éléments importants constituant un fichier BPEL.

La balise <process>

La balise <process> est l'élément racine du fichier BPEL. C'est à l'intérieur de cette balise que se retrouvera la description complète du processus. Grâce à l'attribut *name*, on peut donner un nom au processus. La figure B.2 montre un exemple de la balise *process*.

```
<process
  name="processName"
  xmlns="http://docs.oasis-open.org/wsbpel/2.0/process/executable"
  targetNamespace="http://example.com"
  xmlns:tns="http://example.com"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  [...]
</process>
```

Figure B.1 : Exemple de la balise <process>

La balise <import>

Cette balise permet d'importer un fichier WSDL au sein du programme BPEL. La figure B.2 montre un exemple:

```
<import namespace="http://example.com" location="fichier.wsdl"
importType="http://schemas.xmlsoap.org/wsdl/" />
```

Figure B.2 : Exemple de la balise <import>

La balise <partnerLinks>

Cette balise permet de lier des actions définies dans le fichier WSDL (via *partnerLinkType*) au processus BPEL. L'attribut *myRole* ou *partnerRole* définit si c'est une action qui appelle le processus ou si c'est une action appelée par le processus.

La figure B.3 illustre une section du BPEL relative *partnerLink*.

```
<partnerLinks>
  <partnerLink name="PartnerLink1" partnerLinkType="tns:examplePL" myRole="exampleRole" />
</partnerLinks>
```

Figure B.3 : Exemple de la balise < partnerLinks>

La balise <variables>

Cette balise permet de définir les variables utilisées par le processus.

```
<variables>
  <variable name="var" messageType="tns:exampleMessage" />
</variables>
```

Figure B.4 : Exemple de la balise <variables>

La balise <sequence>

Cette balise va contenir des actions ou de la structure directement liée à l'exécution du processus.

```
<sequence name="Main">
  [Actions]
</sequence>
```

Figure B.5: Exemple de la balise <sequence>

La balise <receive>

Cette balise permet de recevoir un signal de l'extérieur du processus. Cela permet d'instancier un processus par exemple, ou plus généralement d'attendre qu'un évènement se termine avant de continuer le processus.

```
<receive name="Receive1" createInstance="yes" PartnerLink="PartnerLink1" operation="exampleOperation"
portType="examplePortType" variable="var1In"/>
```

Figure B.6: Exemple de la balise <receive>

La balise <reply>

Cette balise permet de renvoyer une réponse à un partnerLink qui en attend une. D'abord le receive, puis, après le traitement, le reply.

```
<reply name="Reply1" PartnerLink="PartnerLink1" operation="exampleOperation" portType="examplePortType"
variable="var1Out"/>
```

Figure B.7: Exemple de la balise <reply>

La balise <invoke>

Cette balise permet d'appeler un service Web. Elle utilise un *partnerLink* « sortant » et peut ou non recevoir une réponse.

```
<invoke name="invoke1" PartnerLink="PartnerLink1" operation="exampleOperation" portType="examplePortType"
inputVariable="varIn" outputVariable="varOut"/>
```

Figure B.8: Exemple de la balise <invoke>

La balise <forEach>

La balise *forEach* permet d'effectuer une boucle. On déclare un compteur (variable qui sera incrémentée à chaque itération), une valeur de départ et une valeur finale pour ce compteur. La boucle peut s'exécuter en mode parallèle.

```
<forEach name="superBoucle" parallel="yes" counterName="index">
  <startCounterValue>1</startCounterValue>
  <finalCounterValue>$suneVariable</finalCounterValue>
  [Instructions]...
</forEach>
```

Figure B.9: Exemple de la balise <forEach>

Dans cet exemple, si *\$suneVariable* est égale à 3, la boucle sera exécutée 3 fois. Le compteur peut être appelé comme n'importe quelle variable (ici *\$index*).

ANNEXE C : Description sommaire de la notation BPMN (*Business Process Model notation*)

1. Définition et concepts

BPMN (Business Process Model notation) est une norme de notation pour la modélisation de processus. BPMN est soutenu par l'OMG/BPMI (*Object Management Group/ Business Process Management Initiative*) depuis leur fusion en 2005. Son objectif est de fournir un cadre permettant de décrire un processus d'une manière commune à tous les utilisateurs et ce, indépendamment de l'outil utilisé.

La notation définit deux concepts pour l'organisation des processus : orchestration et la chorégraphie. Les orchestrations sont internes, elles définissent ce qui se passe à l'intérieur d'un groupement. Les chorégraphies sont interprocessus, elles définissent les communications entre groupements (entre processus). Les échanges internes à un processus BPMN sont effectués au travers d'enchaînements d'activités. Ils signifient qu'un enchaînement entre les différentes tâches du processus est effectué. Les échanges entre les processus sont eux représentés par des messages et répondent au nom de chorégraphie [Stephen, 04].

2. Types d'Objets de la norme

La notation BPMN définit trois types d'objets comme illustré dans le tableau C.1 ci-dessous [Ferrari, 07]:






Notation	Description
	Une tâche : est un élément indivisible. Elle représente une action. Chaque tâche a un début et une fin et donc une tâche ne peut débuter que si la tâche précédente est terminée.
	Les branchements : Il sert à représenter la condition de routage entre le(s) flux en entrée et le(s) flux en sortie. On distingue plusieurs types de branchements : séquentiels, parallèles, conditionnels et ceux de synchronisation.
	Les évènements : servent à identifier un état particulier dans le processus. Ils n'effectuent aucune tâche. Dans BPMN 1.0 il existe 3 types d'évènements : début et fin qui représentent le squelette du processus et les évènements intermédiaires qui peuvent être au milieu d'un flux pour signaler l'attente d'un type d'évènement ou rattaché à une tâche pour faire apparaître une exception au traitement de la tâche.

Tableau C.1 : Les objets de la norme BPMN

Dans ce qui suit, on donnera les différentes notations utilisées pour chacune des catégories définies précédemment.

3. Tâches

Le tableau C.2 suivant illustre les différents types de tâches définies par la notation BPMN :

Notation	Description
	Sous processus qui est une activité décomposée. Elle est représentée par une tâche avec un petit + permettant d'accéder au détail. C'est la manière dans BPMN de définir des abstractions et de choisir la granularité de l'information représentée.
	Tâche d'envoi : activité qui permet d'effectuer l'envoi d'un message.



	Tâche manuelle: activité effectuée par un opérateur humain avec une assistance de l'application logicielle.
	Tâche de réception : activité qui permet d'effectuer la réception d'un message.

Tableau C.2 : Types de Tâches

4. Les objets de connexion

Il existe 3 types de connexions possibles. Le tableau C.3 suivant illustre ces types :

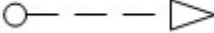
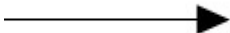

Notation	Description
	Les messages, représentés par un trait en pointillés, servent à décrire les échanges entre processus. Les messages représentent un lien entre des processus. Ils ne sont qu'un signal entre 2 processus et ne déclenchent rien de particulier.
	Les enchainements d'activité, trait plein, représentent le flux entre 2 tâches.
	Les associations servent comme support de rattachement entre une tâche et un objet de données ou avec une activité de compensation. Elles sont représentées par un trait en pointillé.

Tableau C.3 : Les objets de Connexion

5. Les évènements

La notation BPMN sépare en deux (02) catégories les évènements :

- les évènements d'attente symbolisés par un fond blanc,
- les évènements de lancement sur fond noir.

Les premiers donnent le type d'évènement que le processus est en mesure de traiter alors que le deuxième définit le type d'évènement en sortie du processus. Le tableau C.4 suivant illustre quelques types d'évènements utilisés dans la modélisation des processus :







Notation	Description
	Evènements message : permettent de signaler l'arrivée ou l'attente d'un message.
	Evènement Début : indique le point de départ et ne symbolise aucune tâche. Il peut recevoir un évènement de départ.
	Evènement Fin : indique la fin du processus et ne symbolise aucune tâche. Il peut envoyer un évènement de sortie.
	Evènement de compensation : signale qu'une tâche doit être annulée. Ce type d'évènement est utilisé dans les processus transactionnels et permet de définir la tâche qui doit être utilisée pour compenser une tâche particulière.
	Evènement d'annulation : signale la fin du processus et annule les transactions en cours.
	Evènement Temporisation (Timer) : évènement intermédiaire symbolisé par un double trait. C'est une partie importante de BPMN. Il peut être utilisé de 2 manières : au milieu d'un flux pour signaler l'attente d'un type d'évènement ou rattaché à une tâche pour faire apparaître une exception au traitement de la tâche.

