

UNIVERSITE SAAD DAHLEB DE BLIDA

Faculté des sciences

Département d'informatique



MEMOIRE DE MASTER

En Informatique

Option : Système Informatique et Réseaux

THÈME :

**Une ligne de produits orientée service
pour les plateformes de
Téléconsultation médicale**

Réalisé par :

Ali Ben Yahia Mohammed Nadhir

Titouali Soheyb

Encadre par :

Dr. LAHIANI Nesrine

Membre du jury :

Dr. FARAH

Dr. BOUDRAA

Soutenu le :Septembre 2022

Promotion : 2021/2022

Remerciements

Nous tenons tout d'abord à remercier Dieu le tout puissant et miséricordieux, qui nous a donné la force et la patience d'accomplir ce modeste travail.

Nous tenons à remercier sincèrement Madame la promotrice Dr LAHIANI Nasrine, qui a toujours été très disponible tout au long de la réalisation de ce mémoire, ainsi pour l'inspiration, l'aide, la patience et le temps qu'elle a bien voulu nous consacrer et sans qui ce mémoire n'aurait jamais vu le jour.

Nos vifs remerciements vont également au prestigieux membre du jury pour l'intérêt qu'ils ont porté à notre travail en acceptant de l'examiner.

Enfin nous remercions chaleureusement nos chers parents qui nous ont soutenu et encouragés qui ont toujours été là dans les moments difficiles pour nous pousser à aller de l'avant.

Mes remerciements à tous ceux qui m'ont aidé, de près ou de loin, à réaliser ce travail.

Merci.

Dédicaces

C'est avec profonde gratitude que je dédie cet humble travail à la lumière de mes jours, la source de mes efforts « mes parents », qui se sont dépensés pour moi sans compter, et qui m'ont éclairé le chemin avec leurs conseils judicieux. Ce travail est le fruit des sacrifices qu'ils ont consentis pour mon instruction et mon bien-être, que Dieu leurs prête bonheur et longue vie.

A mes très chers frères qui je les souhaite tout le bonheur du monde et la réussite dans leurs études.

A ma famille et mes amies pour leurs soutiens et encouragements

A mes enseignants, particulièrement mon encadreur Dr. LAHIANI Nasrine .

À tous ceux qui ont contribué de près ou de loin à la réalisation de ce travail, Ils vont trouver ici l'expression de mes sentiments de respect et de reconnaissance pour le soutien qu'ils n'ont cessé de me porter.

Sohaib

Dédicaces

Je dédie ce modeste travail particulièrement à mes chers parents, source de vie qui ont consacré leur existence à bâtir la mienne, pour leur soutien, amour, patience et tendresse, pour tous ce qui ils ont fait pour que je puisse arriver à ce stade.

A mes chères frères pour leurs encouragements, amours et soutien.

A toute ma famille source d'espoir et de motivation.

A mes enseignants, particulièrement mon encadreur Dr. LAHIANI Nasrine .

A tous mes amis de promotion .

A toutes les personnes qui m'ont aidé à réaliser ce travail.

Nadir.

Résumé

La réutilisation de logiciels est un problème fondamental et récurrent depuis les origines du génie logiciel. Son objectif principal est de réduire les coûts de développement logiciel en facilitant la réutilisation d'éléments logiciels préexistants. Actuellement, les approches des lignes de produits logiciels et de l'architecture logicielle fonctionnent ensemble pour répondre à ces objectifs.

Notre travail consiste à construire une ligne de produits orientée service pour les plate-formes de Télé-consultation Médicale. L'objectif est de fournir aux utilisateurs du e-santé des services plus efficaces tout en économisant du temps et de l'argent. Ensuite, cette ligne peut générer différentes applications en fonction des besoins des utilisateurs.

Mots clés :

ligne de produit logiciel, service web, Réutilisation, e-santé

Abstract

Software reuse has been a fundamental and recurring problem since the origins of software engineering. Its main objective is to reduce software development costs by facilitating the reuse of pre-existing software elements. Currently, approaches to software product lines and software architecture work together to meet these objectives.

Our job is to build a service-oriented product line for Medical Téléconsultation platforms. The aim is to provide e-health users with more efficient services while saving time and money. Then, this line can generate different applications according to the users' needs.

Keywords :

software product line, Component, Web Service, Reuse, e-health

Table des matières

Table des figures	10
Liste des tableaux	12
Introduction Générale	13
Contexte de travail.....	13
Problématique.....	13
Objectifs du travail.....	14
1 Les lignes de produits logiciel (SPL)	15
1.1 Introduction.....	15
1.2 Ingénierie des lignes des produits	16
1.2.1 Définition des lignes de produits logiciels	17
1.2.2 Ingénierie du domaine	18
1.2.2.1 Analyse du domaine.....	18
1.2.2.2 Conception du domaine	19
1.2.2.3 Réalisation (implémentation) du domaine	19
1.2.3 Ingénierie d'applications	19
1.2.3.1 L'analyse d'application.....	20
1.2.3.2 La conception d'application	20
1.2.3.3 La Réalisation d'application.....	20
1.3 Fonctionnalités et variabilité.....	21
1.3.1 La gestion de la variabilité.....	21
1.3.2 Modélisation de la variabilité	22
1.3.3 Validation du modèle	24
1.4 Orthogonal Variability Model (OVM)	24
1.5 Conclusion	25
2 Les Services Web	26
2.1 Introduction.....	26
2.2 SOA : Service-Oriented Architecture	26
2.3 Les Services Web.....	28
2.3.1 Définition.....	28
2.3.2 Les Principes.....	28
2.3.3 Les caractéristiques des services Web.....	29

2.3.4	Architecture des Services web.....	29
2.3.5	Cycle de vie d'un service web	30
2.3.6	Les langages et les protocoles utilisés par les services web	30
2.3.6.1	XML – EXtensible Markup Language-	30
2.3.6.2	SOAP – Simple Object Access Protocol.....	31
2.3.6.3	WSDL – Service web Description Language.....	32
2.3.6.4	UDDI : Universal Description Discovery and Integration	34
2.3.6.5	WS-BPEL : Web Services Business Process Execution Language .	34
2.3.6.6	REST : Representation State Transfer	35
2.3.7	Fonctionnement des services Web	37
2.3.8	Invocation d'un service web.....	39
2.3.9	Les applications des Services web	40
2.4	Sécurité des web services	40
2.4.1	Vulnérabilités des web services	40
2.4.2	Quelques protocoles de Sécurisation des web services.....	41
2.5	Avantages et inconvénients	41
2.6	Conclusion	42
3	E-santé (E-health)	43
3.1	Introduction	43
3.2	Concepts généraux sur E-santé	43
3.2.1	Définition :	43
3.2.2	Historique de l'E-santé :	44
3.2.3	Pour qui la e-santé ?	44
3.2.4	Dossier patient informatisé	45
3.2.5	Technologies et Applications :	45
3.3	Cadre conceptuel des TIC en santé	46
3.3.1	Définition des TIC	46
3.3.2	Evolution des TIC	46
3.3.3	TIC dans le domaine de la santé	47
3.3.3.1	Genèse des TIC en santé	47
3.3.3.2	La relation entre TIC et la santé	48
3.3.4	Les apports de l'application des TIC en santé	48
3.3.4.1	L'évaluation des TIC en santé.....	48
3.4	La télésanté.....	50
3.4.1	La télémédecine	50
3.4.1.1	La télémédecine en Algérie.....	50
3.4.1.2	Les différents types d'application de la télémédecine	51
3.4.2	m-santé (Mobile-Santé).....	53
3.5	Acteurs de la E-santé.....	54

3.6	Possibilités en matière de la E-santé	54
3.7	Enjeux de la E-santé.....	56
3.7.1	Bénéfices attendus du développement de la e-santé	56
3.7.1.1	Patients :.....	56
3.7.1.2	Établissements de santé et Professionnels du domaine :.....	56
3.7.1.3	Acteurs institutionnels et financeurs publics.....	56
3.8	Problèmes et risques liés à la E-santé.....	57
3.9	Conclusion	57
4	Ingénierie de domaine	58
4.1	Introduction.....	58
4.2	Analyse de domaine :	59
4.2.1	Description du modèle de feature :	60
4.3	Conception du domaine :	62
4.3.1	Présentation d'UML.....	62
4.3.2	Phase d'analyse	63
4.3.3	Phase de conception	66
4.3.3.1	Diagramme de cas d'utilisation	66
4.3.3.2	Diagramme de cas" Admin " :.....	66
4.3.3.3	Diagramme de cas"Médecin " :.....	67
4.3.3.4	Diagramme de cas"Assistant médical" :	67
4.3.3.5	Diagramme de cas"Patient"	68
4.3.3.6	Diagramme de classe	69
4.3.4	Présentation SoaML(Service-oriented architecture Modeling Language)	70
4.3.4.1	Définition SoaML.....	70
4.3.4.2	Diagrammes SoaML :	70
4.3.5	Modélisation de l'architecture d'application	72
4.3.6	Architecture de notre application.....	73
4.4	Conclusion	73
5	Ingénierie de l'application	74
5.1	Introduction.....	74
5.2	Environnement logistique.....	74
5.2.1	Matériels utilisés	74
5.2.2	Environnement de développement	74
5.2.2.1	Technologies de développement	74
5.2.2.2	Outils de programmation.....	75
5.2.3	Dérivation d'une application	75
5.2.3.1	Analyse d'application	76
5.2.3.2	Modélisation d'application.....	76
5.2.3.3	Les interfaces.....	77

5.3 Conclusion.....	88
Introduction Générale	89
Bibliographie	90

Table des figures

1.1	processus classique [18].....	16
1.2	Phases de l'ingénierie LDPs [30]	18
1.3	Processus de développement de l'ingénierie des applications [42].....	20
1.4	Feature Diagram E-Shop [22]	23
1.5	Orthogonal-Variability-Model-Notation [33]	25
2.1	Mécanismes nécessaires pour un environnement d'intégration de services [41]	27
2.2	Architecture d'un service web [11]	30
2.3	Structure d'un message SOAP [2].....	32
2.4	Structure d'un document WSDL [41].....	33
2.5	REST web service Architecture [17].....	36
2.6	REST web service Architecture [13].....	36
2.7	Le fonctionnement des services Web [5].....	37
2.8	Les étapes d'invocation d'un service web [11]	40
3.1	different domaine de E-santé [4]	46
3.2	Plateforme de service de la télémédecine[36]	52
4.1	Feature modèle métiers -1- pour la ligne de produit e-santé.....	59
4.2	Feature modèle métiers -2- pour la ligne de produit e-santé.....	60
4.3	Feature modèle technique pour la ligne de produit e-santé.....	60
4.4	diagramme de cas d'utilisation "Admin"	66
4.5	diagramme de cas d'utilisation " médecin"	67
4.6	diagramme de cas d'utilisation " Assistant médical"	67
4.7	diagramme de cas d'utilisation "patient"	68
4.8	Diagramme de classe de l'application e-health	69
4.9	Diagramme d'interface de service	70
4.10	Diagramme du contrat de service	71
4.11	Diagramme d'architecture des services web	72
4.12	Architecture de notre application.....	73
5.1	Dérivation d'une application	75
5.2	Feature Model d'application (1).....	76
5.3	Feature Model d'application (2).....	76
5.4	Feature Model d'application technique)	77
5.5	Interface Configurateur "1"	78
5.6	Interface Configurateur "2"	79

5.7	Interface Configurateur "3"	80
5.8	Interface Configurateur "4"	81
5.9	Code source de méthode "GET" des médecin.....	81
5.10	Code source de méthode "POST" des médecin.....	82
5.11	Registrassions d'un médecin "OK"	82
5.12	Code source de méthode "PUT"	83
5.13	Code source de "Authentification" des médecin	83
5.14	Test Rest Ful Web Services.....	84
5.15	Afficher les medecin de service Medecin avec la méthode "GET"	85
5.16	Interface d'accueil de patient	86
5.17	Interface de "créer un compte patient"(1)	86
5.18	Interface de "créer un compte patient"(2)	87
5.19	Interface de "prendre RDV"	87
5.20	Interface de "Contacter le médecin"	88

Liste des tableaux

2.1	Récapitulatif de la technologie des services Web.....	38
4.1	Acteurs et leurs tâches.....	65

Introduction Générale

Contexte de travail

La complexité est un problème important à gérer pour les développeurs des systèmes informatiques d'aujourd'hui, mais il en va de même pour l'ensemble du cycle de développement logiciel. Des approches telles que les lignes de produits logiciels apportent des solutions aux problèmes de gestion de la complexité, de réduction des coûts et d'amélioration de la qualité du logiciel final basé sur la réutilisation. La réutilisation devient de plus en plus une méthode et une solution technique pour le développement de systèmes logiciels complexes. L'un des défis majeurs auxquels sont confrontés la communauté de la recherche et l'industrie est de trouver les bons concepts, mécanismes et langages pour une meilleure réutilisation et configuration à grande échelle des systèmes logiciels. De plus, les lignes de produits logiciels sont un paradigme qui préconise une vision de modélisation et de développement dans laquelle l'objectif n'est plus d'obtenir un système logiciel, mais un ensemble de systèmes logiciels avec des caractéristiques communes.

Problématique

Au fil du temps, l'informatique médicale est devenue un véritable discipline scientifique. Les systèmes informatisés sont répandus dans de nombreux domaines de la santé et sont largement répandus dans de nombreux établissements du secteur de la santé. Les progrès en informatique médicale dépendent des méthodes et des outils mis au point par les informaticiens, mais certains programmes de santé sont lourds à utiliser et ne peuvent pas être considérés comme idéaux. D'autre part, la plupart des méthodes existantes pour dériver le SPL (Software Product Line) se concentrent sur le mappage des modèles de caractéristiques aux modèles de variabilité au niveau de la mise en œuvre. Ainsi, des variantes du système sont sélectionnées et configurées dans le produit final et décrites dans des modèles instanciés (par exemple, des modèles UML). Pour représenter les caractéristiques et leurs contraintes. Une expression propositionnelle relie des variantes à des connecteurs logiques et définit un prédicat logique qui sépare les valeurs d'une variable booléenne. Cependant, la dérivation au niveau architectural est rarement considérée comme une étape préparatoire à la dérivation des produits concrets. Par conséquent, les ingénieurs de la lignede produits sélectionnent souvent manuellement les éléments et les variantes de l'architecture de la ligne de produits (PLA) qui font partie d'un produit particulier.

Objectifs du travail

L'objectif de notre travail consiste à concevoir et développer une ligne de produit qui permet de :

- Optimiser le temps, le coût et de minimiser l'effort de développement ;
- Identifier les similarités et variabilités entre les applications de e-Health ;
- Développer les cores assets du domaine (élément réutilisable) :
 - Les diagrammes de modélisation du domaine ;
 - Les services réutilisable.
- Développer des applications e-Health en se basant sur les cores assets ;
- Augmenter la possibilité de réutilisation des éléments logiciels on se basant sur une approche orientée services ;
- Développer des services spécifiques pour les applications finales s'il existe.

Les lignes de produits logiciel (SPL)

1.1 Introduction

À la fin du XIXe siècle, il y avait eu une révolution dans la production industrielle, et il y avait eu un changement radical dans les méthodes de travail, résultant d'une combinaison d'efforts, une productivité accrue et des coûts considérablement réduits. La production de masse a généré des processus qui favorisent le regroupement d'éléments interchangeable sur les lignes de production. Ford Motor Company a développé la première ligne de production pour assembler des voitures en utilisant des pièces interchangeables en 1908.[18]

Ainsi, le concept d'une ligne de produits logiciels propose d'appliquer les principes fordistes au développement logiciel afin que nous puissions appliquer les principes automobiles au développement logiciel. Cependant, la production de logiciels n'a pas suivi le même développement, mais depuis 1990, des mesures ont été prises pour promouvoir de nouvelles méthodes de développement basées sur l'assemblage d'éléments réutilisables et les règles de réutilisation des logiciels. D'où le concept de la ligne de produits logiciels, qui propose d'appliquer les principes fordistes au développement logiciel pour que l'on puisse appliquer les principes automobiles au développement logiciel. En ne développant plus chaque logiciel individuellement, mais en le concevant à travers des éléments réutilisables appelés "assets" (les assets sont des ressources numériques, des composants, des modèles, des processus ou des frameworks qui peuvent être réutilisés). Référentiel de développement logiciel, la principale méthodologie préconisée pour ces principes est regroupée sous un même titre : la méthodologie "ligne de produits logiciels". Évidemment, la rentabilité de ces méthodes dépend principalement de la quantité de similarité partagée entre les logiciels et de la quantité de logiciels à produire. L'objectif est de pouvoir fournir plus rapidement des logiciels qui répondent aux exigences des clients, de dépenser moins d'argent et d'améliorer la qualité du logiciel. L'objectif du développement d'une ligne de produits logiciels est de produire une famille de produits logiciels plutôt que de fournir un seul produit. Ce chapitre présente les notions principales des lignes de produits, il est organisé comme suit : dans la première section nous introduisons la définition des lignes de produits logiciels. Ensuite, dans la deuxième section nous présentons le processus de développement des lignes de produits logiciels qui est structuré en deux sous-processus : l'ingénierie du domaine et l'ingénierie des applications. Et en fin, nous introduisons la notion de variabilité dans les LdPs.

1.2 Ingénierie des lignes des produits

En termes de génie logiciel dit classique, le processus de développement interactif se compose de quatre phases successives :

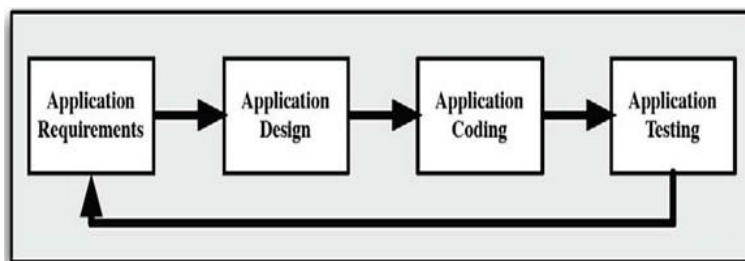


FIGURE 1.1 – processus classique [18]

Première étape : Définir les exigences (Application Requirements).

Deuxième étape : Spécification de la solution répondant à ces besoins (Application Design).

Troisième étape : Mise en place de cette solution (Application Coding).

Quatrième étape : La phase de test de la solution et son évaluation par rapport aux besoins initiaux (Application Testing).

L'Ingénierie des lignes de produits logiciels est un moyen de développer des logiciels qui économise du temps, des coûts et avec une bonne qualité. C'est une transposition des chaînes de production au monde du logiciel. L'ingénierie des lignes de productions logicielles vise à développer une famille de produits, il y a donc des similitudes entre les produits qui composent la ligne de produits logiciels qui est basée sur la réutilisation et la configuration des systèmes logiciels.

Krueger définit la réutilisation comme : "Le processus de création des systèmes logiciels à partir des logiciels existants au lieu de les créer à partir de zéro" [19]

L'ingénierie de ligne de produits logiciels comporte trois activités principales :

- (1) L'ingénierie de domaine ;
- (2) L'ingénierie d'application ;

La gestion de variabilité. Dans la section suivante on va citer quelques définitions des lignes de produits logiciels pour donner une idée plus précise sur ce principe.

1.2.1 Définition des lignes de produits logiciels

Selon Clements et Northrop : " une ligne de produits logiciels comporte un ensemble de produits logiciels qui partagent des caractéristiques (Features) communes et satisfaisantes des besoins d'un domaine particulier. [8]

C'est-à-dire au lieu de développer les applications individuellement, on définit une plate-forme commune (architecture du système) qui sera réutilisée dans toutes les applications du domaine. L'ingénierie des lignes de produits logiciels s'intéresse plutôt à modéliser et développer une famille de produits.

La définition propose par Jan Bosch est la suivante : "A software product line consists of product line architecture and a set of reusable components that are designed for incorporation into the product line architecture. In addition, the product consists of the software products that are developed using the mentioned reusable assets." [7]

Cette définition, montre l'importance du concept d'architecture logicielle dans une ligne de produits. Par conséquent, une ligne de produits est définie par cette architecture (appelée architecture de référence) et un ensemble de composants réutilisables au sein de cette architecture. [16]

Les architectures de référence fournissent un cadre pour développer des composants réutilisables et assurer leur bonne combinaison. En tant que tel, il semble être un guide pour assembler des artefacts(assets) réutilisables pendant la phase de conception du développement de produits. Il est ainsi possible de créer un produit unique en réutilisant les artefacts prévus à cet effet.

L'ingénierie des lignes de produits logiciels est dévisée en deux phases complémentaires : l'ingénierie de domaine et l'ingénierie d'application. [31]

Dans la phase de l'ingénierie de domaine(domain engineering) en définit la plate forme de base de la ligne de produits, alors que l'ingénierie d'application s'appuie sur le développement (dérivation) des applications à partir de la plate-forme commune. Le principe derrière ces deux phases rejoint l'idée de compenser le temps de développement des artefacts communs, pendant la première phase, par la réutilisation de ces artefacts pendant la deuxième phase. Ces deux activités seront présenter on détaille dans la section suivent.

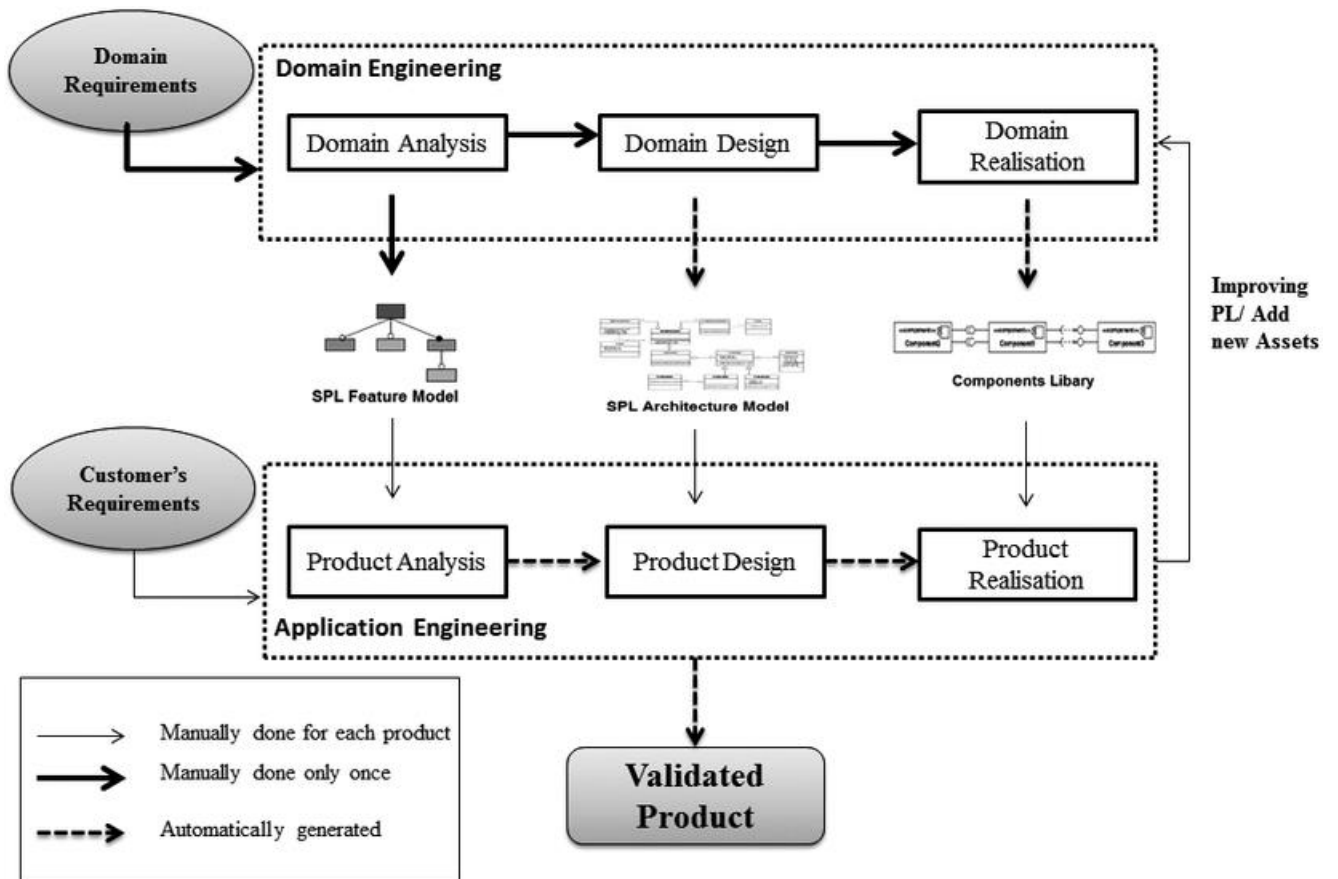


FIGURE 1.2 – Phases de l'ingénierie LDPs [30]

1.2.2 Ingénierie du domaine

L'objectif principal est de produire des artefacts réutilisables et de fournir des moyens permettant leur utilisation effective pour construire un nouveau produit au sein d'une ligne de produits. On peut voir ces artefacts comme le résultat du transfert de connaissances avancées sur un domaine (celui de la ligne de produits) dans un monde informatique, sous une forme configurable. [41]

1.2.2.1 Analyse du domaine

L'objectif est d'étudier le domaine pour identifier les caractéristiques communes et variables pour la famille des produits (features).

L'analyse de domaine implique la collecte, l'extraction et la formalisation des informations (connaissances) dans le domaine, c'est-à-dire la transformation des connaissances métier en représentations informatiques et l'identification des règles métier, des contraintes et des exigences. Dans l'analyse de domaine, on extrait les éléments communs et autres éléments variables de tous les produits en fonction des exigences (variables) du domaine, cette étape est la plus importante car le développement des lignes de produits est basé sur la notion de variables. L'analyse de domaine est une activité plus complexe que l'analyse des exigences lors du développement

d'une seule application. Une activité essentielle de cette étape consiste à déterminer le périmètre fonctionnel (scope), qui est le périmètre de la ligne de produits qui définit explicitement tous les produits à étudier et définit implicitement les futurs produits qui pourraient entrer dans la ligne de produits.

1.2.2.2 Conception du domaine

Le but de la conception de domaine est de concevoir des éléments logiciels qui feront partie d'une ligne de produits; ces éléments sont des composants configurables et des architectures de référence. Clements propose cette définition sur l'architecture de référence : "Description of the structural properties for building a group of related systems (i.e., product line), typically the components and their interrelationships. The inherent guidelines about the use of components must capture the means for handling required variability among the systems. (This is called a reference architecture)" [41]

Cette définition considère une architecture de référence comme une description d'un ensemble de composants et de leurs relations permettant de construire des systèmes logiciels appartenant à une famille de produits. De plus, l'architecture de référence doit contenir les informations nécessaires pour gérer la variabilité au sein du système. Une architecture de référence définit de manière abstraite les composants structurels d'un produit et leurs relations. Une architecture de référence est un élément clé d'une plate-forme de ligne de produits. Habituellement, les experts dans le domaine se concentrent d'abord sur cela. Ils se basent sur tous les modules communs identifiés lors de l'analyse du domaine pour obtenir la structure initiale de l'architecture.

1.2.2.3 Réalisation (implémentation) du domaine

Cette étape consiste à détailler et implémenter les différentes parties réutilisables de l'architecture dans des composants logiciels réutilisables. Il s'agit principalement de construire les assets.

Un « asset » : un artefact logiciel nécessaire au développement de produits, qui peut être des fichiers de code source, des bibliothèques, des modèles...ect. De plus il faut associer à chaque « feature model » les artefacts logiciels permettant de l'implémenter.

Ce modèle doit être flexible pour faciliter la maintenance et la réutilisation dans la partie ingénierie d'application.

1.2.3 Ingénierie d'applications

C'est la mise en oeuvre d'un produit issu du domaine. L'objectif de dériver de manière semi-automatique les logiciels finaux à partir des éléments réutilisables.

Une ligne de produits doit intégrer des contraintes de cohérence pour la sélection lors de la dérivation. Bien que cette activité soit fondamentale pour le succès de l'ingénierie des lignes de produits dans le monde du logiciel, elle n'a pas reçu beaucoup d'attention jusqu'à récemment.

Ce processus de développement se base sur les artéfacts réutilisables produits lors de la phase d'ingénierie du domaine. Plus précisément, la construction d'un produit logiciel spécifique est un processus d'assemblage et de configuration des artéfacts réutilisables en conformité avec l'architecture de référence.

L'ingénierie des applications est divisée en trois activités principales. Ces trois activités du développement sont : l'analyse d'application, la conception d'applications et finalement la réalisation d'applications.

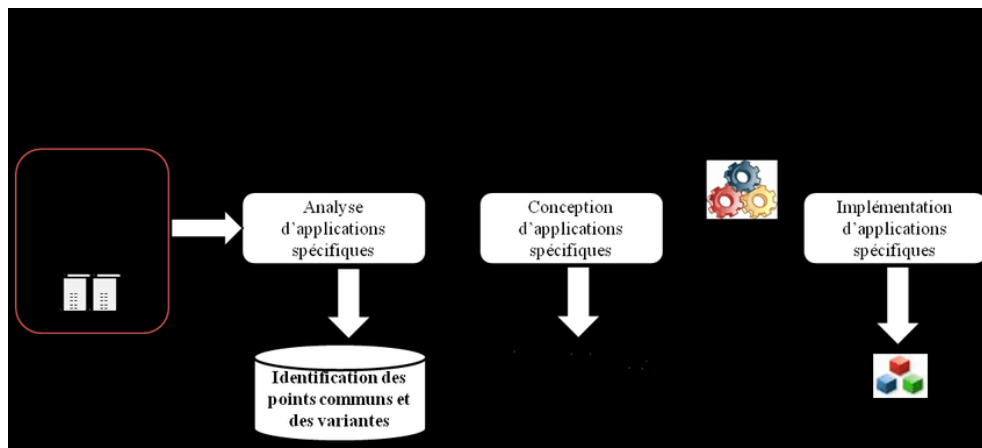


FIGURE 1.3 – Processus de développement de l'ingénierie des applications [42]

1.2.3.1 L'analyse d'application

Consiste à l'analyse des exigences du produit, on trouve ces exigences à partir des différents intervenants du développement de produits (les utilisateurs finaux). Il y a trois types des exigences : -les exigences du domaine. -les exigences prévues dans l'analyse du domaine sous forme de variabilités. -les exigences non prévues dans l'analyse du domaine, qui donnent lieu à des configurations particulières.

1.2.3.2 La conception d'application

Cela inclut la recherche de l'architecture du produit à partir de l'architecture globale définie lors de la conception du domaine, ainsi que l'instanciation et la configuration des points de modification définis dans la conception du domaine lors de cette phase de développement pour créer l'architecture de l'application.

1.2.3.3 La Réalisation d'application

Inclure les produits spécifiques requis pour la mise en œuvre. Assemblez ensuite les composants correspondant aux parties sélectionnées à l'étape précédente.

1.3 Fonctionnalités et variabilité

La variabilité est une notion très importante liée aux lignes de produits logiciels, qui permet de créer différentes applications répondant aux exigences des différents utilisateurs.

La variabilité a été définie comme le « regroupement des caractéristiques qui différencient les produits de la même famille », et cela nous fait parler sur les points de variation. Les points de variation peuvent être définis comme abstraits dans la ligne de produits et redéfinis par chaque produit d'une manière spécifique. Il existe quelques éléments logiciels directement réutilisables. Avant de réutiliser un élément, il est nécessaire de déterminer dans quelles conditions il peut effectivement être réutilisé et comment. L'élément peut être réutilisé dans différents contextes, et il est utilisé dans des conditions et sa utilisation sera plus facile. De manière simplifiée, chaque produit appartenant à une ligne de produits logiciels définit un contexte spécifique.

On classifie les différents types de variabilité dans les lignes de produits comme suite :

-Similarité : Il s'agit d'une caractéristique commune à tous les produits de la série. Cette fonctionnalité est implémentée dans le cadre de la plate-forme ou du noyau commun de la famille.

-Variabilité : Il s'agit d'une caractéristique que certains produits peuvent avoir en commun, mais pas tous les membres de la famille de produits. Cette fonctionnalité doit être modélisée comme un point de changement et mise en œuvre d'une manière qui ne se produit que dans des produits sélectionnés.

- Produit-Spécifique : Il s'agit d'une fonctionnalité qui n'est incluse que dans un ou une partie des produits de la série. Il s'agit généralement d'une réponse à une demande spécifique d'un client pour un produit connexe. Cette fonctionnalité ne sera pas intégrée à la plateforme de la ligne de produits, mais devra être compatible avec celle-ci.

L'identification, l'utilisation, la représentation, la mise en œuvre et l'évolution de la variabilité dans une lignée de produits s'appelle : la gestion de la variabilité.