Tableau C.4 : Types d'évènements

6. Les branchements

Le branchement est un objet essentiel dans la norme BPMN. Il sert à représenter la condition de routage entre le(s) flux en entrée et le(s) flux en sortie. Le branchement n'est pas une tâche et n'effectue aucune action.

Lorsque le losange est vide, chaque sortie est une alternative et il n'y a pas de différenciation entre les sorties. Le losange vide est utilisé lorsque le niveau d'abstraction du modèle est élevé et que l'on ne désire pas compliquer une vue ou bien lorsque la règle de traitement n'est pas connue.

Il est donc possible de définir un comportement de traitement précis.

Les branchements sont autant utilisés pour diviser un flux en plusieurs flux que pour réunir plusieurs flux en un seul.

Les différents branchements possibles dans BPMN 1.1 sont :

- les branchements exclusifs,
- les branchements parallèles,
- les branchements conditionnels,
- les branchements de synchronisation.

La figure C.1 suivante illustre ces différentes catégories de branchement :

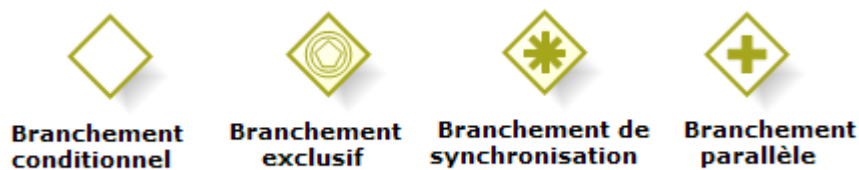


Figure C.1 : Types de Branchement

Exemple d'achat de produits

Pour terminer, nous allons illustrer à travers un exemple simple la modélisation, par la notation BPMN, d'un service d'achat d'un produit (pièce de rechange par exemple). La demande d'achat est effectuée dans le processus appelé « Dept. 1 » et le traitement de la demande ainsi que l'opération d'achat sont effectuées à travers le processus appelé « Dept. 2 ». La figure C.2 suivante illustre cette modélisation en utilisant la notation BPMN :

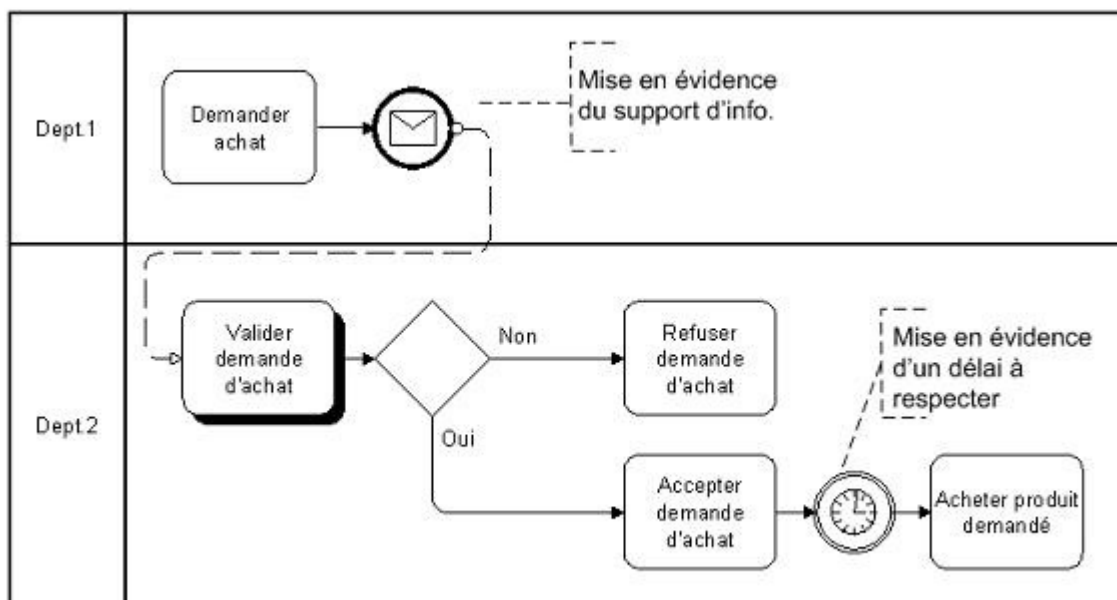


Figure C.2 : Exemple de modélisation du processus « Achat de produits »

ANNEXE D : Description sommaire des approches EA, OOAD et BPM

1. EA (Enterprise Architecture) : regroupe l'ensemble d'outils de conception qui permettent de définir la structure et les opérations d'une organisation. Cette approche définit trois types de vue comme illustrée dans la figure D.1 suivante [Platt, 02] :

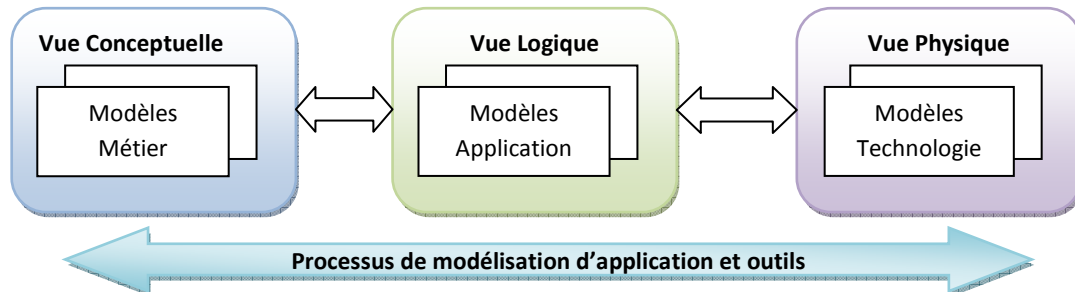


Figure D.1 : Vues de l'Architecture d'Entreprise (EA)

- Vue conceptuelle : définit les besoins métier et génère un modèle métier. On utilise dans ce niveau les différentes techniques de modélisation conceptuelle, comme l'analyse par cas d'utilisation (Use Cases), diagramme d'activités et modélisation des processus.
- Vue logique : représente les modèles d'application, ces derniers déterminent la façon de correspondre les objectifs avec les besoins métier.
- Vue physique : chaque élément du modèle application doit être transformé en un élément d'une technologie réelle. Ces vues réalisent donc le modèle d'Implémentation.

2. OOAD (Object Oriented Analysis and Design) : initialement basée sur l'utilisation d'objets en tant qu'abstractions du monde réel, l'approche objet a pour but une modélisation des propriétés statiques et dynamiques de l'environnement dans lequel sont définis les besoins, appelé le « domaine du problème » [Muller et Gaertner, 02].

Les principes fondamentaux de cette approche sont :

- Encapsulation : signifie que l'objet informatique forme une boîte noire qui regroupe son état (propriétés) ainsi que ses opérations (comportement).
- Informations cachées : les objets bien structurés ont des interfaces qui permettent de cacher au monde extérieur tous les détails sur l'implémentation et les mécanismes internes des objets.
- Classes et instances : la classe décrit le domaine de définition d'un ensemble d'objets, et chaque objet appartient à une classe. Les objets informatique sont construit à partir de la classe par un processus appelé « instantiation ».
- Messagerie : les objets se communiquent à travers les messages.
- Héritage : utilisé pour dériver une nouvelle classe à partir d'une superclasse.
- Polymorphisme : désigne la possibilité de déclencher des opérations différentes en réponse à un même message.

L'approche orientée services support le cycle de vie complet pour l'analyse, conception et développement. En effet, OOAD permet d'identifier tous les objets ainsi que les hiérarchies de leurs

classes pour une éventuelle implémentation. OO (Oriented Object) Développement permet un développement incrémental des applications en implémentant à la fois un scénario métier. OO Runtime Environment, exemple courant est la machine virtuelle du Java (JVM) qui exécute le code et qui fournit des services supplémentaires aux applications telles que « ramasse miettes » (*Garbage Collector*) dans le monde Java.

3. BPM (Business Process Modeling) : c'est une discipline qui définit plusieurs modèles et notations qui permettent aux entreprises de spécifier étape par étape leurs processus métier ainsi que leurs gestions en incluant dans ces processus les différents acteurs (système informatique, personnes) et leurs interactions au sein du processus métier.

Plusieurs approches ont été proposées dans ce domaine par différents éditeurs, et tous proposent généralement des notations additionnelles par rapport à celles définies dans UML.

4. Synthèse des approches

La méthode Orientée Objets (OOAD) offre un cadre méthodologique très intéressant pour SOA, cependant, elle couvre seulement le niveau d'abstraction le plus bas dans l'architecture à travers les classes et objets. De plus, OOAD est généralement pilotée par un modèle de cas d'utilisation qui se focalise sur un domaine du problème, or dans des projets de développement pour les grandes entreprises, les concepteurs sont souvent brouillés par le nombre important de cas d'utilisations.