Toute l'architecture d'une ligne de produits et ses propriétés dépendent de la gestion de la variabilité, activité qui constitue le principal enjeu lors de la conception d'une ligne de produits.

1.3.1 La gestion de la variabilité

La variabilité est au cœur d'une ligne de produits logiciels. Chaque produit appartenant à une famille de produits logiciels identifie un contexte spécifique. Chaque produit est constitué d'un ensemble d'Éléments Logiciels exprimés en « Fonctionnalités ». Chaque fonctionnalité rassemble les exigences à respecter par le produit logiciel final qui contient cette fonctionnalité. Le but d'une fonctionnalité est de décrire un ensemble d'exigences étroitement liées qui peuvent être directement réutilisées par différents produits finaux. Les caractéristiques sont séquentiellement décomposées en sous-caractéristiques jusqu'à ce que des caractéristiques terminales soient obtenues. Idéalement, chaque fonction finale est associée à un élément logiciel réutilisable

(composant, service, etc.) qui satisfait aux exigences identifiées par la fonction correspondante. La gestion de la variabilité joue donc un rôle essentiel pour déterminer quand, dans quelles conditions et comment un feature (et les éléments logiciels associés) peuvent être réutilisés de manière optimale.

La gestion de la variabilité est décomposée en trois activités principales : L'identification, implémentation, l'évolution.

L'identification de la variabilité qui détermine :

- les features terminales qui permettent de distinguer les produits logiciels appartenant à la même ligne de produits (variabilités) ;
- les contraintes qui existent entre ces features ;
- où, comment et pourquoi ces variabilités peuvent apparaître (points de variation). Les points de variation correspondent souvent aux features non terminales qui facilitent la structuration des features terminales.

L'implémentation de la variabilité qui détermine quels mécanismes (héritage, paramétrisation, configuration, génération, template instanciation, plugins, . . .) peuvent être utilisés afin de la retranscrire au niveau du code ou de l'architecture.

L'évolution de la variabilité montre comment éviter d'introduire des conflits ou des interactions involontaires entre features lorsque de nouveaux points de variation ou de nouvelles variantes émergent.

Les outils de gestion de variabilité offrent principalement trois types de fonctionnalités : a. Des environnements de développement spécifiques aux lignes de produits logiciels. b. Des outils de configuration et de gestion du changement destinés aux lignes de produits logiciels. c. Des outils de tests et de vérification adaptés aux lignes de produits logiciels.

1.3.2 Modélisation de la variabilité

Après avoir défini le concept de variabilité, nous devons trouver un moyen de l'exprimer clairement, c'est-à-dire de trouver un langage de modélisation qui nous permette d'intégrer les concepts liés au logiciel de la ligne de produit pour représenter notre système.

La modélisation de la variabilité est une technique utilisée d'une part pour documenter la variabilité et d'autre part pour justifier la variabilité.

Ses principaux objectifs sont :

- o Explicitation de la variabilité dès les premières phases du projet ;
- o Diminution de la complexité liée à la gestion de la variabilité tout au long du processus de développement.

De nombreux langages permettent la description graphique de la variabilité. Le premier langage dédié à la modélisation de la variabilité. Ce langage de modélisation définit diverses structures (graphiques et textuelles) pour modéliser les Features et leurs relations avec un modèle spécifique appelé "Feature diagram". Ce langage est avant tout conçu pour être minimal et facile à utiliser par rapport à d'autres langages de modélisation plus complexes comme UML.s que UML.

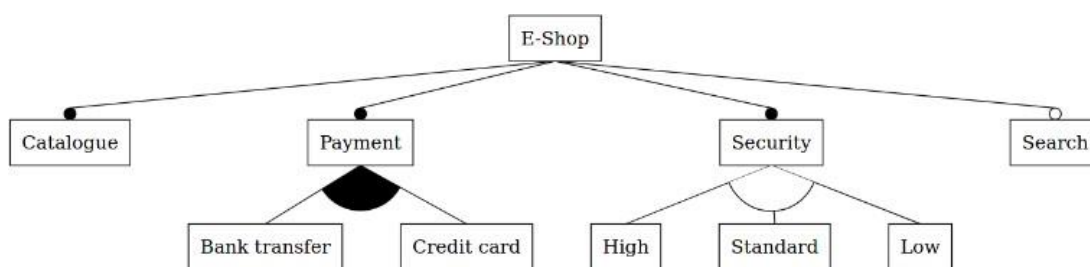


FIGURE 1.4 – Feature Diagram E-Shop [22]

Un Feature Diagram se présente généralement sous la forme d'un arbre composé de nœuds reliés par des arêtes. Chaque nœud correspond à une à une fonctionnalité (feature) et chaque arête correspond à une relation entre deux à une fonctionnalité (features). Différents types de features existent :

- La racine : Un Feature Diagram est caractérisé et identifié par une feature spécifique appelée la racine (root). Cette racine détermine le point d'entrée du diagramme qui est l'unique nœud ne possédant pas de parent ;
- Les features optionnelles : Si une feature est optionnelle (optional) alors si un de ses parents est sélectionné elle ne l'est pas nécessairement. Graphiquement, une feature optionnelle est décorée par un cercle vide ;
- Les features obligatoire(mandatory) : Si une feature est obligatoire (mandatory) alors si un de ses parents est sélectionné elle est dès lors aussi sélectionnée. La racine est par définition toujours obligatoire. Graphiquement, une feature obligatoire n'est pas décorée par un cercle vide.

Ces features sont reliées par différents types de relations :

- Les décompositions. Les features peuvent être décomposées en sous-features suivant différents types de décompositions. Ces décompositions définissent des contraintes entre des features partageant le même parent ;
- Alternative : Cette décomposition signifie que si le parent est sélectionné alors ses enfants peuvent être sélectionnés ;

- Or : Cette décomposition signifie que si le parent est sélectionné alors au moins un de ses enfants peut être sélectionné.

Dans certains cas, on veut pouvoir exprimer des contraintes entre des features qui ne partagent pas le même parent. Ces contraintes peuvent être établies entre toutes les features appartenant au même Feature Diagramme. Elles sont représentées de manière textuelle afin d'éviter tout surchargement réduisant la lisibilité du modèle

Les deux contraintes les plus utilisées sont :

- Requires : La contrainte requires entre deux features (f1 requires f2) permet d'exprimer que si la feature f1 est sélectionnée alors f2 doit nécessairement l'être, mais pas inversement ;
- Excludes : La contrainte excludes entre deux features (f1 excludes f2) permet d'exprimer que si une des deux features est sélectionnée alors l'autre feature ne peut pas l'être en même temps pour le même produit.

1.3.3 Validation du modèle

Le modèle de feature doit être validé avant son utilisation. Il est assez simple. La validation est effectuée par l'instanciation du modèle pour chaque application envisagée dans le domaine d'analyse et vérifier si le modèle instancié représente correctement les features de l'application.

1.4 Orthogonal Variability Model (OVM)

OVM est un langage de modélisation permet de déterminer la variabilité d'une ligne de produits logiciels d'une manière orthogonale, en d'autres termes, il fournit une vue en coupe transversale de la variabilité entre tous les objets de la ligne de produits. OVM inter-relie la variabilité dans les modèles de base tel que les modèles d'exigences, les modèles de conception, les modèles de composants.

OVM est une approche plus récente utilisée pour documenter la variabilité son principal objectif est de définir explicitement les point de variabilité et les variant d'une ligne de produit logicielle.

L'OVM nous permet aussi de gérer la modélisation de l'architecture où chaque composant réalise une variante n'est prise que dans une application pour laquelle la variante correspondante a été liée.

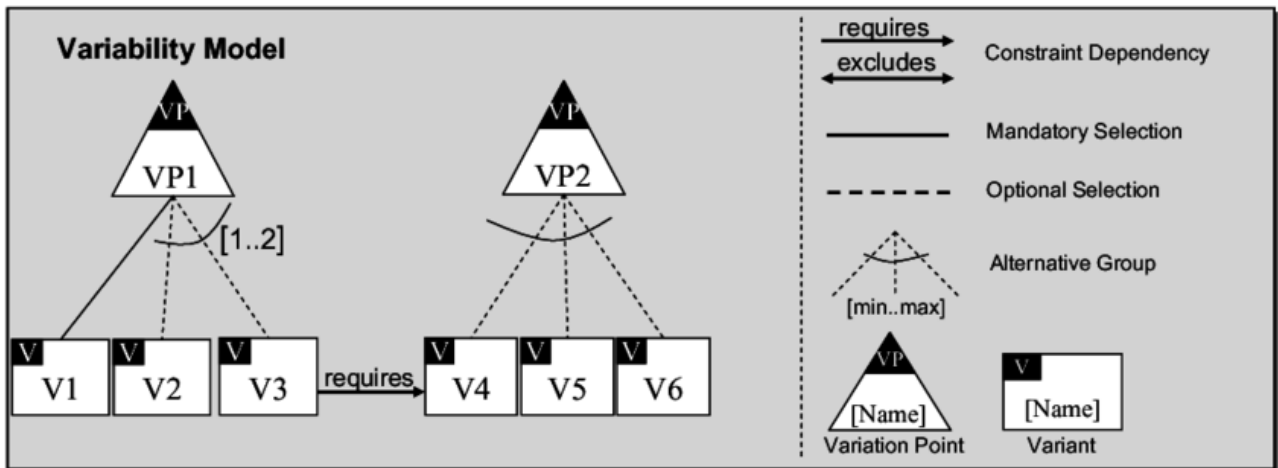


FIGURE 1.5 – Orthogonal-Variability-Model-Notation [33]

Un OVM est composé de deux éléments principaux : les points de variation (VP) et variantes (V).

Les points de variation sont des aspects qui montrent des variations qui peuvent couvrir différents produits. Les points de variation étant sélectionné par le client ou l'ingénieur de la ligne de produits logiciels. Les points de variation peuvent changer, car les variations sont liées aux ces points. La relation entre les points de variation et les variantes est similaire à une relation père-fils. Un point de variante a au moins un enfant et chaque variante a au plus un parent.

Bien que les Feature Diagram soient similaires à OVM, leur principale différence réside dans leur structure et leur relation avec les informations. Les Feature Diagram se composent uniquement d'éléments organisés en un seul arbre et une seule entité racine, tandis qu'OVM se compose de points de variation et de variantes organisés en un ou plusieurs arbres. La fonction racine est toujours requise dans le Feature Diagram. Autrement dit, la fonction racine fait partie de chaque produit, il n'y a donc pas de produits vides. Les points de variation peuvent être facultatifs. Cela permet au produit d'être configuré sans variantes et sans points de variation.

1.5 Conclusion

Dans ce chapitre, nous avons introduit les concepts de base des lignes de produits logiciels, en particulier le concept de variabilité, sa gestion et sa modélisation, ainsi que les avantages et les enjeux du développement de lignes de produits.

Les Services Web

2.1 Introduction

Les approches orientées services [32] ont été introduites comme un nouveau paradigme pour le développement logiciel durant les années 2000. Ce paradigme utilise la notion de service comme élément de base pour la construction d'applications logicielles. Un service est défini comme une entité logicielle qui peut être utilisée de façon statique ou dynamique pour la réalisation d'une application logicielle. Un consommateur, ou client, sélectionne un service à partir de sa description. Il l'utilise sans avoir connaissance de la technologie sous-jacente nécessaire à son implantation ni de sa plate-forme d'exécution. Inversement, le service ne connaît pas le contexte dans lequel il va être utilisé par un client. Cette indépendance à double sens est une propriété forte des services qui facilite le faible couplage.

Chaque service est constitué de deux parties : sa description et son implémentation. Le fournisseur de service définit la syntaxe de l'interface, la sémantique des opérations et les comportements du service dans la description de service. Il peut également décrire certaines propriétés non fonctionnelles telles que la qualité du service, le coût, la localisation, le nombre d'appels autorisés à ce service et par ce service, etc. Ces propriétés sont généralement décrites en utilisant des langages fondés sur XML et des protocoles standards de l'Internet. Une architecture à service (SOA pour Service-Oriented Architecture) regroupe un ensemble de services et des mécanismes d'assemblage permettant le développement d'applications basées sur la réutilisation de services. Lors de la sélection d'un service, un contrat est mis en place, de façon tacite ou explicite, entre consommateur et fournisseur. La description du service peut être considérée comme le contrat de base. Cependant, une négociation peut avoir lieu entre l'utilisateur du service et le fournisseur du service. Les contrats de services permettent de réduire le couplage lié aux dépendances lors de la création d'une application par assemblage de services. Ceci améliore aussi le niveau d'abstraction des applications et élève la granularité de la réutilisation.

2.2 SOA : Service-Oriented Architecture

SOA est apparu en 1996 dans une note de recherche du Gartner Group.

« L'architecture orientée service constitue un style d'architecture basée sur le principe de séparation de l'activité métier en une série de services. Ces services peuvent être assemblés et liés entre eux selon le principe de couplage lâche pour exécuter l'application désirée. »[10]

L'architecture orientée service (SOA) est une philosophie d'organisation et de gestion des systèmes. Collecte d'informations sur les bonnes pratiques résultant des développements Génie logiciel. L'objectif de cette structure est de réguler la distribution des ressources, et les fonctions des systèmes d'information de base et des ressources en technologie de l'information pour les différents intervenants dans les processus opérationnels.

Nous pouvons diviser leurs éléments en deux catégories, comme le montre la figure 2.1 :

les mécanismes de base qui assurent la publication, la découverte, la composition, la négociation, la contractualisation et l'invocation des services ;

les mécanismes additionnels qui prennent en charge les besoins non-fonctionnels tels que la sécurité, les transactions ou encore la qualité de service.

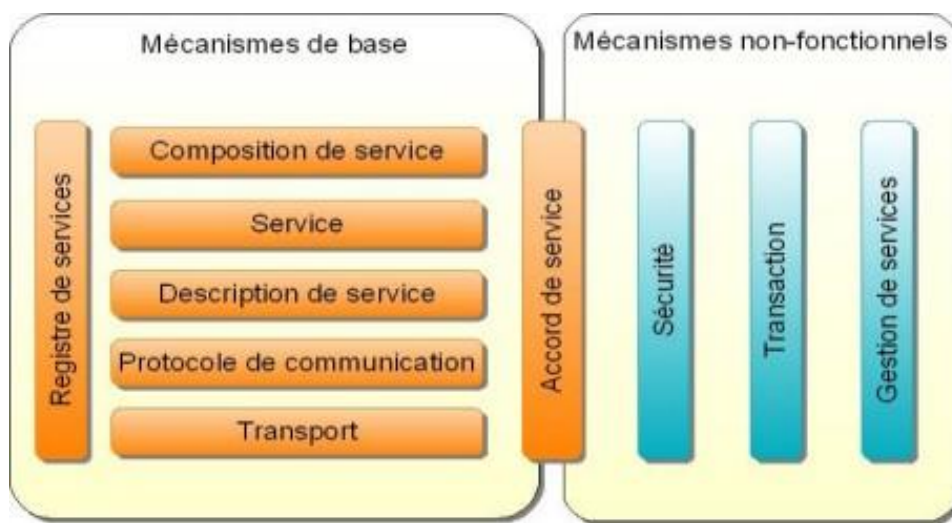


FIGURE 2.1 – Mécanismes nécessaires pour un environnement d'intégration de services [41]

Les mécanismes de transport et le protocole de communication sont à la base de l'environnement et de l'intégration des services. Ils assurent la communication, c'est-à-dire les demandes et entre les différents acteurs. Pour réaliser des interactions de base, il est également nécessaire de pouvoir décrire en langue une description de l'environnement d'intégration. Cette description doit être une compréhension de la fonction du service, de ses éléments non fonctionnels et de la façon dont il devrait être. Au contraire, il est nécessaire d'ajouter un journal de service qui permet l'enregistrement de son service, afin que le client puisse rechercher un service qui réponde à ses besoins.

La convention de services fait partie des mécanismes sous-jacents, mais aussi de la partie non fonctionnelle, représentant le contrat existant entre le fournisseur de services et le client. Dans le présent contrat, les fonctions à exécuter par la Prestation sont déterminées. Mais elle identifie également des caractéristiques non fonctionnelles, telles que le temps de réponse ou la fiabilité, auxquelles le service est lié.

Les services représentent, bien évidemment, le cœur d'une SOA. Il est donc logique de trouver des standards concernant l'exposition et l'invocation de services. Le premier, et le plus important de ces standards, est celui des Web Services (WS).