Quand au Framework EA, il ajoute au dessus des solutions architecture un plan d'urbanisation du système, mais n'offre pas la possibilité d'une part d'identifier les services et d'une autre part la réutilisation et longévité des services.

En fin pour les approches BPM, elles permettent d'avoir une vue de bout-en-bout pour les unités fonctionnelles du métier, mais ne couvrent pas les domaines architecture et implémentation du cycle de développement.

Il est important à souligner qu'aucunes des approches citées ne montre comment valoriser et réutiliser les systèmes existants (applications légataires).

La figure D.2 suivante illustre la position des trois approches (OOAD, EA et BPM) dans cycle de développement ainsi que leurs niveaux d'abstraction (domaines) [Zimmerman et al, 04].

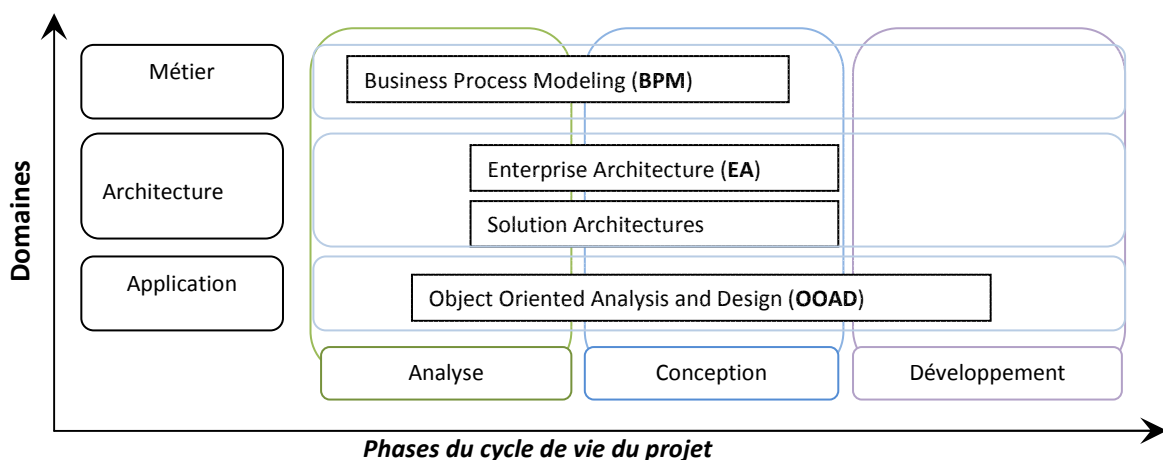


Figure D.2 : Position des approches OOAD, EA et BPM

ANNEXE E : Description sommaire des formalismes CTT (*ConcurTaskTrees*) et SADT (*Structured Analysis and Design Technic*)

I. Le formalisme CTT (*ConcurTaskTrees*)

I.1. Les tâches du formalisme CTT

Une tâche définit comment l'utilisateur atteint un but dans un domaine applicatif particulier, Le but étant l'obtention d'un état désiré du système

– Elle est donnée par les éléments suivants : (1) un identifiant, (2) une catégorie (qui va exécuter la tâche ?), (3) un type d'action (selon la catégorie), (4) une fréquence d'utilisation (faible, moyenne, haute), (5) une plate-forme cible (Desktop, PDA, GSM, etc...), (6) les options (itérative, optionnelle, coopérative), (7) les pré-conditions, (8) la liste d'objets nécessaires à sa réalisation, (9) la liste des actions d'entrée ou de sortie des objets et enfin (10) les durées d'exécution d'une tâche (min, max, moyenne).

Les catégories de tâches

CTT inclut quatre catégories de tâches qui aident à construire l'arbre des tâches :

1. *les tâches utilisateurs* : activités cognitives ou physiques de l'utilisateur indépendantes de toute interaction sur le système ;
2. *les tâches application* : tâches effectuées par le système pour rendre compte de l'état du système ;
3. *les tâches interaction* : tâches réalisées par les interactions de l'utilisateur sur le système
4. *les tâches abstraites* : tâches raffinées par les tâches précédentes. Il s'agit de tâches de haut niveau sans savoir exactement la portée de son action.

Le tableau E.1 suivant dresse un ensemble illustratif de graphiques liés aux tâches CTT.





Icône Graphique	Nom	Description
	Utilisateur	C'est une tâche entièrement réalisée par l'utilisateur. • ex. choisir un hôtel dans une liste
	Application	C'est une tâche entièrement réalisée par l'application. • ex. compiler un programme
	Interactive	C'est une tâche itérative réalisée par le système et l'utilisateur.
	Abstraite	C'est une tâche qui décrit une interaction complexe, pas dans les trois tâches précédentes.

Tableau E.1 : Les graphiques du formalisme CTT

Opérateurs temporels

En outre les modèles CTT définissent un certain nombre d'opérateurs temporels permettant la définition des relations temporelles entre deux tâches (séquence, choix, parallélisme, interruption, synchronisation, etc.).

Dans ce qui suit, nous présentons quelques relations temporelles définies dans le formalisme CTT.

- **T1 ||| T2 imbrication** : actions en parallèle
- **T1 |[] T2 synchronisation** : échange d'information
- **T1 >> T2 activation** : fin de T1 déclenche T2
- **T1 |= T2 ordre quelconque** : T1>>T2 ou T2>>T1
- **T1 []>> T2 activation avec passage d'information**

- **T1 [] T2 choix** : soit T1 soit T2
- **T1 [> T2 désactivation** : dès qu'une action de T2 se produit, T1 est désactivée
- **T1 > T2 interruption** : T2 peut interrompre T1, T1 peut reprendre à la fin de T2.
- **T1*** signifie une itération
- **T1(n) itération finie** : définit un nombre d'itérations spécifié.
- **[T1] option** : signifie que la tâche T1 n'est pas obligatoirement faite

I. La technique SADT (*Structured Analysis and Design Technic*)

L'acronyme S.A.D.T signifie : *Structured Analysis and Design Technic*. Cette méthode a été mise au point par la société *Softtech* aux Etats Unis. La méthode SADT est une méthode d'analyse par niveaux successifs d'approche descriptive d'un ensemble quel qu'il soit. On peut appliquer le SADT à la gestion d'une entreprise tout comme à un système automatisé. Le SADT est avant tout un langage de communication. Cette communication se fait à différents niveaux. Au niveau de l'élaboration du projet tout d'abord en permettant par son formalisme à chacun de participer, ensuite lors d'explications à des intervenants extérieurs son formalisme permet à chacun d'appréhender le système automatisé.

I.1. Les actigrammes et les datagrammes

Dans une analyse SADT, on peut modéliser deux types d'analyse. L'analyse par des actigrammes (boîtes d'action) et l'analyse par des datagrammes (boîtes de donnée).

Sur des actigrammes, les actions sont reliées entre elles par des flux de données alors que les datagrammes se sont les données qui sont reliées entre-elles par des flux d'activité.

La figure E.1 suivante illustre les graphiques associés aux actigrammes et datagrammes.

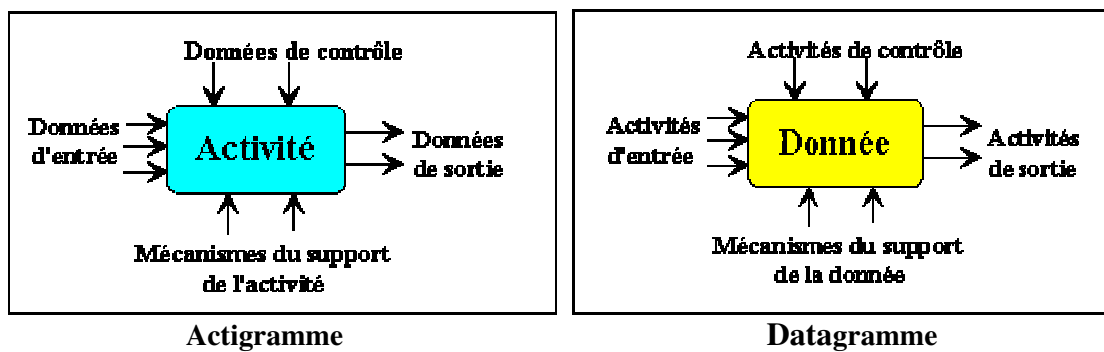


Figure E.1 : Les Actigrammes et les Datagrammes

ANNEXE F : Description sommaire de WSRP (*Web Services for Remote Portlet*)

Nous présentons dans cette partie une introduction sommaire de la spécification WSRP et ses éléments clé. Pour cela, nous introduisons les notions de portail Web et de portlets qui représentent les fondements du développement des services orientés présentation. Par la suite nous décrivons les principes de base relative à la spécification WSRP.