2.3 Les Services Web

Les services Web sont certainement la technologie la plus connue et la plus populaire dans le monde industriel et académique pour la mise en place de l'architecture à services. Dans une première partie, nous allons définir les services Web. Ensuite, nous allons présenter les principes des services Web et détaillerons les trois standards utilisés par cette technologie, que sont WSDL, UDDI et SOAP.

2.3.1 Définition

Un service web est un protocole d'interface informatique de la famille des technologies web permettant la communication et l'échange de données entre applications et systèmes hétérogènes dans des environnements distribués. Il s'agit donc d'un ensemble de fonctionnalités exposées sur internet ou sur un intranet, par et pour des applications ou machines, sans intervention humaine, de manière synchrone ou asynchrone. Le protocole de communication est défini dans le cadre de la norme SOAP dans la signature du service exposé (WSDL). Actuellement, le protocole de transport est essentiellement TCP (via HTTP). Le concept a été précisé et mis en œuvre dans le cadre de Web Services Activity2, au W3C, particulièrement avec le protocole SOAP. Associé avec les échanges de données informatisés (EDI), le consortium ebXML l'a utilisé pour automatiser des échanges entre entreprises. Cependant le concept s'enrichit avec l'approfondissement des notions de ressource et d'état, dans le cadre du modèle REST, et l'approfondissement de la notion de service, avec le modèle SOA

2.3.2 Les Principes

Les services Web ont été créés pour rendre disponibles des applications sur l'Internet ou dans un Intranet. Ces services respectent les principes de l'approche orientée service précédemment présentés ; ils sont donc décrits, publiés et découverts. Un fournisseur de service Web enregistre son service, en décrivant ses fonctionnalités et certains de ses aspects non-fonctionnels dans un fichier WSDL, auprès d'un annuaire UDDI. Un client interroge un annuaire UDDI pour trouver un service qui répond à ses besoins. Pour le consommateur, le service Web est une boîte noire qui ne donne pas de détails techniques sur son implantation, seulement des informations sur ses fonctionnalités et quelques propriétés, sa localisation et les moyens pour l'interroger. Les communications se font par le protocole SOAP. [41]

2.3.3 Les caractéristiques des services Web

La technologie des services Web repose essentiellement sur une représentation standard des données (interfaces, messageries) au moyen du langage XML. Cette technologie est devenue la base de l'informatique distribuée sur Internet et offre beaucoup d'opportunités au développeur Web.

Un service Web possède les caractéristiques suivantes :

- Il est accessible via le réseau ;
- Il dispose d'une interface publique (ensemble d'opérations) décrite en XML ;
- Ses descriptions (fonctionnalités, comment l'invoquer et où le trouver ?) sont stockées dans un annuaire ;
- Il communique en utilisant des messages XML, ces messages sont transportés par des protocoles Internet (généralement HTTP, mais rien n'empêche d'utiliser d'autres protocoles de transfert tels : SMTP, FTP, BEEP...) ;
- l'intégration d'application en implémentant des services Web produit des systèmes faiblement couplés, le demandeur du service ne connaît pas forcément le fournisseur.

Ce dernier peut disparaître sans perturber l'application cliente qui trouvera un autre fournisseur en cherchant dans l'annuaire.

2.3.4 Architecture des Services web

Jusqu'à présent, l'accès aux ressources d'application ou aux bases de données sur Internet se fait En envoyant des requêtes basées sur des langages de script (PHP, JSP, etc.). Il s'agit donc d'une conversation entre couches de présentation basée sur HTML (protocole) http) et des applications installées sur des serveurs distants. Avec les services web, des conversations peuvent désormais s'établir entre des applications qui peuvent être installés sur des machines distantes grâce au standard XML. En effet, pour avoir une conversation sur Internet, ces applications doivent "parler" la même langue, un xml.

L'architecture de référence des services web est basé sur les trois concepts suivants :

- **Le fournisseur de service** : C'est le propriétaire du service.
- **Le client (ou le consommateur de service)** : C'est un demandeur de service. D'un point de vue technique, il est constitué par l'application qui va rechercher et invoquer un service.
- **L'annuaire des services** : C'est un registre de descriptions de services offrant des facilités de publication de services à l'intention des fournisseurs, Ainsi que des facilités de recherche de services à l'intention des clients.

Les interactions de base qui existent entre ces trois éléments sont les opérations de publication, de recherche, d’invocation et de lien (bind). Pour bien comprendre le fonctionnement de cette architecture, nous expliquons ci- dessous le rôle de chacun des éléments précédents :

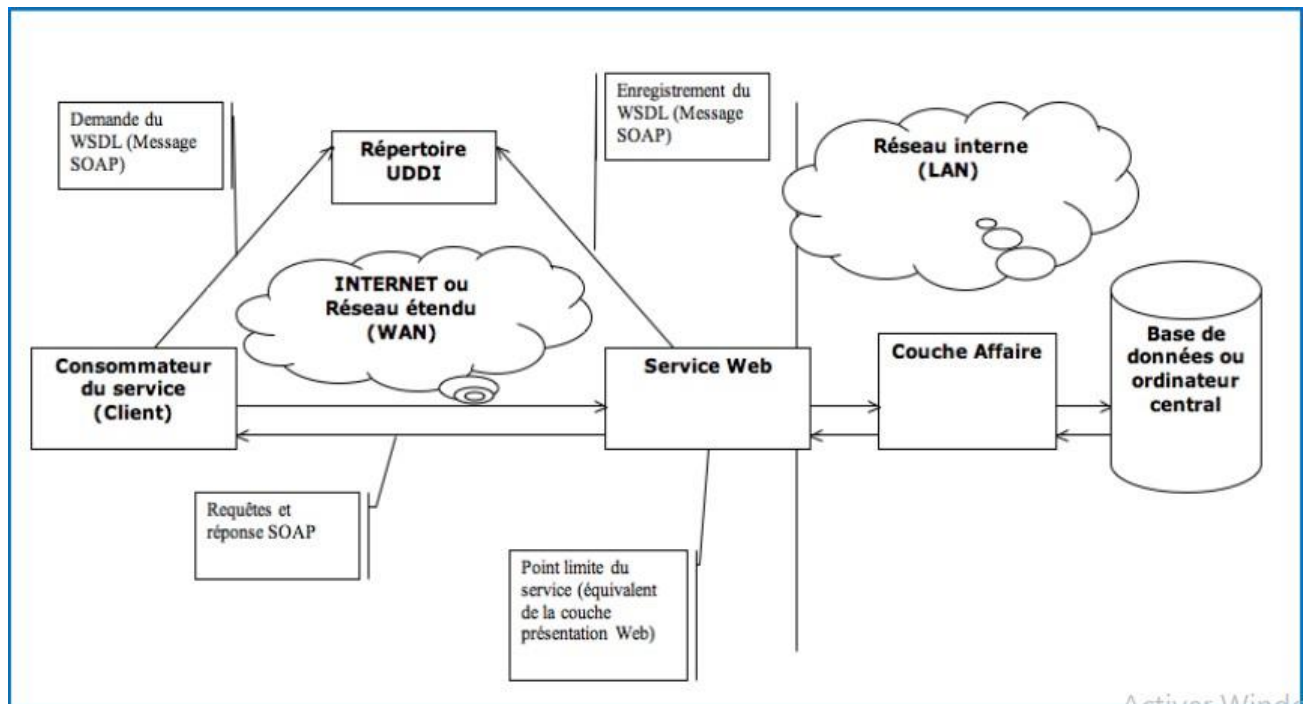


FIGURE 2.2 – Architecture d’un service web [11]

2.3.5 Cycle de vie d’un service web

L’emploi d’un service web connaît le cycle de vie suivant [11] :

- D’abord, on effectue le déploiement du service web, en fonction de la plate-forme ;
- Ensuite, on enregistre le service web à l’aide de WSDL (Services web Description Language), dans l’annuaire UDDI ;
- L’étape suivante est la découverte du service web par le client, par l’intermédiaire d’UDDI qui lui donne accès à tous les services web inscrits. Pour ce, on utilise SOAP ;
- Enfin, Le client invoque le service web voulu, ce qui termine le cycle de vie de ce service web.

2.3.6 Les langages et les protocoles utilisés par les services web

2.3.6.1 XML – EXtensible Markup Language-

Langage de balisage extensible est un langage de balisage généraliste recommandé par le W3C, est un sous-ensemble du langage SGML Langage de balisage extensible est un langage

de balisage généraliste recommandé par le W3C comme HTML. XML est un sous-ensemble du langage SGML. Autrement dit, contrairement aux autres langages de balisage, XML n'est pas prédéfini et nous devons définir nos propres balises. L'objectif principal de ce langage est d'échanger des données entre différents systèmes. . De nombreux autres langages sont basés sur XML tels que XHTML, MathML, SVG, XUL, XBL, RSS, RDF. Vous pouvez créer votre propre langage basé sur XML.

Exemple :

```
<?xml version="1.0" encoding="UTF-8" ?>
<Etudiant>
  <nom></nom>
  <prenom></prenom>
  <Matricule></Matricule>
</Etudiant>
```

2.3.6.2 SOAP – Simple Object Access Protocol

SOAP est un protocole de communication basé sur XML qui permet aux applications d'échanger des informations via HTTP. Par conséquent, il permet l'accès aux services Web et l'interopérabilité des applications sur le Web. SOAP est un protocole simple et léger qui repose entièrement sur des normes établies telles que HTTP et XML. Étant portable, il ne dépend pas du système d'exploitation ou du type d'ordinateur. SOAP est une spécification non exclusive.

Un message SOAP est un document XML constitué d'une enveloppe composée de deux parties : l'entête(header) qui peut être facultatif et le corps (body).

- SOAP est un moyen XML de définir quelles informations envoyer et comment les envoyer ;
- SOAP est un protocole de communication conçu pour communiquer sur Internet ;
- SOAP assure le transport de données pour les services Web ;
- SOAP peut être utilisé pour diffuser des messages ;
- SOAP facilite la connexion des applications clientes aux services distants et d'appeler des méthodes distantes.

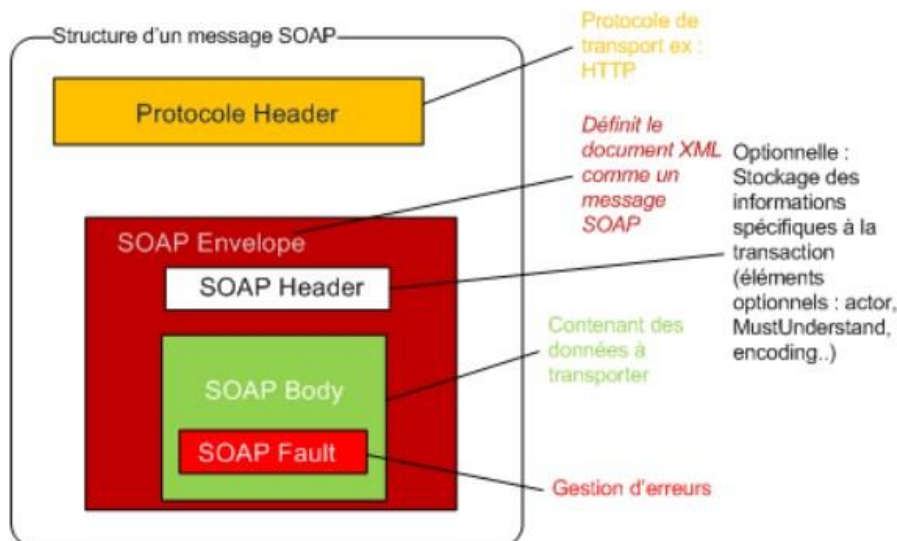


FIGURE 2.3 – Structure d'un message SOAP [2]

- **SOAP envelope** (enveloppe) est l'élément de base du message SOAP. L'enveloppe contient la spécification des espaces de désignation (namespace) et d'encodage des données.
- **SOAP header** (entête) est une partie facultative qui permet d'ajouter des fonctionnalités à un message SOAP de manière décentralisée sans nécessiter un accord entre les parties communicantes. Ceci indique si le message est obligatoire ou facultatif. L'en-tête est particulièrement utile lorsque le message doit être traité par plusieurs intermédiaires.
- **SOAP body** (corps) est le conteneur des informations pour le destinataire du message, qui contient les méthodes et les paramètres qui seront exécutés par le destinataire final.
- **SOAP fault** (erreur) est un élément facultatif défini dans le corps SOAP pour signaler les erreurs.

2.3.6.3 WSDL – Service web Description Language

Est un langage de description standard. Il s'agit de l'interface présentée à utilisateur. Il montre comment utiliser et interagir avec les services Web. WSDL est basé sur XML et permet Décrivez avec précision les détails du service Web tels que le protocole, les ports utilisés, le fonctionnement Formats de messages exécutables, d'entrée et de sortie et exceptions pouvant être envoyées.

- **Structure d'un document WSDL** La figure suivante montre la structure d'un document WSDL :

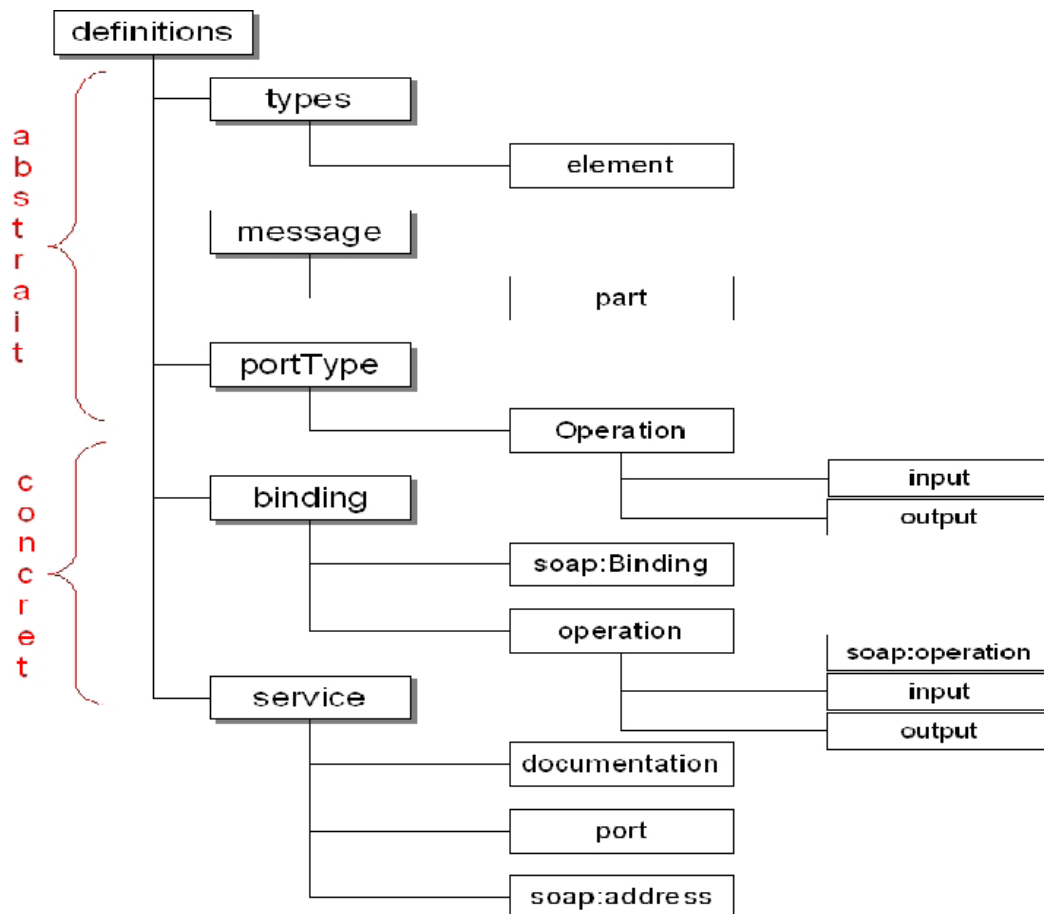


FIGURE 2.4 – Structure d'un document WSDL [41]

<definition> : C'est la racine de tout document WSDL, contient la définition du service. Cette balise peut contenir les attributs précisant le nom du service, et les espaces de nommage. <definitions> contient trois types d'éléments : <message>, <prototype>, <binding> et <service>.

<message> et **<port type>** : Ces éléments définissent les opérations fournies par le service, leurs paramètres d'entrée et de sortie, ... etc. La balise <message> correspond à un paramètre d'entrée ou de sortie de <operation>. La balise <portType> définit un ensemble d'opérations. Une <operation> définit des messages d'entrée /sortie. Par exemple, dans Java, une méthode (opération) et une interface (portType).

<binding> : Cet élément relie les <portType> à un protocole particulier, et on peut définir un binding pour chaque protocole supporté. Les bindings possibles sont SOAP, DCOM ou CORBA. Actuellement seul SOAP est utilisé.

<service> : Cet élément identifie les informations supplémentaires nécessaires pour appeler le service, en particulier l'URI du destinataire. Un < service > est conçu comme un

ensemble de ports, un < port > étant lié à < binding > pour l'URI. Il est également possible d'identifier des types de données complexes dans la balise < types > immédiatement avant la balise < message > . Enfin, chaque élément WSDL peut être documenté à l'aide du < document > . Ce composant contient des informations relatives à la compréhension du document par les utilisateurs du service.

2.3.6.4 UDDI : Universal Description Discovery and Integration

Un des principaux avantages de l'adoption des services Web par les entreprises est le partage de services sur Internet ou dans un Intranet. Le partage de services Web permet le développement plus rapide d'applications, soit un gain temps pour l'entreprise ainsi qu'un gain d'argent. L'élément clé pour ce partage des services est l'annuaire UDDI [41] .

L'objectif primaire d'UDDI [40] est la spécification d'un canevas pour décrire et découvrir des services Web. Le noyau d'UDDI travaille avec la notion de "business registry", qui est un service sophistiqué de noms et répertoires. Plus précisément, UDDI définit des structures de données et APIs pour publier les descriptions des services dans le registre et pour interroger ce registre afin de chercher des descriptions publiées. Parce que les APIs d'UDDI sont aussi spécifiés en WSDL avec une attache SOAP, le registre peut être invoqué comme un service Web (en conséquence, ses caractéristiques peuvent être décrites dans le même registre, comme un autre service).

La spécification de registre UDDI a deux objectifs principaux en termes de découverte de services : premièrement, permettre aux développeurs de découvrir des informations sur les services. Deuxièmement, activer la liaison dynamique, permettant aux clients d'interroger le registre et d'obtenir des recommandations pour les services qui les intéressent.

2.3.6.5 WS-BPEL : Web Services Business Process Execution Language

WS-BPEL est un langage de procédés basé sur la technologie XML, tout comme les autres standards des services Web. WS-BPEL permet de construire des procédés interprétables et exécutables par un moteur d'orchestration [25]. Les procédés peuvent être modélisés de deux manières :

abstraite : à ce niveau, seuls les échanges de messages entre les différents participants sont spécifiés. Mais le comportement interne de ces participants n'est pas explicité. [41]

exécutable : les activités du procédé sont ordonnées, les partenaires impliqués sont identifiés ainsi que les messages qui sont échangés. A ceci s'ajoute le traitement des fautes et des exceptions pour les cas d'erreurs [41].

WS-BPEL vise à modéliser le comportement des processus à travers un langage de spécification de processus opérationnels exécutables et abstraits. Ce faisant, il étend le modèle

d'interaction des services Web et lui permet de prendre en charge les transactions commerciales. Il définit également un modèle d'intégration interopérable qui devrait aider à étendre l'intégration des processus d'automatisation au sein et entre les entreprises. Son développement est venu de la notion que la programmation dans le grand et la programmation dans le petit exige différents types de langues.

En tant que tel, il est sérialisé en XML et vise à permettre la programmation dans le grand.

2.3.6.6 REST : Representation State Transfer

REST (Representational State Transfer) est une architecture de services Web. Élaborée en l'an 2000 par Roy Fielding, l'un des créateurs du protocole HTTP, du serveur Apache HTTPd et d'autres travaux fondamentaux, REST est une manière de construire une application pour les systèmes distribués comme le World Wide Web.

Le principe de REST est d'utiliser HTTP pour La mise en œuvre d'un Web Service n'est plus un simple protocole de transport, mais définit l'API (Application Programming Interface) de chaque service, c'est-à-dire la définition des messages entre le client et le serveur.

• Avantages de REST

Les avantages sont les suivants :

- Utilisation d'HTTP comme protocole applicatif et non protocole de transport ;
- cache réseau → en respectant les entêtes et les requêtes préconisés dans la norme HTTP, on permet ainsi l'utilisation efficace de serveurs caches entre le serveur et les clients de l'application ;
- interface uniforme → chaque Web Service REST a une interface orientée autour des messages de HTTP.[15]

•Principes de REST

1• Adressage

L'adressage est l'idée que tous les objets et les ressources de votre système est accessible par un identifiant unique.

2• Interface

Une des caractéristiques de REST est l'interface limitée,l'idée sous-jacente est que nous adhérons à un ensemble limité d'opérations pour le protocole d'application de votre service de distribution. Cela signifie que nous n'utilisons que des méthodes HTTP pour les services web. HTTP a un petit ensemble fixe de méthodes opérationnelles :

a)GET : Le but de la commande GET est de récupérer une représentation d'une ressource.

b)POST : La méthode POST permet de crée une nouvelle ressource sur le système.

c)PUT : Cette méthode n'est utilisée pour créer de nouvelles ressources que lorsque le client contrôle une partie de l'URI.

d)DELETE : La méthode DELETE supprime une ressource. Le serveur doit renvoyer une réponse indiquant le succès ou l'échec de l'opération.

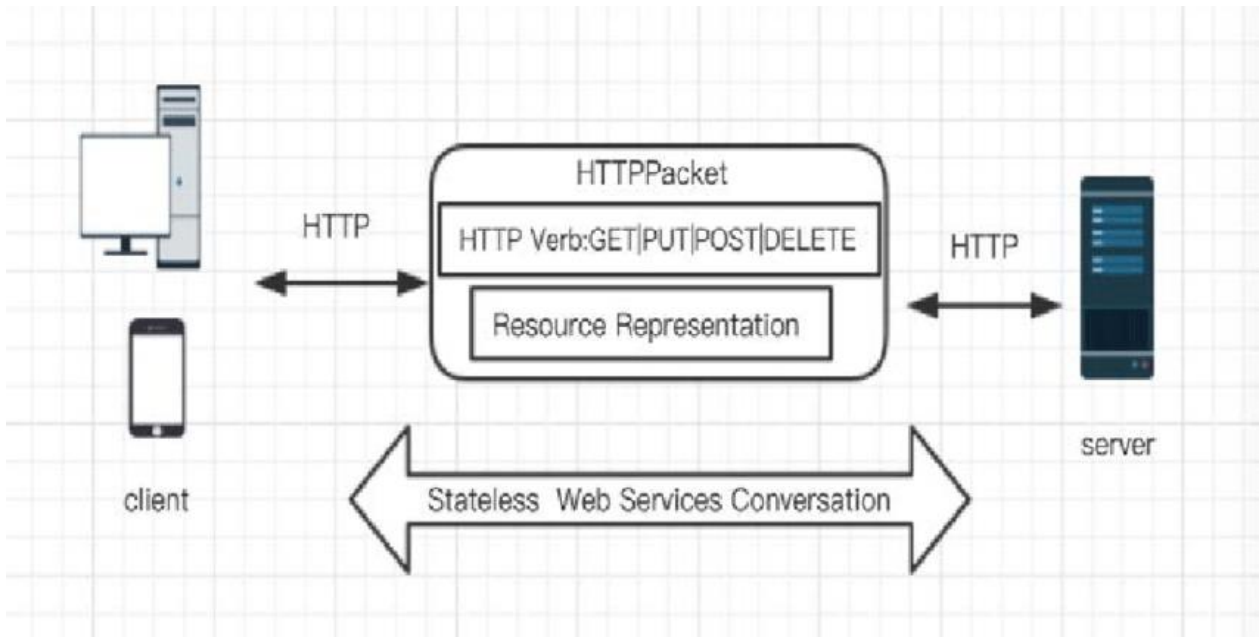


FIGURE 2.5 – REST web service Architecture [17]

REST VS SOAP :



FIGURE 2.6 – REST web service Architecture [13]

Selon cette architecture, nous concluons que l'architecture REST est plus facile à utiliser. REST est plus facile à mettre en œuvre et est utilisé par les web services pour gérer les ressources distantes avec toutes les phases classiques, créer, restaurer, modifier et supprimer.

2.3.7 Fonctionnement des services Web

Le fonctionnement des services Web s'articule autour de trois acteurs principaux illustrés par le schéma suivant :

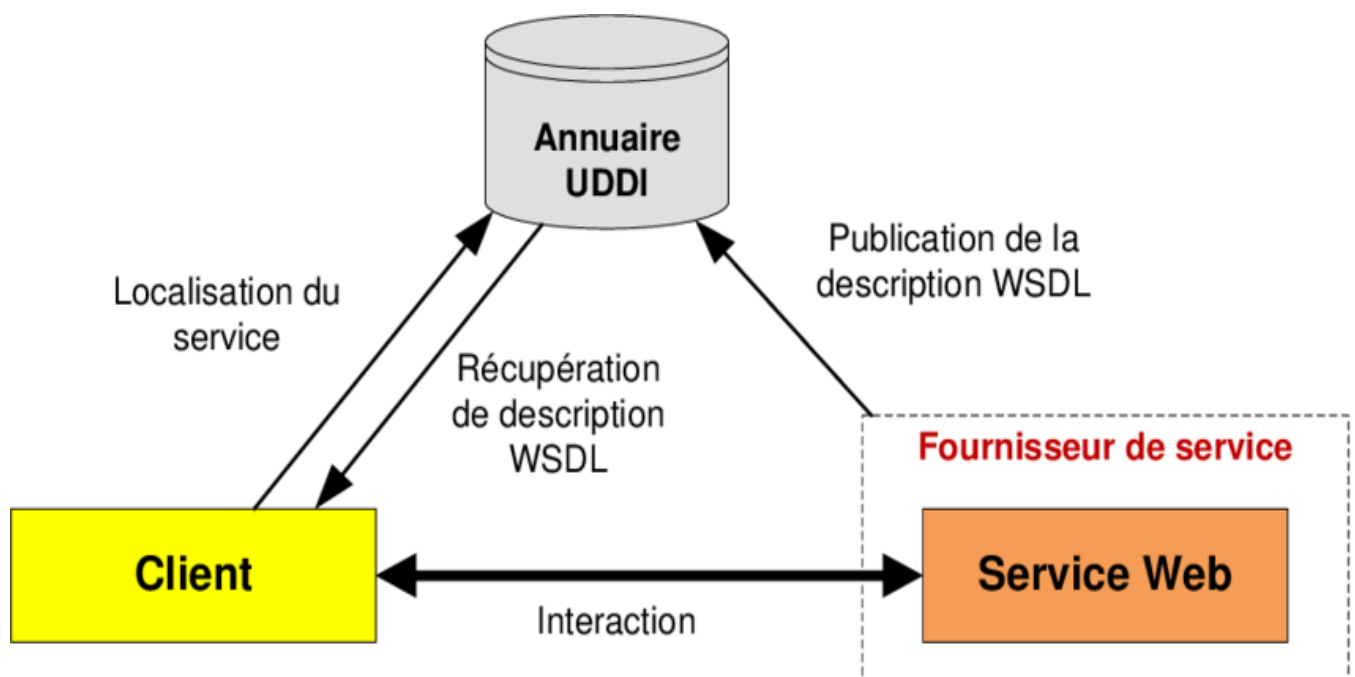


FIGURE 2.7 – Le fonctionnement des services Web [5]

Fournisseur de service

Le fournisseur de service met en application le service Web et le rend disponible sur Internet.

Service requester programme client

C'est n'importe quel consommateur du service Web. Le demandeur utilise un service Web existant en ouvrant une connexion réseau et en envoyant une demande en XML (REST, XML-RPC, SOAP).

Annuaire service registry

Le registre de service est un annuaire de services. Le registre fournit un endroit central où les programmeurs peuvent publier de nouveaux services ou en trouver. Les interactions entre ces trois acteurs suivent plusieurs étapes :

1-La publication du service : le fournisseur diuse les descriptions de ses services Web dans l'annuaire.

2-La recherche du service : le client cherche un service particulier, il s'adresse à un annuaire qui va lui fournir les descriptions et les URL des services demandés afin de lui permettre de les invoquer.

3-L'invocation du service : une fois que le client récupère l'URL et la description du service, il les utilise pour l'invoquer auprès du fournisseur de services.

Le Tableau 1 récapitule les informations concernant la technologie des services Web en fonction des caractéristiques importantes de l'approche à services.

	Services web
spécification	- interface WSDL - Des propriétés non-fonctionnelle
implantation	- Nombreux langages - Génération des talons clients
Découverte	- UDDI - Découverte active
Composition	- oui, orchestration WS-BPEL
Communication	- SOAP sur HTTP - synchrone
Liaison	- type : direct - statique

TABLE 2.1 – Récapitulatif de la technologie des services Web.

2.3.8 Invocation d'un service web

Les étapes les plus importantes de l'invocation d'un service web sont [11] :

1- Le fournisseur (service) est responsable de l'enregistrement et de la publication des services sur le serveur UDDI. Le processus est effectué en envoyant un message (encapsulé dans l'enveloppe SOAP) à l'annuaire UDDI. Ce message compile l'emplacement du service, la méthode d'invocation (et les paramètres associés) ainsi que la forme de réponse. Toutes ces informations seront ensuite formalisées à l'aide de WSDL.

2- Un utilisateur qui souhaite consulter le service s'informe d'abord au sujet d'un serveur UDDI dont l'adresse est connue pour demander quels services sont disponibles qui sont appropriés à ses besoins. Le serveur renvoie la liste des possibilités à partir de laquelle on choisit. À ce stade, l'utilisateur n'a que URL4 spécifier le service sélectionné.

3. L'utilisateur récupère ensuite l'interface WSDL, accessible à partir de l'URL et permettre à l'utilisateur de savoir comment utiliser le service. À partir de cette interface, il peut générer automatiquement un service "proxy." Puisqu'il s'agit d'un objet local, il exécute les mêmes fonctions de service à distance et permet à l'utilisateur d'accéder au service à distance en toute transparence. "Proxy" est créé à l'aide d'un outil et peut être généré dans un grand nombre de langages de programmation différents. Le service est utilisé simplement sur la base d'une méthode "proxy" qui correspond aux besoins de l'utilisateur.

4. Le proxy représente l'appel de la méthode distante sous forme d'une requête SOAP où seront inclus les paramètres fournis par l'utilisateur. Ces paramètres seront empaquetés grâce à la méthode standard de présentation des données, ce qui permet d'assurer la compatibilité entre machines. Cette requête est ensuite émise vers l'URL désignant le service web.

5. Sur la machine hébergeant le service, la requête est réceptionnée puis ouverte par un Stub.

6. Une fois la requête est comprise, une réponse SOAP est construite puis émise en direction de l'expéditeur initial.

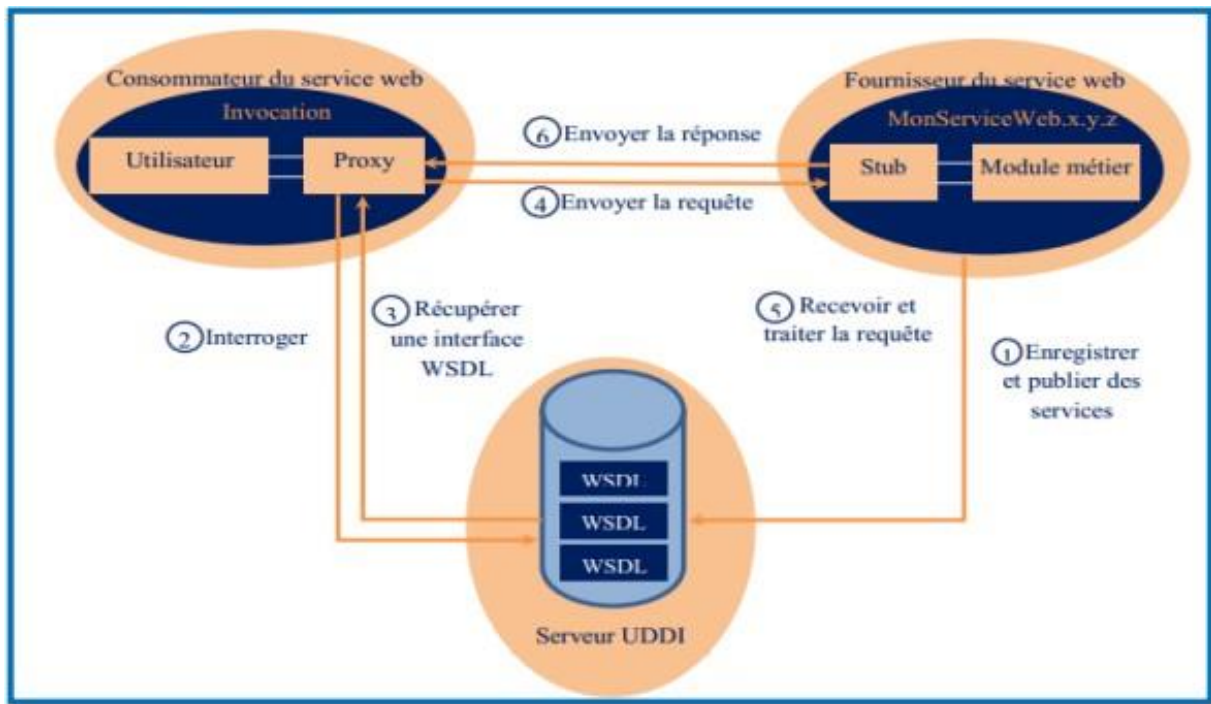


FIGURE 2.8 – Les étapes d’invocation d’un service web [11]

2.3.9 Les applications des Services web

Les applications des services Web sont diverses, tant dans les domaines B2C, B2B, que dans le domaine de la gestion, comme la gestion des stocks, la gestion commerciale, etc...