Les portails Web

Les portails proposent un point d'accès unique à un ensemble de contenus et d'application à caractères analytique ou collaboratif en facilitant l'accès à des données structurées comme les base de données ou non structurées comme les documents texte indépendamment de leur format et de leur localisation. En effet c'est le portail qui ce charge de l'agrégation de l'information.

Les portlets

Les portlets sont des programmes destinés à produire du code balisé HTML, XML afin d'afficher des informations dans les portails.

Ce sont des modules intégrés à un portail d'entreprise, qui permet à l'utilisateur de disposer, dans la même fenêtre, d'un accès centralisé et convivial à différentes ressources, données, applications, sites Web, etc., de modifier l'interface du portail selon ses besoins et de personnaliser ainsi son environnement de travail.

Le WSRP

Le WSRP : *Web Services for Remote Portlet* est une standardisation de l'OASIS : Organisation for the Advancement of Structured Information Standards. L'intégration de contenu et/ou d'interface applicative dans un point d'accès unique pour l'utilisateur final a nécessité des lourds développements spécifiques pour les fournisseurs de portail. Ainsi, le WSRP devrait permettre à un designer ou un administrateur de choisir parmi un large choix de contenu et/ou d'interface applicative et de les intégrer en quelques instants et simplement dans le portail [Schaeck et Thompson, 03]. Au même, titre que les web services l'aspect réutilisation est une base de l'objectif de la standardisation WSRP.

La couche relative à WSRP se trouve entre d'une part les couches relatives aux technologies des protails et d'autres des couches relatives aux services Web (figure F.1). Ainsi, le WSRP représente une intersection entre les portails et les services Web.

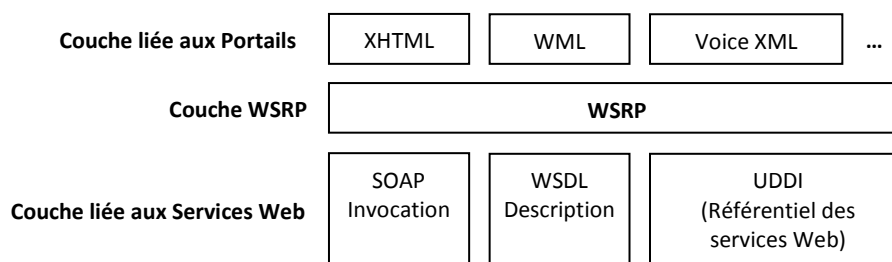


Figure F.1 : Couche WSRP par rapport aux portails et les services Web.

Les intervenants dans le WSRP :

Le WSRP définit cinq éléments intervenant dans l'exécution des services orientés présentation :

- **Portlet** : Les portlets sont hébergées par le Producer des Web Services et génère le code.
- **Producer** : c'est le conteneur des portlets. Il fournit les 4 interfaces Web Services pour les portlets. Ces interfaces seront décrites par la suite. Il peut également gérer l'enregistrement du Consumer. Par exemple, il peut exiger que le Consumer soit enregistré pour que ce dernier puisse interagir complètement avec les portlets. L'enregistrement représente la relation qui existe entre le Consumer et le Producer.
- **Portlet management** : C'est une portlet spécial avec un PortletHandle. Le Consumer utilise des Portlethandles durant la communication pour adresser et interagir avec les portlets via le Producer. Les portlets fournies par le Producer sont disponibles pour tous les Consumers et sont appelées : « Producer Offered Portlets ». Les portlets sont pré-configurées et ne sont pas modifiables par les Consumers. Cependant, le Producer peut fournir une interface de Portlet Management qui permet au Consumer de cloner les portlet de Producer et de les personnaliser.
- **Consumer** : C'est un système intermédiaire qui communique en Web Service au profit de ces utilisateurs. C'est lui qui est chargé de rassembler et d'agrèger les codes balisés fournis par les portlets pour les présenter à l'utilisateur final. En général, c'est un portail.
- **End-User** : Comme son nom l'indique c'est l'utilisateur final, c'est celui qui a besoin du service. C'est à lui qu'est présenté le résultat agrégé du Consumer.

Les interfaces de WSRP :

Le WSRP définit quatre types d'interface avec les portlets :

- 1) *Service Description* : Un producer peut être découvert par des mécanismes comme UDDI, ce mécanisme fournit également les informations concernant les possibilités du service (par exemple en termes de qualité de service).
- 2) *Markup* : Cette interface contient les opérations pour demander la génération des Markup (fragment d'interface généralement codés en HTML) et les méthodes nécessaires pour interagir avec ce Markup.
- 3) *Registration* : Cette interface contient toutes les opérations permettant d'enregistrer la description des services WSRP au niveau du référentiel UDDI.
- 4) *Portlet Management* : Cette interface offre des mécanismes pour la gestion et la sécurité lors de l'exécution des portlets distantes.

ANNEXE G : Tableaux représentant les différents acteurs métier de la maintenance avec leurs catégories, besoins, tâches et leurs activités.

Acteur	Chef du bureau de méthodes (BM)
Catégorie	Personnel du bureau de méthodes
Besoins	Etre informé des pannes mécaniques, électriques et automatiques, qui surgissent sur l'unité et les besoins en termes de pièces de rechange et de la maintenance. Moyens de communication.
Tâches	Assurer le fonctionnement continu de l'unité du coté matériel et la coordination au sein du bureau de méthodes.
Activités	Consulter les rapports des pannes et les besoins en matière de maintenance, ainsi que faire des demandes d'approvisionnement (DF) en pièces de rechange.

Acteur	Méthodiste mécanicien
Catégorie	Personnel du bureau de méthodes
Besoins	Etre informé des pannes mécaniques, qui surgissent sur l'unité et les besoins en termes de pièces de rechange et de la maintenance. Moyens de communication, et de la documentation technique relative à la maintenance mécanique.
Tâches	Assurer la maintenance dans le domaine mécanique.
Activités	Assister les opérateurs de la maintenance et noter leurs besoins dans le domaine mécanique. Faire des rapports sur l'état des machines et des besoins envers le chef du bureau de méthodes.

Acteur	Méthodiste électricien
Catégorie	Personnel du bureau de méthodes
Besoins	Etre informé des pannes électriques, qui surgissent sur l'unité et les besoins en termes de pièces de rechange et de la maintenance. Moyens de communication, et de la documentation technique relative à la maintenance électrique.
Tâches	Assurer la maintenance dans le domaine électrique.
Activités	Assister les opérateurs de la maintenance et noter leurs besoins dans le domaine électrique. Faire des rapports sur l'état des machines et des besoins envers le chef du bureau de méthodes.

Acteur	Méthodiste automaticien
Catégorie	Personnel du bureau de méthodes
Besoins	Etre informé des pannes automatiques, qui surgissent sur l'unité et les besoins en termes de pièces de rechange et de la maintenance. Moyens de communication, et de la documentation technique relative à la maintenance automatique.
Tâches	Assurer la maintenance dans le domaine automatique.
Activités	Assister les opérateurs de la maintenance et noter leurs besoins dans le domaine de l'automatique. Faire des rapports sur l'état des machines et des besoins envers le chef du bureau de méthodes.

Acteur	Planificateur du bureau de méthodes
Catégorie	Personnel du bureau de méthodes
Besoins	Etre informé des demandes d'intervention (DI), changement dans le plan de production et la fin des travaux de maintenance. Moyens de communication.
Tâches	Assurer la planification et la gestion des opérations de la maintenance
Activités	Création et saisie des différents documents relatifs à la gestion de la maintenance. Sauvegarde dans la GMAO les différents résultats des travaux et informer le chef du bureau de méthodes sur l'état d'avancement des travaux liés à la maintenance industrielle.

Acteur	Mécanicien
Catégorie	Opérateur de maintenance
Besoins	Moyens de réparation et de maintenance, documentation relative à la maintenance en mécanique, être sollicité de régler un problème et fournir des informations relatives aux problèmes.
Tâches	Assurer la maintenance dans le domaine mécanique.
Activités	Recherche la cause de l'arrêt ou le dysfonctionnement mécanique et régler le problème.

Acteur	Electricien
Catégorie	Opérateur de maintenance
Besoins	Moyens de réparation et de maintenance, documentation relative à la maintenance en électrique, être sollicité de régler un problème et fournir des informations relatives aux problèmes.
Tâches	Assurer la maintenance dans le domaine mécanique.
Activités	Recherche la cause de l'arrêt ou le dysfonctionnement mécanique et régler le problème.