B2C (Business to Consumer) : Qualifie une application, un site Internet destiné au grand public.

B2B (Business to Business) : Qualifie une application, un site Internet destiné au commerce de professionnel à professionnel.[26]

2.4 Sécurité des web services

2.4.1 Vulnérabilités des web services

1. Défis de services (DOS) :

- **Causes :**
 - Le réseau de la machine infectée se connecte au serveur et sature ses ressources ;
 - Attaque l’application et force une boucle infinie ;
 - attaque par paquets IP.
- **Risques :**
 - perte de connexion, perte d’activité.

- **Solution :**

- Revoir la conception des protocoles Internet ;
- Il est nécessaire de pouvoir filtrer efficacement les listes IP contenant des millions d'adresses de manière intelligente.

2. Pirater le réseau privé en envoyant du code malveillant :

- Logiciels mis en frontal sur internet sans avoir été prévu.

3. SQL Injection :

- **Risques :**

- Accès non autorisé, vol de données

- **Causes :**

- Erreur de parsing des requêtes SQL; écrites dans les programmes sous forme de chaînes de caractères et interprétées dynamiquement.

- **Solutions :**

- Validez dynamiquement les requêtes SQL ;
- Compiler les requêtes et les procédures stockées.

2.4.2 Quelques protocoles de Sécurisation des web services :

Authentification de différents utilisateurs en utilisant différentes méthodes :

- Le login et mot de passe (le moins sécurisé) ;
- Kerberos : le protocole d'authentification réseau est Mécanisme de clé privée et utilisation de tickets au lieu de mots de passe en clair. Évitez le risque d'interception non autorisée des mots de passe des utilisateurs. Kerberos a d'abord été implémenté sur les systèmes Unix ;
- SAML(Security Assertion MarkupLanguage) :basé sur le format XML utilisé pour définir les informations d'identification et d'autorisation. La spécification prévoit que les informations d'identification peuvent être transmises d'un service Web à un service Web à l'aide de n'importe quel protocole de transport.
Chaque document SAML a un élément XML "request". Les éléments-fils obligatoires sont "AuthenticationQuery" et "Subject".

2.5 Avantages et inconvénients

Les services web ont des avantages et des inconvénients :

Avantages :

Le principal avantage des services Web est qu'ils peuvent communiquer sur une variété de plates-formes. Les clients et les serveurs n'ont pas besoin d'avoir beaucoup en commun pour

communiquer. Pour atteindre ce dernier, les services Web utilisent un format standardisé que tous les systèmes peuvent comprendre. De plus :

- Les services Web utilisent des protocoles ouverts et standards (basés sur le protocole HTTP);
- Les services Web peuvent fonctionner à travers de nombreux pare-feux sans modifier les règles de filtrage ;
- Les protocoles et les formats de données sont aussi textuels que possible pour faciliter la compréhension du fonctionnement global des échanges ;
- Les outils de développement basés sur ces normes permettent la création automatique de programmes à l'aide de services Web existants.

Inconvénients :

Cependant, un des inconvénients vient de ces formats. XML est un format assez volumineux qui produit un grand nombre de paquets. Cela peut causer des problèmes avec des connexions réseau lentes. Une autre possibilité de connecter les deux systèmes ensemble via Internet consiste à utiliser une API Web. Ces interfaces sont également accessibles depuis Internet. Ils sont généralement plus rapides, mais fixent des objectifs plus clairs pour les clients et les serveurs, ce qui limite l'interopérabilité.

- Les normes de services Web dans certains domaines sont actuellement à jour ;
- Par rapport à d'autres méthodes approches de l'informatique répartie telles que RMI, CORBA ou DCOM, les performances des services Web sont faibles ;
- Lors de l'utilisation du protocole HTTP, le service Web peut contourner les mesures de sécurité définies par des pare-feu.

2.6 Conclusion

Les services Web ne sont que des transactions accessibles en échangeant des documents XML entre deux URL. Ce dernier offre une réelle souplesse d'utilisation et permet d'accélérer le développement des applications. En adoptant les services Web, nous favoriserons l'interopérabilité et la réutilisation du code, mais cela ne suffit pas pour assurer une automatisation complète de l'interopérabilité. De plus, les aspects sémantiques sont pris en compte lors de la conception d'une description de service web.

Aux termes de ce chapitre, nous avons introduit le domaine des services Web en présentant leur architecture, leur principe de fonctionnement et leurs standards de base, et nous avons illustré quelques moyens fondamentaux qui participent au développement et à la sécurisation des web services, et à la fin, nous avons mentionné quelques avantages et inconvénients des-dit services Web.

E-santé (E-health)

3.1 Introduction

Au cours des dernières années, les méthodes de communication et les technologies de l'information et de la communication (TIC) ont continué à progresser de plus en plus dans les systèmes de santé afin de fournir des soins de qualité accessibles au plus grand nombre de patients possible, et les professionnels de la santé associés aux ingénieurs techniques ont réussi à combiner les TIC avec la médecine qui a mené à la télémédecine. Ainsi, la croissance des réseaux de la santé électronique a été rapide, dans le but de répondre à plusieurs exigences telles que l'amélioration continue de la qualité des soins, l'amélioration de la gestion des patients, la modernisation des établissements de santé, et la technologie pour établir et diffuser une communication plus personnalisée et humaine entre l'hôpital et les patients... etc.

La téléconsultation est maintenant présentée comme une nouvelle forme de réglementation de la pratique médicale qui sert le cheminement du patient. C'est un moyen potentiel d'améliorer l'accès aux soins pour les populations isolées, les personnes âgées et les handicapés, en fournissant des soins à proximité des lieux de vie des patients. Il s'agit également d'une consultation pour un patient effectuée par une télémédecine. L'adoption de cette nouvelle pratique médicale est motivée par le développement de plateformes plus efficaces que jamais. La consultation à distance est un élément clé de l'amélioration du système de santé et répond immédiatement aux principaux enjeux de la pratique médicale, comme l'amélioration de la qualité de la gestion et du suivi des patients, la réduction des coûts, etc.,

La télémédecine regroupe toutes les pratiques médicales qui permettent la gestion à distance des patients grâce aux technologies de l'information et des communications en santé et la santé électronique (E-santé).

3.2 Concepts généraux sur E-santé

3.2.1 Définition :

La e-santé, ou santé électronique, décrit l'ensemble des moyens et services liés à la santé qui utilisent les nouvelles technologies de l'information et de la communication. La e-santé fait appel à Internet, aux applications pour smartphones et aux objets connectés.

Le terme e-santé regroupe ainsi un ensemble de techniques de pointe que le Conseil National de l'ordre des médecins divise en plusieurs catégories : la télé santé, la télémédecine, la m-santé (m pour « mobile », comprend l'ensemble des objets connectés et des applications mobiles) et la robotique³. Le terme Santé est à prendre au sens large et comme l'OMS le souligne, ne concerne pas que les maladies et l'homme malade mais est aussi relatif à un état complet de bien-être physique, mental et social. Le développement de l'e-santé s'appuie sur un domaine scientifique particulier, l'informatique médicale (ou informatique de santé), domaine qui a des liens étroits avec l'informatique mais dont les problématiques sont spécifiques du domaine santé. [12]

3.2.2 Historique de l'E-santé :

Malgré une démocratisation récente, la e-santé est un phénomène ancien. La "révolution numérique" a commencé dans les années 1970, et le marché industriel de la santé numérique a émergé dans les années 1990, un changement radical de la médecine traditionnelle qui a donné accès à l'information de santé. L'essor d'Internet dans les années 2000, le développement de politiques publiques ambitieuses dans les années 2010, Mais depuis 2020, c'est la e-santé qui a fait les progrès les plus remarquables. En effet, la crise sanitaire a dynamisé ce secteur face à des enjeux qui nécessitent plus que jamais des solutions digitales. Dès lors, les soins médicaux et à distance sont invités à des applications pour aborder la question de la continuité des soins et montrer l'évolution déterminante de la pratique médicale.

3.2.3 Pour qui la e-santé ?

La e-santé est de plus en plus populaire dans la société. Pour répondre au vieillissement de la population et même pour augmenter les dépenses de santé, la e-santé se développe à grande vitesse. Les patients et les professionnels de la santé utilisent de plus en plus les applications de santé en ligne dans leur vie quotidienne. En vue d'atteindre :

- Evolution de l'opinion des professionnels de santé (changement d'avis) : téléconsultation et téléspécialiste ou visioconférence, parfois pluridisciplinaire.
 - Assistance à distance, principalement par le diagnostic et le conseil thérapeutique, pour les patients locaux pauvres : Assistance à distance.
 - Réalisation entièrement et exclusivement à distance des actes médicaux : télédiagnostic, téléchirurgie, etc.
 - Organiser la circulation des données dans un réseau de santé : cyber-réseaux de santé.
- Quelle que soit l'application, les concepts fondamentaux inhérents à la télémédecine ne sont plus aller vers l'information, mais déplacer l'information et la rendre disponible quand et où elle est nécessaire.
- Par conséquent, l'application correspondante est conçue pour connecter :

- Le médecin avec son patient ;
- Le médecin avec un (ou plusieurs) autre(s) médecin(s) ;
- Le médecin avec un (ou plusieurs) autre(s) professionnel(s) de la santé.

3.2.4 Dossier patient informatisé

Lorsqu'un patient va consulter un médecin ou bien est hospitalisé, ses plaintes, ses symptômes, ses signes cliniques et biologiques, ses images médicales, ses traitements et d'autres données encore sont saisies dans ce qui est appelé son dossier médical. Le développement de l'informatique et des réseaux fait que, de plus en plus, ce dossier médical tend à être informatisé. À l'hôpital, après chaque séjour d'un patient, un compte rendu d'hospitalisation est élaboré et stocké dans son dossier. L'informatisation du dossier médical en facilite la communication aux différents professionnels de Santé habilités à le consulter. Lorsque les données du patient sont enregistrées de manière structurées et codées, elles peuvent être traitées afin de générer des alertes notamment lors de la prescription médicamenteuse envisagée plus loin.[3]

3.2.5 Technologies et Applications :

La e-santé inclut :

La télésanté : qui comprend les actes de prévention et de soins réalisés à distance : information via des portails grand public, sites de promotion de la santé, systèmes d'alerte téléphonique, prescriptions électroniques à distance...

La télémédecine : (actes médicaux réalisés à distance par un médecin) : consultation par vidéoconférence, téléassistance d'un médecin lors d'une intervention, télésurveillance du patient, télé-expertise (échange des avis des médecins)...

La m-santé (mobile santé) : qui comprend les applications numériques pour smartphones ou objets connectés (bracelets...) en lien avec la santé. Les systèmes de dossiers électroniques de santé.

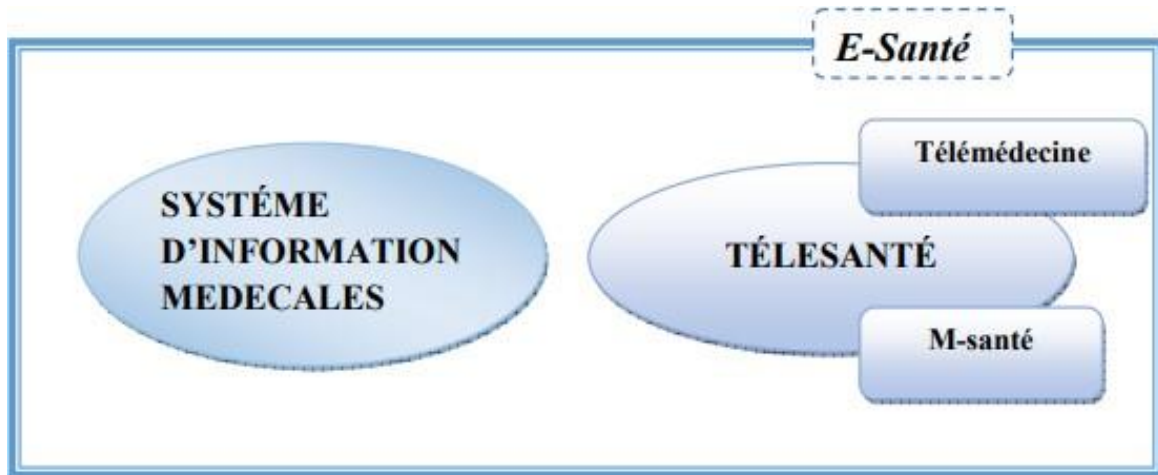


FIGURE 3.1 – different domaine de E-santé [4]

3.3 Cadre conceptuel des TIC en santé

3.3.1 Définition des TIC

Le terme "technologies de l'information et des communications" transcrit une locution anglaise et est utilisé dans divers forums internationaux qui sont presque présents dans le domaine des technologies de l'information à distance. Il a des définitions différentes selon la perspective ou le temps des auteurs, en raison de la confusion progressive des frontières des zones concernées et le développement rapide des technologies avec la convergence numérique.

Le dictionnaire Larousse définit les technologies de l'information et de la communication comme étant un « ensemble des techniques et des équipements informatiques permettant de communiquer à distance par voie électronique (câble, téléphone, Internet, etc.). » [20]

La définition des TIC reste particulièrement vague : le terme "discours technique" est utilisé au lieu du terme "technique", qui serait plus simple et plus précis. Les technologies de l'information et de la communication (TIC) sont des outils de soutien pour le traitement de l'information et de la communication, le traitement de l'information, la survie de l'information et la technologie.

3.3.2 Evolution des TIC

L'historique des TIC remonte après l'invention de l'écriture, les premiers pas vers une société de l'informatique ont été marqué par le télégraphe électrique, le téléphone et la radiotéléphonie alors que, la télévision, l'Internet puis la télécommunication mobile et le GPS ont associé l'image au texte et à la parole. L'Internet et la télévision devenant accessibles sur le téléphone portable qui est aussi appareil photoLa convergence entre l'informatique et les communications dans la dernière décennie du XXe siècle a bénéficié de la miniaturisation des composants, permettant la production d'appareils "multifonctionnels" abordables. Depuis les années 2000, l'utilisation des TIC

n'a cessé de croître. C'est particulièrement vrai dans les pays riches. Les TIC sont appliquées dans tous les domaines tels que l'agriculture, la biodiversité, le commerce, la télémédecine et la gestion de l'information pour de multiples bases de données, la robotique et les utilisations militaires.

Le terme TIC est une invention des ingénieurs réseaux désigne Technologies de l'information et de la communication; elles sont également désignées par les « nouvelles technologies de l'information et de la communication » (NTIC). Les TIC sont essentiellement des moyens au service de l'apprentissage (Tardif, 1998). Dans l'éducation, elles évoquent plutôt les technologies de l'information et de la communication pour l'éducation (TICE). En matière d'évolution technologique, les TIC sont le résultat de la convergence de trois technologies : l'informatique, les télécommunications, et l'audiovisuel. Ces trois domaines sont associés dans l'ordinateur connecté. Internet est venu concrétiser la convergence informatique

L'enseignement des TIC fait référence à la connaissance et à l'utilisation d'un outil informatique qui peut être utilisé dans nos nombreuses tâches quotidiennes. C'est aussi la formation des jeunes instruits dans la société de demain pour pouvoir utiliser les diverses technologies de l'information et de la communication, ainsi que l'information et la communication dans le monde entier. Cela sera réalisé par les enseignements qu'ils recevront dans la juridiction de l'ordinateur.

3.3.3 TIC dans le domaine de la santé

Comme le secteur de la santé n'était pas différent des autres secteurs, l'utilisation des TIC a été une solution à de nombreux problèmes.

Dans cette section, nous tenterons d'identifier les pratiques des TIC dans le domaine de la santé.

3.3.3.1 Genèse des TIC en santé

Les réseaux de santé constituent depuis le milieu des années 1980 une tentative de réponse originale aux cloisonnements des systèmes de santé dans certain nombre de pays.

Les enjeux de maîtrise de l'information et d'amélioration des processus de communication sont essentiels dans l'affirmation de ces organisations (établissements de santé) d'interface entre la médecine de ville et l'hôpital. Les espaces innovants de recomposition des positionnements (TIC), ils se construisent progressivement par les interactions entre acteurs, à la fois humains et dispositifs techniques, autour de nouvelles pratiques coopératives, qui réalisent l'amélioration de la coordination entre tous les acteurs est déterminante.

3.3.3.2 La relation entre TIC et la santé

« Les technologies de l'information et des communications (TIC) peuvent aujourd'hui se prêter à de multiples applications dans le secteur de la santé. Elles ont considérablement progressé et l'on s'accorde largement à penser qu'elles peuvent contribuer à améliorer la qualité et la sécurité des soins ainsi que leur adéquation aux besoins des patients, tout en renforçant l'efficacité (des services plus adaptés, une meilleure disponibilité et moins de gaspillage). Les partisans de l'utilisation des TIC dans le secteur de la santé font valoir notamment qu'elle contribuera à réduire le nombre d'erreurs de médication ». [1]

L'Organisation mondiale de la Santé (OMS) définit la cybersanté comme l'utilisation des technologies de l'information et de la communication (TIC) au service de la santé. Le terme "santé numérique" comprend la cybersanté, la santé sur mobile et la télémédecine et renvoie à l'utilisation généralisée des technologies numériques pour les soins de santé et d'autres objectifs de santé dans de très nombreux contextes, dans les établissements de soins comme en dehors. Compte tenu de l'accroissement du nombre de dispositifs, les TIC peuvent jouer un rôle essentiel dans le domaine de la santé, en particulier dans les stratégies de traitement et de prévention. Par exemple, les dispositifs à porter sur soi et les appareils de maintien en forme peuvent désormais jouer un rôle clé en matière de surveillance et de prévention.

3.3.4 Les apports de l'application des TIC en santé

Les progrès de la médecine ont permis au cours du siècle dernier d'éradiquer ou de soigner un grand nombre de maladies graves, mais de nouvelles maladies apparaissent issues de la mutation des éléments pathogènes, devenus plus résistants, des changements de comportements et de l'apparition des risques de la vie moderne liés à l'utilisation de la technologie médicale, ces changements exigent la pratique d'une évaluation totale pour chaque projet de TIC en santé, afin de ressortir ces avantages et ces inconvénients[24]

3.3.4.1 L'évaluation des TIC en santé

Pour calculer ou mesurer l'impact de l'utilisation des TIC en santé, « il existe encore peu d'éléments sur l'impact des TIC de santé. Pour beaucoup d'aspects de l'incidence que peuvent avoir les TIC sur l'efficacité des systèmes de soins de santé, il n'existe guère de données sur lesquelles fonder une quelconque estimation quantitative. Il est tout aussi difficile d'obtenir des chiffres fiables sur le « taux de réussite » des projets ou programmes de TIC de santé »[23] .

«L'évaluation des technologies de santé (Health Technology Assessment) peut être définie quant à elle comme l'analyse, l'agrégation et la synthèse de l'information scientifique sur l'impact des technologies sur la santé ou sur le système de soins. On distingue classiquement quatre principaux axes dont l'ensemble constitue une évaluation type HTA »[27] :

1- L'évaluation technique :

Cet axe a comme objectif de vérifier que la technologie fait ce pour quoi elle a été conçue. Les dispositifs médicaux sont ici un bon exemple pouvant nécessiter l'avis d'un ingénieur biomédical (le médicament constitue un sujet à part avec la problématique de la qualité pharmaceutique, du mécanisme d'action, de tests précliniques). Cette évaluation technique aide aussi à l'implémentation d'un équipement ou d'un dispositif en prenant en compte les contraintes d'installation dans son environnement (ex : robot chirurgical) [14] .

2- L'évaluation médico-économique :

L'évaluation médico-économique des TIC en santé est nécessaire pour justifier certains choix d'allocations des ressources dans un contexte de progression des dépenses de santé. Elle repose, soit sur les méthodes traditionnelles d'évaluation en économie de la santé (coûts/avantages), soit sur des approches nouvelles « multicritères » qui ont l'avantage de faire apparaître d'autres bénéfices que les méthodes traditionnelles « uni-critère » ne prennent pas en compte. La majorité des études sur la télémédecine ont utilisé la méthode de minimisation des coûts. Cette méthodologie est insuffisante car elle considère que les deux options étudiées (l'une avec télémédecine, l'autre sans) sont strictement identiques du point de vue des résultats et le problème se réduit alors à la seule comparaison des coûts. Il faut en fait utiliser une méthode qui évalue la totalité des bénéfices apportés par un projet de télémédecine. L'analyse ne doit pas porter sur le coût de la télémédecine, mais sur la prise en charge assistée par télémédecine par comparaison avec une prise en charge conventionnelle sans télémédecine (diagnostic traditionnelle).[21]

L'évaluation ne se fera seulement sur le calcul de son impact sur l'investissement, mais aussi sur :

- La réduction des coûts de financement ;
- La réduction des erreurs médicales ;
- La réduction des maladies nosocomiales ;
- L'augmentation de nombre des personnes traitées ;
- L'amélioration de la qualité effectuée, et le degré de l'accessibilité réalisée. . .

3- L'évaluation médicale :

L'évaluation médicale est la méthode la plus importante dont le processus de l'évaluation des TIC en santé, par laquelle on peut connaître l'efficacité des outils des technologies de l'information et de communication en santé, et leur degré d'efficacité. vise tout d'abord à apporter une connaissance sur le rapport intrinsèque entre bénéfices et risques d'une technologie de santé. Elle permet ensuite de connaître le progrès d'une technologie par rapport aux alternatives (notions équivalentes : efficacité relative, valeur thérapeutique ajoutée), dans les conditions théoriques de l'expérimentation (Relative Efficacy) ou en pratique réelle (Relative Effectiveness).

4- L'évaluation de « l'acceptabilité sociale » :

Avant l'installation d'une technologie en santé, il faut prendre en compte les composantes éthiques, juridiques, traditionnelles, de la société, et leur acceptabilité de cette technologie. « La décision d'adapter ou non une technologie de santé doit tenir compte d'éléments scientifiques mais également du contexte, comme les facteurs éthiques, juridiques mais aussi psychologiques de la population. Une fois réalisée, l'évaluation est présentée sous la forme d'un rapport, recommandation ou avis qui devrait refléter la connaissance dans les quatre domaines. Lors de son écriture, il est essentiel de ne pas négliger la rédaction et la présentation des résultats : le rapport d'HTA doit être lisible, clair, complet et informatif, bien référencé, mais également concis si on veut qu'il soit lu et compris par les décideurs » [27]

3.4 La télésanté

Regroupe notamment la télémédecine et la m-santé.

3.4.1 La télémédecine

Pour l'Organisation mondiale de la santé (OMS), la télémédecine est une composante de la médecine. Elle s'inscrit dans l'univers des professionnels de l'e-santé, « permet d'apporter des services de santé à un être numérisé vivant, là où la distance, l'isolement du patient, sont un facteur critique pour sa santé, par des professionnels (médecins notamment) utilisant les TIC pour diagnostics, e-traitement, e-prévention, e-recherche, e-éducation, e-formation continue »[10]. Selon le code de santé publique (art. L.6316-1), elle est « une forme de pratique médicale à distance utilisant les technologies de l'information et de la communication »[9].

Définie comme :« la télémédecine est une forme de pratique médicale à distance utilisant les technologies de l'information et de la communication. Elle met en rapport, entre eux ou avec un patient, un ou plusieurs professionnels de santé, parmi lesquels figure nécessairement un professionnel médical et, le cas échéant, d'autres professionnels apportant leurs soins au patient. » [34]

La télémédecine ou bien la médecine à distance est l'une des pratiques des nouvelles technologies médicales, et son émergence est une étape importante pour résoudre le grand nombre de difficultés et de limites complexes qui retardent l'amélioration des systèmes de santé dans le monde. Elle est considérée comme une meilleure solution.

3.4.1.1 La télémédecine en Algérie

Au cours des prochaines années, l'émergence de ces nouvelles technologies de l'information et le déploiement à grande échelle des réseaux de télécommunications auront un impact significatif sur le comportement des individus et des groupes. La télémédecine, en particulier, a le

potentiel de modifier considérablement la pratique médicale et l'organisation de la population.

L'Algérie est le plus grand pays d'Afrique, d'une superficie de 2.381.741 km², divisé en 58 wilayas et 1541 communes.

L'Algérie doit prendre en compte les facteurs qui peuvent bénéficier au développement d'e-santé dans un pays aussi vaste que le nôtre, où les besoins de santé augmentent avec la croissance démographique. Les besoins évolutifs ou les populations ont tendance à augmenter, de même que leur sous-répartition entre le nord et le sud. Cela obligera les institutions publiques à mettre en œuvre de nouvelles politiques qui nécessitent l'utilisation des nouvelles technologies médicales comme solution à leurs besoins. Cela donne à l'Algérie l'opportunité de faire un investissement privilégié dans les ressources humaines et matérielles et doit être soutenu de manière durable. Assurer l'accès aux services que les usagers représentent à tous les niveaux.

En Algérie, l'introduction de la télémédecine est encore à ces débuts. Le réseau de « Santé-Algérie », opérationnel depuis janvier 1999. La première étape du démarrage effectif d'un projet pilote de télémédecine remonte à 2008. Le 11 août 2011, à Alger, le ministre de la Santé et de la Population Réforme hospitalière a officiellement approuvé le deuxième projet de télémédecine entre le CHU et l'hôpital Laghouat à Bab ElOued (Maillot). Les pratiques de télémédecine sont largement appliquées à d'autres hôpitaux universitaires avec un contrat signé de trois ans. CHU de Sétif en 2014, CHU de Batna en 2015, CHU de Tlemcen en 2015, CHU de Tizi Ouzou en 2016. [14]

3.4.1.2 Les différents types d'application de la télémédecine

L'informatique est un composant fondamental de la télémédecine lorsque l'information entre le patient et le médecin transite via des moyens logiciels, matériels et réseaux. La télémédecine assistée par des moyens informatiques est la solution la plus prometteuse en termes d'efficacité de l'acte :

- l'information est accessible quasiment en temps réel ;
- le volume de données échangées peut être suffisant pour inclure des photos, voir la vidéo temps réel ;
- L'aspect vidéo temps réel, c'est-à-dire avec une faible latence dans la communication, réduit la sensation de distance avec le médecin et rassure le patient.

Aujourd'hui, de nombreux domaines d'application et de services en télémédecine dans le domaine.



FIGURE 3.2 – Plateforme de service de la télémédecine[36]

. Télésurveillance :

Si les données sont stockées, les paramètres physiologiques peuvent être surveillés à distance en temps réel ou de manière différée. elle Il peut être utilisé pour traiter des patients ou pour suivre l'évolution de l'état de santé de personnes présentant des risques liés à la maladie ou exposées à des situations critiques.

• Téléconsultation :

Pour les problèmes de santé complexes, rares ou à risque élevé, il peut être très utile d'obtenir un deuxième avis d'un spécialiste.

Une téléconsultation est une consultation à distance d'un patient par un médecin (médecin généraliste ou tout autre médecin spécialiste), auquel cas le patient peut ou non avoir accès à d'autres professionnels de santé (ex : médecin, infirmier, pharmacien...).

• Télédiagnostic :

Il est possible d'assurer une aide au diagnostic à distance par des spécialistes et est connu dans divers domaines (électrocardiogramme, dermatologie à distance, endoscopie à distance, etc.). Dans ce cas, l'examen médical doit être effectué par un médecin présent. Les patients qui bénéficient de cette application peuvent se trouver en milieu rural où il n'y a pas un centre spécialisé.

- **Téléassistance :**

Elle a pour objet de permettre à un professionnel médical d'assister à distance un autre professionnel de santé au cours de la réalisation d'un acte. » La téléchirurgie entre dans ce cadre : un expert peut guider à distance un chirurgien moins expérimenté ou réaliser lui-même l'intervention par robotique. La première chirurgie à distance, baptisée opération Lindbergh, a ainsi été effectuée en septembre 2001 : un chirurgien français, le Professeur Jacques Marescaux, a réalisé de New York la cholécystectomie d'une patiente se trouvant à Strasbourg.

- **Télé-chirurgie :**

L'exploitation et la manipulation des équipements médicaux contrôlée à distance par le praticien sur le patient, afin d'optimiser les résultats ; la télé chirurgie est également utilisée pour la formation des jeunes chirurgiens.

- **Télé-expertise :**

La télé-expertise a pour but de permettre à un professionnel de santé de solliciter à distance l'avis d'un ou de plusieurs professionnels médicaux en raison de leurs formations ou de leurs compétences particulières, sur la base des informations médicales liées à la prise en charge d'un patient.

- **Télé-formation :**

Utilisation de dispositif informatique en particulier pour l'aide à la formation continue des médecins : contacts professionnels via le réseau, consultation des informations médicales (imagerie, banque de données, d'essais cliniques et suivi d'études épidémiologiques).

3.4.2 m-santé (Mobile-Santé)

La définition la plus communément admise de la m-santé ou mobile-santé est la suivante : « Les pratiques médicales et de santé publique supportées par des appareils mobiles, tels que les téléphones mobiles, les dispositifs de surveillance des patients, les PDA et autres appareils sans fil ». [39] Selon l'OMS, la m-santé prend en compte toutes « les pratiques médicales et de santé publique reposant sur des dispositifs mobiles tels que les téléphones portables, tablettes, systèmes de surveillance des patients, assistants numériques personnels et autres appareils sans fil ».

La santé mobile ou mobile health (m-santé) recouvre une gamme large et diversifiée de produits physiques (objets connectés) ou d'applications liées à la santé ou au « bien-être ». Certains de ces produits proposent des conseils personnalisés, collectent des données personnelles

(poids, tension artérielle, fréquence cardiaque, etc.), fournissent des informations médicales, etc. Ces produits peuvent diffuser des messages de santé publique, améliorer la prévention et même soutenir les soins médicaux. Ainsi, m-santé s'adresse à la fois aux citoyens (utilisateurs du système de santé ou grand public) et aux professionnels de santé.

Les interventions de santé mobile, telles que les applications pour smart-phone basées sur SMS et à plusieurs composants, les sites Web optimisés pour les mobiles et les interventions sur les réseaux sociaux, fournissent une plate-forme facilement accessible et adaptable pour soutenir l'observance du traitement.

3.5 Acteurs de la E-santé

Les acteurs concernés en santé sont nombreux, parfois cachés ;

- Fabricants de matériel informatique.
- Éditeurs de logiciels non spécifiques puis spécifiques.
- Acteurs de la filière imagerie médicale numérique.
- Éditeurs d' applications web.
- Fabricants de téléphones.
- Éditeurs de logiciels mobile.
- Acteurs de la toile ou Web, FAI fixes puis du web mobile.
- Acteurs historiques : laboratoires de biologie médicale.
- Fabricants de certains objets connectés médecins : ECG, spiromètre, tensiomètre automatique, balance ; ou consultants : GPS, cardiofréquence-mètre, bracelets multifonctions..
- Fabricants de véhicules automobiles, ECG à interprétation automatique.. une grande partie des unités centrales, des calculateurs sont dans des objets dont l'écran n'est pas central. [6]

Professionnels de santé :

- individuellement, puis structures groupes cliniques hôpitaux; La plupart des utilisent quotidiennement des systèmes informatisés dans leur pratique (médecins, chirurgiens, dentistes, pharmaciens, sages-femmes mais aussi les professions paramédicales comme les infirmier(e)s, kinésithérapeutes. . .).
- Les structures leur permettant, ou proposant l'exercice de la télémédecine.

3.6 Possibilités en matière de la E-santé

Les systèmes de santé numérique permettent de transformer fondamentalement les soins de santé et de donner aux patients, aux prestataires de soins, aux gestionnaires et aux décideurs les informations et les outils dont ils ont besoin pour administrer et renforcer les systèmes de santé, dispenser de meilleurs soins et améliorer la qualité des traitements et les taux de survie. Ils sont de nature à élargir l'accès à des soins de qualité et à améliorer la prévention et les résultats pour

les patients, y compris pour les patients souffrant de maladies chroniques telles que les maladies non transmissibles.