Acteur	Mécanicien
Catégorie	Opérateur de maintenance
Besoins	Moyens de réparation et de maintenance, documentation relative à la maintenance en automatique, être sollicité de régler un problème et fournir des informations relatives aux problèmes.
Tâches	Assurer la maintenance dans le domaine mécanique.
Activités	Recherche la cause de l'arrêt ou le dysfonctionnement mécanique et régler le problème.

Acteur	Aide Mécanicien
Catégorie	Opérateur de maintenance
Besoins	Moyens de réparation et de maintenance, documentation relative à la maintenance en mécanique, être sollicité de régler un problème et fournir des informations relatives aux problèmes.
Tâches	Assurer la maintenance dans le domaine mécanique.
Activités	Aider les opérateurs de la maintenance au niveau de la mécanique.

ANNEXE H : Extraits des fichiers WSDL relatifs au service Web « Réservation des pièces de rechange (PdR) »

Fichier WSDL : DemandeRéservPdR.wsdl

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions name="demReservRessWSDL">
<types>
<xsd:schema targetNamespace="http://j2ee.netbeans.org/wsdl/demReservRessWSDL">
<xsd:import namespace="http://xml.netbeans.org/schema/SoaMiXmlSchema"
schemaLocation="SoaMiXmlSchema.xsd"/>
</xsd:schema>
</types>
<message name="demReservPdROperationRequest">
<part name="requete" element="ns:criterElement"/>
</message>
<message name="demReservPdROperationResponse">
<part name="reponse" element="ns:listPdRElement"/>
</message>
<message name="demReservPersOperationRequest">
<part name="requete" element="ns:criterElement"/>
</message>
<message name="demReservPersOperationResponse">
<part name="reponse" element="ns:listPersElement"/>
</message>
<portType name="DemandeRéservPdRType">
<operation name="demReservPdROperation">
<input name="input1" message="tns:demReservPdROperationRequest"/>
<output name="output1" message="tns:demReservPdROperationResponse"/>
</operation>
<operation name="demReservPersOperation">
<input name="input2" message="tns:demReservPersOperationRequest"/>
<output name="output2" message="tns:demReservPersOperationResponse"/>
</operation>
</portType>
<binding name="demReservRessWSDLBinding" type="tns:DemandeRéservPdRType">
<soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
<operation name="demReservPdROperation">
<soap:operation/>
<input name="input1">
<soap:body use="literal"/>
</input>
<output name="output1">
<soap:body use="literal"/>
</output>
</operation>
<operation name="demReservPersOperation">
<soap:operation/>
<input name="input2"/>
<output name="output2"/>
</operation>
</binding>
<service name="demReservRessWSDLService">
<port name="DemandeRéservPdR" binding="tns:demReservRessWSDLBinding">
<soap:address location="http://localhost:8080/demReservRessWSDLService/DemandeRéservPdR"/>
</port>
</service>
<plnk:partnerLinkType name="demReservRessWSDL">
<plnk:role name="DemandeRéservPdRTypeRole" portType="tns:DemandeRéservPdRType"/>
</plnk:partnerLinkType>
</definitions>
```

Fichier WSDL : LecturePièceRechange.wsdl

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions name="LecturePièceRechangeWSDL">
xmlns:tns="http://j2ee.netbeans.org/wsdl/LecturePièceRechangeWSDL"
xmlns:ns="http://xml.netbeans.org/schema/PartXmlSchema"
<types>
<xsd:import namespace="http://xml.netbeans.org/schema/PartXmlSchema"
schemaLocation="PartXmlSchema.xsd"/>
</xsd:schema>
```

```

</types>
<message name="getListPartOperationRequest"/>
<message name="getListPartOperationResponse">
  <part name="liste" element="ns:listPdRElement"/>
</message>
<message name="rechPartOperationRequest">
  <part name="critere" element="ns:criterElement"/>
</message>
<message name="rechPartOperationResponse">
  <part name="liste" element="ns:listPdRElement"/>
</message>
<message name="getPartOperationRequest">
  <part name="code" element="ns:champElement"/>
</message>
<message name="getPartOperationResponse">
  <part name="part" element="ns:pieceRechangElement"/>
</message>
<portType name="LecturePièceRechangeWSDLPortType">
  <operation name="getListPartOperation">
    <input name="input1" message="tns:getListPartOperationRequest"/>
    <output name="output1" message="tns:getListPartOperationResponse"/>
  </operation>
  <operation name="rechPartOperation">
    <input name="input2" message="tns:rechPartOperationRequest"/>
    <output name="output2" message="tns:rechPartOperationResponse"/>
  </operation>
  <operation name="getPartOperation">
    <input name="input3" message="tns:getPartOperationRequest"/>
    <output name="output3" message="tns:getPartOperationResponse"/>
  </operation>
</portType>
<binding name="LecturePièceRechangeWSDLBinding"
type="tns:LecturePièceRechangeWSDLPortType">
  <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
  <operation name="getListPartOperation">
    <soap:operation/>
    <input name="input1">
      <soap:body use="literal"/>
    </input>
    <output name="output1">
      <soap:body use="literal"/>
    </output>
  </operation>
  <operation name="rechPartOperation">
    <soap:operation/>
    <input name="input2">
      <soap:body use="literal"/>
    </input>
    <output name="output2">
      <soap:body use="literal"/>
    </output>
  </operation>
  <operation name="getPartOperation">
    <soap:operation/>
    <input name="input3">
      <soap:body use="literal"/>
    </input>
    <output name="output3">
      <soap:body use="literal"/>
    </output>
  </operation>
</binding>
<service name="LecturePièceRechangeWSDLService">
  <port name="LecturePièceRechangeWSDLPort" binding="tns:LecturePièceRechangeWSDLBinding">
<soap:address
location="http://localhost:8080/LecturePièceRechangeWSDLServ/LecturePièceRechangeWSDLPort"/>
  </port>
</service>
  <plnk:partnerLinkType name="LecturePièceRechangeWSDL">
<plnk:role name="LecturePièceRechangeWSDLPortTypeRole"
portType="tns:LecturePièceRechangeWSDLPortType"/>
  </plnk:partnerLinkType>
</definitions>

```

ANNEXE I : Procédure globale pour l'ajout d'un nouveau service métier dédié à la maintenance industrielle dans l'architecture SOA

- D'abord, on définit la nouvelle fonctionnalité en termes des besoins, contraintes et les objectifs attendus par acteurs métier de cette fonction.
- Déterminer le domaine fonctionnel relatif à cette nouvelle fonctionnalité au sein du système d'information global de l'entreprise
- Identifier les cas d'utilisation métier potentiels impliqués dans cette fonction.
- Modéliser les scénarios relatifs à chaque cas d'utilisation en utilisant les différents diagrammes d'interactions et d'activités.
- Utiliser les modèles issus de l'étape précédente (analyse métier) afin d'identifier les nouveaux services métier candidats et leurs opérations. Généralement, les cas d'utilisation métier constituent des bons candidats pour les services métier à exposer. Par ailleurs, les interactions de ces services avec d'autres entités (services, applications ou acteurs humains) peuvent former des opérations pour les services interagissant.
- Associer les services identifiés lors de l'étape précédente à une ou plusieurs catégories de services pour une éventuelle composition ou d'orchestration de services au sein d'un processus métier.
- Analyser les besoins des utilisateurs impliqués dans cette nouvelle fonction en termes des interactions Homme-Machine, de mobilité des acteurs humains et les tâches coopératives pouvant être exécutées par les utilisateurs du système de la maintenance.
- Analyser les applications existantes afin d'extraire ou d'explorer de nouveaux services susceptibles de réaliser une partie des nouveaux objectifs visés par les acteurs métier.
- Spécifier le ou les processus métier relatifs à l'orchestration des nouveaux services identifiés. Cette spécification doit être effectuée à l'aide de la notation BPMN (exemple d'outil : *BizAgi Modeler*).
- Convertir le modèle BPMN vers le langage BPEL pour une éventuelle exécution du processus métier par le moteur BPM. Pour cela, on peut utiliser un outil de translation BPMN vers BPEL tels que : *Intalio Designer, Eclipse Modeler, etc.*
- L'étape suivante est de concevoir les différentes maquettes et prototypes relatifs aux nouveaux services identifiés en se référant aux besoins des utilisateurs en termes d'interface.
- Nous passons par la suite à la spécification des services Web à travers leurs interfaces. Pour cela, nous définissons d'abord le *modèle de classes* (diagramme de classes UML) relatif aux *objets métier* manipulés par les nouveaux services Web identifiés. Pour cela, nous pouvons utiliser un outil de modélisation orienté objet tel que *IBM Rational Rose*.
- Ensuite, nous spécifions les types de données d'échange (TDE) sous forme de *schémas XML* en utilisant des outils de conception de schémas XML, qui sont généralement intégrés dans un IDE (*Integrated Development Environment*) de développement de projets SOA, par exemple l'IDE *Sun Netbeans*. Enfin, nous pouvons développer les interfaces WSDL associés à chaque service Web en utilisant les TDE et en référant au catalogue (description de services en termes d'opérations) des nouveaux services identifiés lors de la phase 2.
- La dernière étape est de concevoir en détail les différents types de composants réalisant (offrant une implémentation) les nouveaux services identifiés. Cette conception doit concerner la structure interne de chaque composant services, ses relations avec les composants existant, les interfaces fournies et celles requises par chaque service métier, etc. Nous obtenons ainsi le *modèle de services*.
- Après avoir conçu les différents types de services, nous terminons notre processus de modélisation par l'allocation des nouveaux services métier selon leur rôle dans une couche spécifique du modèle d'architecture SOA tels que : couche processus, services, etc.

ANNEXE J : Ensemble de prototypes du portail Web dédié à la maintenance



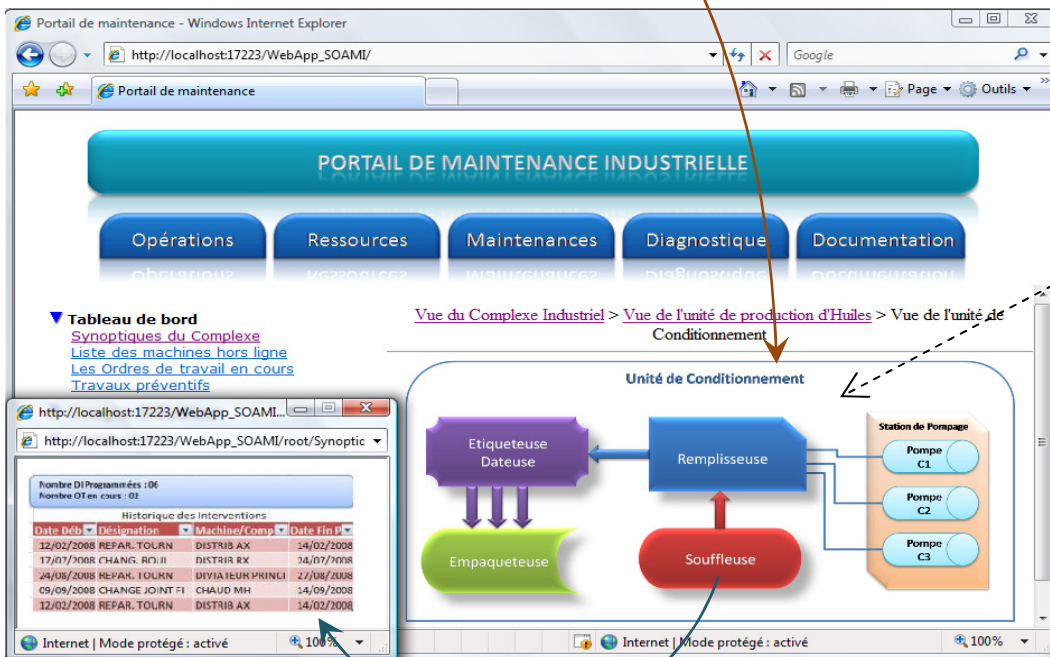
Vue de synoptique relative aux unités de production du Complexe industriel

Suite à un clic sur la zone « unité de production des huiles », une vue sur l'unité s'affiche



Vue de synoptique relative à l'unité de production d'huiles

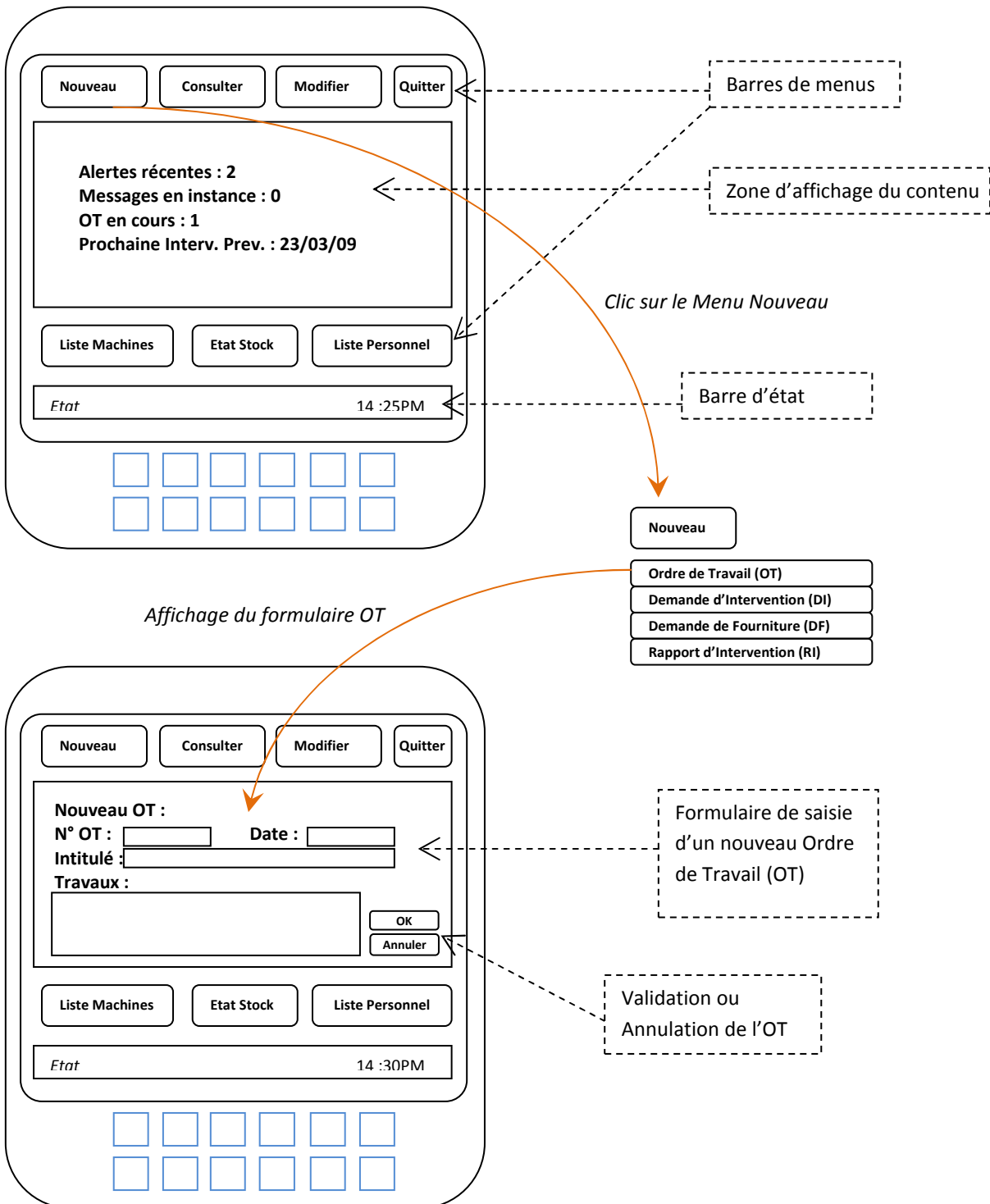
Suite à un clic sur « unité de Conditionnement », une vue interne de l'unité de conditionnement



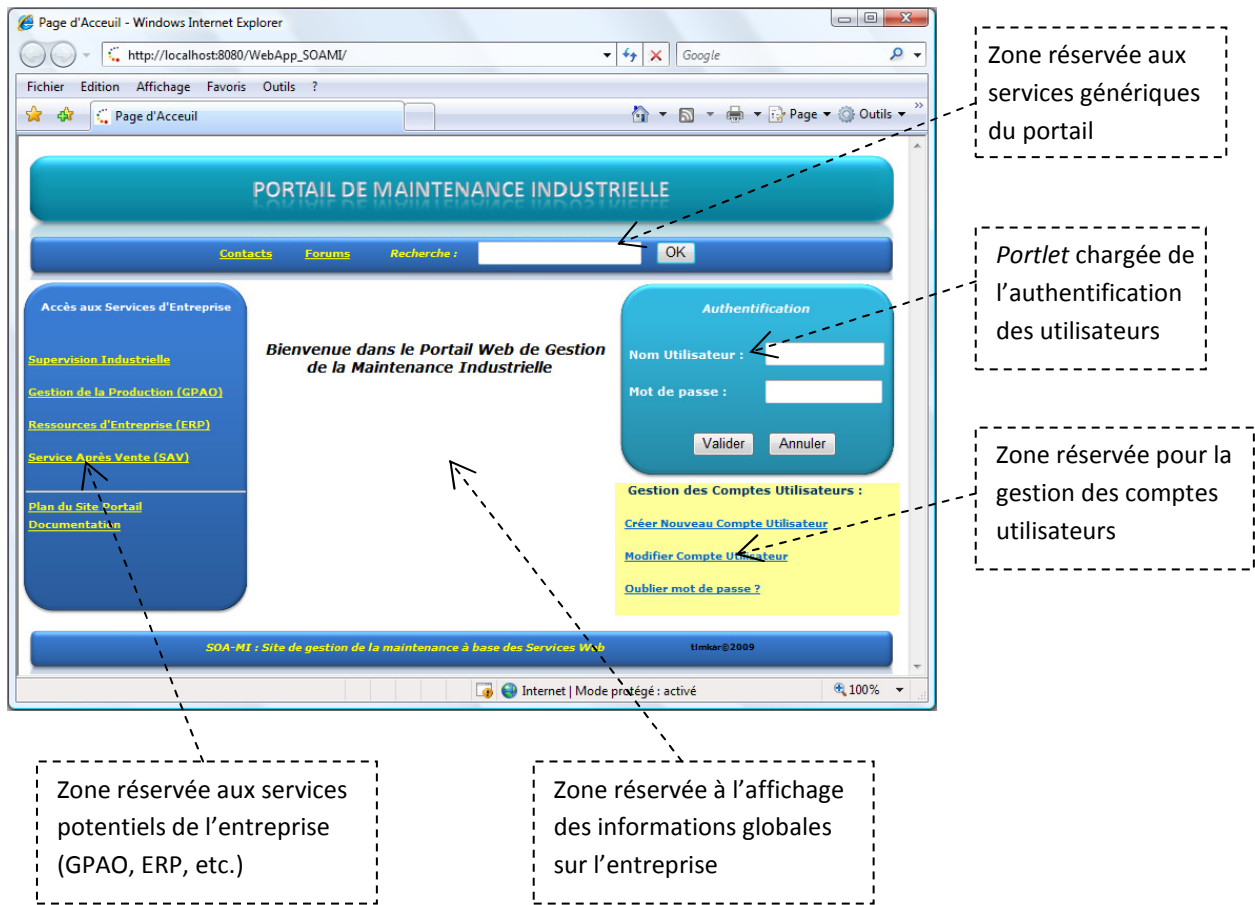
Vue de synoptique relative à l'unité de Conditionnement

Affichage des Informations de la Maintenance sur un type de machine (Travaux en cours, DI programmées, etc.)

Prototypé d'une Interface pour un dispositif mobil (PDA) :



Prototypé de la page d'accueil du portail Web dédié à la gestion de la maintenance



ANNEXE K : Vue globale de la démarche orientée services proposée

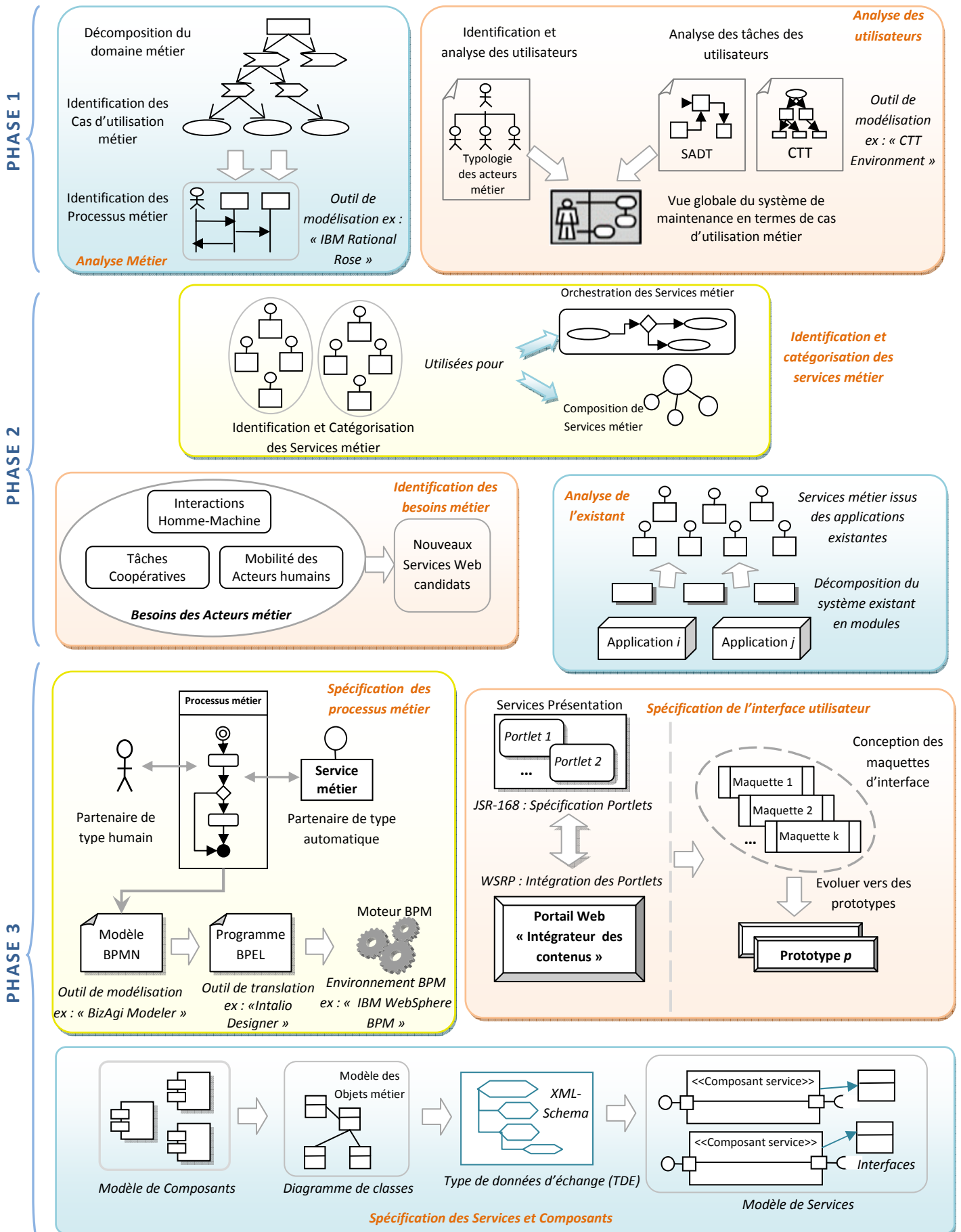


Figure K.1 : Vue synthétique de la démarche orientée services proposée.

Nom	Description
Application composite	Application constituée par assemblage d'un ensemble de composants et utilisant les services métier.
BAM : Business Activity Monitoring	Ensemble d'outils fournissant un tableau de bord, des indicateurs de mesure de la performance métier, des outils de monitoring, de reporting et permet le contrôle du rendu fonctionnel des processus métier ; associé de façon classique à un outil de BPM et considéré à juste titre comme une application à part entière.
BPEL : Business Process Execution Language	Conçu par IBM, BEA et Microsoft, c'est la représentation XML d'un processus exécutable, qui peut être déployée sur n'importe quel moteur de processus métier. L'élément premier d'un processus BPEL est une « activité », qui peut être l'envoi d'un message, la réception d'un message, l'appel d'une opération (envoi d'un message, attente d'une réponse), ou une transformation de données. L'activité est définie par la combinaison de Services Web. BPEL utilise WSDL pour décrire les actions d'un processus.
BPM Business Process Management	Terme générique désignant à la fois la modélisation et la traduction, à l'aide d'un "moteur", des processus modélisés dans la réalité de l'entreprise. A cela s'ajoute, bien évidemment, des outils – indicateurs, tableaux de bord... – de suivi de l'exécution des processus et d'évaluation de leur performance (voir BAM).
BPMN : Business Process Management Notation	Ou <i>Notation pour la Gestion des Processus Métiers</i> . Utilisée pour la modélisation graphique des processus métier (pas l'exécution). Particulièrement pertinente dans le cadre de la modélisation des processus, de leur analyse et de leur simulation. Cette notation est sous le contrôle de l'OMG (Object Management Group).
Composant de service	Composant logiciel <i>fournisseur</i> de service divisé en vue externe et vue interne. La vue externe est la partie <i>service</i> du composant, qui peut être utilisé par d'autres éléments du système, jouant le rôle de <i>consommateur</i> de service. La vue externe (ou spécification de service) est constituée par un ensemble d'opérations de service regroupées en interface, le contrat et toutes les informations liées à son utilisation (voir QoS et SLA).
Composant Entité (ou objet métier)	Composant de service qui donne un accès aux informations liées à un objet métier clé. Les opérations sont de type CRUD (Create, Read, Update, Delete).
Composant Fonction	Composant de service qui prend en charge un processus de traitement, ou une adaptation à une vision métier particulière, sous forme de composition de services.
Composant Présentation	Composant de service dédié à l'interaction entre acteurs humains et le système. Il prend en charge les IHM et la gestion de la session utilisateur.
Composant Processus	Composant de service qui automatise un processus métier.
Composant technique	Composant de service dédié à des fonctions transverses (annuaire, messagerie, éditique) non spécifiques au métier de l'entreprise.
OASIS (www.oasis-open.org)	Organization for the Advancement of Structured Information Standards. Consortium international qui travaille pour la normalisation et la standardisation de formats de fichiers ouverts basés notamment sur XML.
OMG (www.omg.org)	L'OMG (Object Management Group) est une association américaine à but non-lucratif créée en 1989 dont l'objectif est de standardiser et promouvoir le modèle objet sous toutes ses formes. L'OMG est par exemple à l'initiative des normes UML et CORBA ou encore MDA et BPMN.
Processus métier	Processus de bout en bout de l'entreprise, qui délivre une valeur ajoutée tangible à l'extérieur par une collaboration de plusieurs unités et acteurs. Un processus métier peut avoir une longue durée de vie et est potentiellement interrompu.
QoS Quality of service.	La qualité de service est une notion née chez les opérateurs de télécommunications vers 1997. On parle de contrat de niveau de service quand une entreprise exige de son

	opérateur une haute disponibilité de son réseau. La gestion du niveau de service s'est, depuis, étendue au système d'information.
SAM : Service Activity Monitoring	Ensemble d'outils permettant de surveiller la qualité de service et agir sur les paramètres des services pour satisfaire les demandes des consommateurs et les contraintes des fournisseurs. Ils couvrent les fonctionnalités liées au suivi des erreurs, de la performance, de la qualité et de la sécurité.
SCA : Service Component Architecture	Spécification qui a pour objectifs de fournir un modèle pour création des composants de service dans un éventail de langages de programmation et un modèle réunissant les composants de service dans une solution métier-activités qui sont au cœur de la construction des applications basées sur SOA.
Silo Applicatif	Terme utilisé pour désigner une découpe verticale d'un système, par applications (silos applicatifs).
SLA : Service Level Agreement	Contrat définissant les engagements du fournisseur de service quant à la qualité de sa prestation, et les pénalités engagées en cas de manquement. Cette qualité doit être mesurée selon des critères objectifs acceptés par les deux parties. Ex : temps de rétablissement du service en cas d'incident
SOA : Service Oriented Architecture	Une architecture orientée services consiste essentiellement en une collection de services qui interagissent et communiquent entre eux. Cette communication peut consister en un simple retour de données ou en une activité (coordination de plusieurs services). Le service est un composant clef de l'Architecture Orientée Services. Il consiste en une fonction ou fonctionnalité bien définie. C'est aussi un composant autonome qui ne dépend d'aucun contexte ou service externe. Il est divisé en opérations qui constituent autant d'actions spécifiques que le service peut réaliser.
SOAP : Simple Object Access Protocol	SOAP est un protocole de RPC orienté objet bâti sur XML. Il permet la transmission de messages entre objets distants, ce qui veut dire qu'il autorise un objet à invoquer des méthodes d'objets physiquement situés sur un autre serveur.
TDE : Type de Donnée d'Echange	Structure des informations échangées par les messages ou les flux inter composants. Type des paramètres des opérations de service. Par opposition, les <i>données persistantes</i> sont les informations contenues dans les bases de données.
UDDI : Universal Description Discovery and Integration	UDDI est un annuaire de services fondé sur XML et plus particulièrement destiné aux services Web. Un annuaire UDDI permet de localiser sur le réseau le service Web recherché. C'est un élément clé dans les spécifications de Services Web WS-*, car il permet l'accès aux répertoires des utilisateurs potentiels de services web. UDDI est une spécification mise au point par l'OASIS.
W3C (http://www.w3.org)	Le World Wide Web Consortium, abrégé W3C, est un consortium fondé en octobre 1994 pour promouvoir la compatibilité des technologies du World Wide Web telles que HTML, XHTML, XML, CSS, PNG, et SOAP
Web service	Il s'agit d'une technologie permettant à des applications de dialoguer à distance via Internet, et ceci indépendamment des plates-formes et des langages sur lesquelles elles reposent. Pour ce faire, les services Web s'appuient sur un ensemble de protocoles Internet très répandus (XML, HTTP), afin de communiquer. Cette communication est basée sur le principe de demandes et réponses, effectuées avec des messages XML. Les services web sont décrits par des documents WSDL (Web Service Description Language), qui précisent les opérations pouvant être invoquées, leurs signatures et les points d'accès du service (URL, port .). Les services Web sont accessibles via SOAP, la requête et les réponses sont des messages XML transportés sur HTTP.
Workflow ou Processus de traitement	Processus qui représente le déroulement d'une activité localisée, de courte durée et non interrompible.
WSDL : Web Services Description Language	Le WSDL décrit une Interface publique d'accès à un Service Web, notamment dans le cadre d'architectures de type SOA (Service Oriented Architecture). C'est une description fondée sur le XML qui indique « comment communiquer pour utiliser le service ». Le WSDL sert à décrire le Protocole de communication (SOAP RPC ou SOAP orienté message), le format de messages requis pour communiquer avec ce service, les méthodes que le client peut invoquer ainsi que la localisation du service.

Résumé

Ce mémoire contribue au domaine de la modélisation des systèmes de maintenance industrielle, en proposant une démarche globale d'analyse et de modélisation. Cette démarche s'appuie sur les architectures orientées services (SOA : *Services Oriented Architecture*) en vue de la conception d'un système de maintenance industrielle fondé sur les nouvelles formes de maintenance, notamment la e-maintenance. Cette dernière, offre aux entreprises une flexibilité accrue pour d'une part, la gestion des différentes activités de la maintenance et d'autre part, l'accès aux différentes ressources liées à la maintenance, à travers les technologies du Web, précisément les services Web.

Dans cette optique, nous proposons un cadre méthodologique basé sur des approches issues du domaine de l'orienté services, en mettant en avant les phases importantes du processus développement informatique. Pour cela, nous partons des étapes préliminaires d'analyse du système jusqu'à la spécification détaillée des différents composants de l'architecture orientée services dédiée aux systèmes de la maintenance industrielle, tout en focalisant sur l'apport potentiel des services Web.

La démarche proposée a été appliquée en vue de valider les principes fondateurs sur une étude de cas représentative, relative à la maintenance industrielle dans une unité de conditionnement des huiles d'un groupe agro-alimentaire situé en Algérie.

Ce travail nécessite cependant des améliorations et des extensions dans le cadre de perspectives de recherche, afin d'affiner d'avantage les éléments mis en avant dans la démarche proposée et aussi de l'appliquer sur d'autres domaines applicatifs.

Mots clés : Maintenance industrielle, e-maintenance, architecture orientée services, services Web, analyse, modélisation, spécification, approches orientées services, services métier, processus métier.

Abstract

This memory contributes to the field of the modeling of the industrial maintenance systems, by proposing a global method of analysis and modeling. This method is based on approach of services oriented architectures (SOA) for the design of a industrial maintenance system founded on the new forms of maintenance, in particular e-maintenance. This form of maintenance, offers to the companies an increased flexibility for on the one hand, the management of the various activities of maintenance and on the other hand, the access to the various resources related to maintenance, through technologies of the Web, precisely the Web services.

Accordingly, we propose a methodological framework based on approaches resulting from the field from oriented services, by proposing the important phases of the software process development. For that, we leave the preliminary steps of analysis of the system towards the detailed specification of the various components of services oriented architectures dedicated to the industrial maintenance systems, while focusing on the potential contribution of the Web services.

The proposed method was applied in order to validate the principles founders to a representative case study, relating to the industrial maintenance in a unit of conditioning of oils of an agro-alimentary group located in Algeria.

However, this work requires improvements and extensions within the framework of research perspectives, in order to furthermore refine all the elements highlight in the proposed method and also to apply it to other applicatifs fields.

Key words: Industrial maintenance, e-maintenance, services oriented architectures, Web services, analyzes, modeling, specification, services oriented approaches, business services, business process.