La santé numérique offre des possibilités et des avantages importants lorsqu'il s'agit [28] :

- 1- D'améliorer la santé publique et les établissements de soins de santé, par exemple en ce qui concerne les procédures hospitalières, les dossiers médicaux électroniques et les renseignements de santé. Le fait que les prestataires et professionnels des établissements de soins échangent rapidement des renseignements et des données permet de faciliter l'accès à des soins de qualité.
- 2- D'aider les praticiens à améliorer les opérations/actes chirurgicaux et les conseils fournis, y compris en ce qui concerne la téléchirurgie. L'utilisation des techniques de santé numérique permet à divers centres de santé éloignés ou fonctionnant par satellite d'utiliser et d'appliquer ces techniques à des fins de diagnostic à distance et de télémedecine. Des hôpitaux en Chine ont par exemple eu recours à la chirurgie à distance, rendue possible par la 5G, pour des opérations du foie et la pose d'implants de stimulation cérébrale profonde utilisés dans le cadre de la lutte contre la maladie de Parkinson.
- 3- D'utiliser des dispositifs de santé individuels et des appareils spécialisés (capteurs, moniteurs, montres-bracelets et téléphones mobiles) à des fins de surveillance et de remontée de l'information. Des téléphones mobiles ont été utilisés pour prendre des images échographiques à distance dans des cas de calculs rénaux ou de grossesse, ainsi que pour effectuer des frottis. Les applications médicales se sont en outre rapidement développées. À l'échelle mondiale, quelque 325 000 applications de santé sur mobile étaient disponibles en 2017.
- 4- D'améliorer la qualité et la précision des analyses et des prévisions concernant les ensembles de données sur la santé, notamment grâce à l'intelligence artificielle, aux mégadonnées et aux simulations faisant appel à la réalité virtuelle. Les données des dispositifs et des capteurs peuvent être regroupées, ce qui permet d'obtenir des images, d'établir des diagnostics et d'analyser des données grâce à l'informatique en périphérie.

Le fait de pouvoir enregistrer des données sur le lieu des soins et prendre des photographies pouvant être analysées à l'aide de l'intelligence artificielle ou par des experts non présents sur le lieu des soins accroît les possibilités d'offrir des soins de santé universels dans diverses spécialités médicales. Les données de santé peuvent également être exploitées pour fournir des soins préventifs ciblés à une communauté. Les possibilités offertes par la santé numérique sont de plus en plus reconnues par les pouvoirs publics - selon l'OMS, à la mi-2018, quelque 120 pays avaient formulé des stratégies de santé numérique, de télésanté ou de cybersanté.

3.7 Enjeux de la E-santé

3.7.1 Bénéfices attendus du développement de la e-santé

La e-santé contribue à l'amélioration du système de soins et représente une réponse concrète et efficace aux défis du système de santé.

3.7.1.1 Patients :

- **Qualité :**

- Des soins plus fiables et coordonnés.
- Moins de bilans.
- Limitant les appels d'urgence déraisonnables.
- Une prise en charge à domicile facilitée (maladies chroniques).

- **Accessibilité :**

- Meilleur accès aux soins dans les zones à faible densité médicale.

- **Proximité :**

- Améliorer la relation entre médecins et patients

3.7.1.2 Établissements de santé et Professionnels du domaine :

- Division du travail entre médecine urbaine, hôpitaux et secteurs médico-sociaux dans une logique de parcours de soins.

- **Qualité :**

- accès à des connaissances médicales validées.
- Travail en équipe et en réseau.
- Ressources de formation pour les travailleurs de la santé.

- **Efficiace :**

- Meilleure utilisation du temps médical pur.
- Outils d'aide à la décision et au diagnostic.

3.7.1.3 Acteurs institutionnels et financeurs publics

- **Efficiace :**

- Une meilleure maîtrise de l'utilisation des ressources médicales.
- Une régulation plus fine de l'économie de la santé

- Potentiel de redistribution plus efficace ou d'économies importantes sur les frais médicaux.
- **Qualité :**
- Renforcer le système de veille sanitaire et améliorer le niveau de prévention.

3.8 Problèmes et risques liés à la E-santé

Les hôpitaux sont des établissements incroyablement complexes, qui mêlent connaissances et compétences médicales complexes et actes infirmiers de base et avancés. Les grands hôpitaux font souvent naître leur propre écosystème dans une zone et sont entourés de toute une série de fonctions et d'entreprises (par exemple l'imagerie, les fonctions de laboratoire, les analyses, les thérapies et les maisons de soins). La mise en place de technologies nouvelles ou complexes a souvent des effets profonds qui nécessitent des ajustements en matière d'emploi, de formation et de personnel. Même s'ils investissent des montants importants dans la santé numérique, les établissements de soins risquent de ne pas en tirer pleinement parti si les autres systèmes et processus ne s'adaptent pas.

Les possibilités offertes par la santé numérique ont toutefois des avantages et des inconvénients importants en ce qui concerne la confidentialité des données des patients et des registres médicaux nationaux. La combinaison des analyses d'ADN et des analyses génomiques permet d'identifier les vulnérabilités biologiques de populations déterminées, et il est désormais possible de prévoir et de pronostiquer l'état de santé futur d'une communauté ou d'une nation à partir des dossiers médicaux et des registres d'informations génétiques dont on dispose actuellement. Le problème que pose l'obtention d'ensembles de données propres et complets doit être réglé, et il faut en outre se pencher attentivement sur la question de la confidentialité des résultats obtenus dans le cadre de la collecte et du stockage des données et les questions relatives au consentement des patients. [?]

3.9 Conclusion

Dans ce chapitre, nous avons introduit les notions fondamentales de e-santé et de télé-médecine. Au début, nous avons défini ce qu'est la e-santé, afin de fournir un rappel historique sur l'émergence et l'évolution de la e-santé. Ensuite, nous avons défini certaines technologies et applications de e-santé, et le Cadre conceptuel des TIC en santé. Ensuite, nous avons cités les acteurs , les enjeux , et les problèmes et risques liés à la E-santé. Tout cela s'inscrit dans le cadre de l'importance et de la nécessité de recourir à la e-santé pour améliorer les soins de santé et réaliser un saut qualitatif dans le domaine de la santé.

Chapitre 4

Ingénierie de domaine

4.1 Introduction

Dans ce chapitre, nous donnerons une description générale de notre application, puis nous mettrons en évidence les aspects conceptuels de l'application, qui constituent une étape fondamentale avant la mise en œuvre, en détaillant les différents schémas et scénarios à mettre en œuvre à l'étape suivante. Apprenez à mieux connaître notre application.

Cette application a pour objectif de mettre en évidence une ligne de produits orientée service pour les plate-formes de Télé-consultation Médicale , pour le développement de la télé-consultation en termes d'accroissement et d'autre part pour l'efficacité et Offrir de meilleurs soins de santé au patient (à distance) et réduire les coûts grâce à des services de bon qualité.

Comme on a déjà présenté dans le chapitre 1, l'ingénierie de domaine vous permet de gérer un ensemble de produits logiciels dans son ensemble, plutôt que comme un ensemble de produits individuels où Chaque produit est traité indépendamment des autres produits.

On peut définir la portée d'une ligne de produits logiciels comme un ensemble de produits planifiés. La définition des membres de la famille de produits prévus permet d'identifier et de planifier les points communs réutilisables et les points de différence entre eux. L'objectif principal de l'ingénierie de domaine est de produire des artefacts réutilisables tels que les exigences, les composants et les classes sont créés pour être réutilisés dans les produits planifiés.

Pour atteindre ces objectifs l'ingénierie de domaine est composée de l'analyse de domaine (feature model), la conception du domaine. Dans la phase de conception de domaine, nous allons désigner les acteurs qui participent à la plateforme, déterminer leurs besoins et les objectifs attendus. Nous décrirons et planifierons ensuite le fonctionnement de notre plateforme en modélisant les différents objectifs à atteindre. Pour la modélisation, nous nous appuyerons sur le langage de modélisation UML (Unified Modeling Language)

Dans ce chapitre on va appliquer ces activités dans le cadre de notre domaine d'étude l'e-santé.

4.2 Analyse de domaine :

Le but de l'analyse de domaine est d'étudier le domaine d'une ligne de produits et d'identifier les similitudes et la variabilité entre les produits.

Pour ce faire, nous étudierons les exigences du domaine. Analyser les exigences pour identifier les exigences communes à tous les produits et les exigences spécifiques à des produits bien particuliers.

Tout d'abord, un domaine est un secteur de métier ou de technologies ou des connaissances qui comprend un ensemble de concepts et de termes que les utilisateurs de ce secteur peuvent comprendre.

Dans notre cas c'est le domaine de la LdP est l'e-santé. Les applications d'e-santé est largement utilisé par les médecins auprès de leurs patients dans les cliniques publiques ou dans les cabinets médicaux , se caractérise par certains points communs et aussi par des aspects spécifiques. C'est ce qu'on appelle la variabilité.

Nous allons utiliser un Feature Diagram pour analyser les similitudes et les différences(variabilités) dans notre ligne de produits. La figure ci-dessous montre un Feature Diagram de notre ligne de produits :

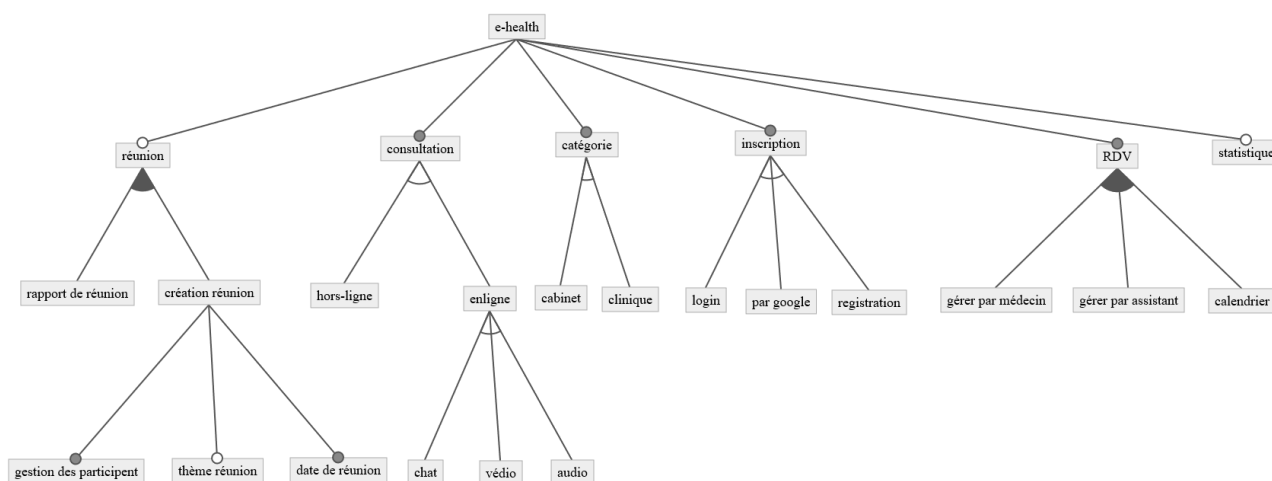


FIGURE 4.1 – Feature modèle métiers -1- pour la ligne de produit e-santé

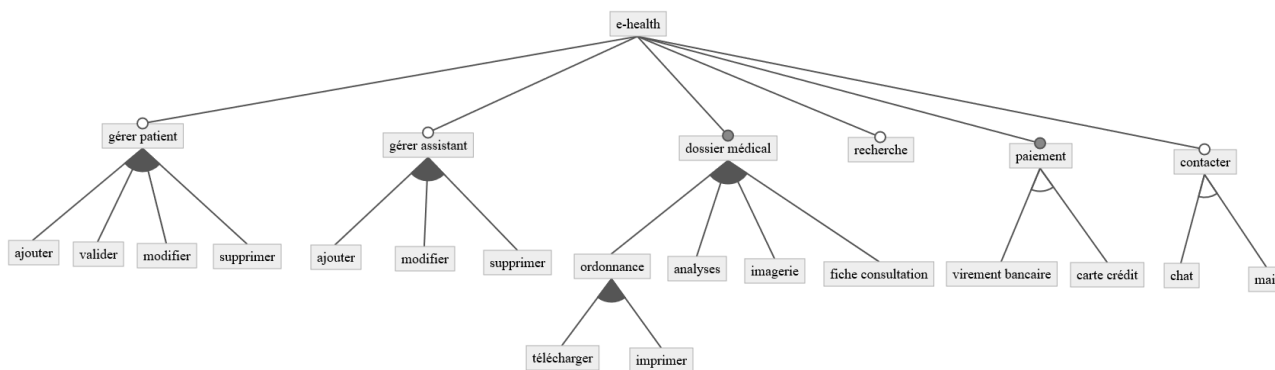


FIGURE 4.2 – Feature modèle métiers -2- pour la ligne de produit e-santé

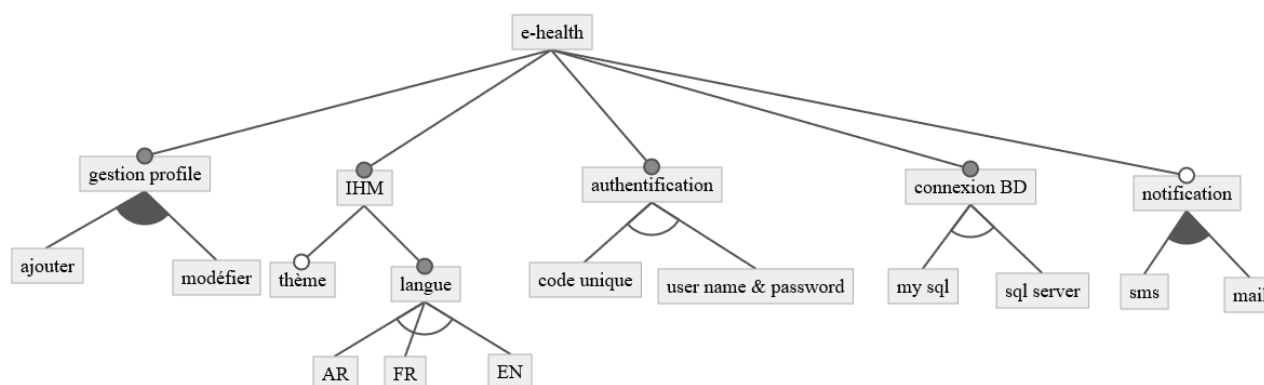


FIGURE 4.3 – Feature modèle technique pour la ligne de produit e-santé

4.2.1 Description du modèle de feature :

Les features qui doivent existés dans tous les membres de la LdP sont représentés par des features Obligatoires et les features qui varient d'un membre à un autre sont représentés par des features Optionnels.

Le modèle de features que nous avons construit se compose de deux diagrammes qui se distinguent par les types des features qu'il comporte :

Le premier diagramme : Diagramme de Features métier -1-, contient les fonctionnalités pour lesquelles la LdP est créée , c'est-à-dire les features liées au cœur du métier de la ligne de produit.Ce diagramme montre qu'une application d'e-santé doit comporter au moins les fonctionnalités : "consultation","ordonnance","catégorie","inscription","RDV","paiement",et "dossier médical". Cependant, dans certain contextes, d'autres fonctionnalités peuvent être exigés par les utilisateurs des applications, tel que : les réunion qui organisés par les médecins (présenté par le feature "réunion"),les statistiques d'un médecin (présenté par le feature "statistique")dans le cas ou les utilisateurs de l'application sont intéressés par voir les statistique d'un médecin,la gestion des patients et la gestion des assistants ce qui permet de modifier(ajouter,supprimer) ces deux derniers, ainsi les features "rechercher" et "contacter".

Le feature "réunion" implique deux autres features : « création de réunion », et le « rapport de réunion »(contenait les conclusions sur la discussion). Le feature " création de réunion " implique trois features : « gestion des participant»(membres de la réunion),et« thème de réunion »(le sujet de la réunion), et « date de réunion ». Dans le feature " création de réunion" les features « date de réunion »et « gestion des participant »sont obligatoires car c'est le feature où Vous précisez pour chaque réunion la Date, et les participants.

Le feature "consultation" est un feature obligatoire et implique deux autres features :soit « hors-ligne » soit « enligne ».

Le feature " catégorie "Précise si le médecin travaille dans sa propre cabinet ou dans une clinique.

Le feature "inscription" signifie l'ouverture d'un compte dans la plate forme qui est soit par faire un login s'il y a un compte déjà existe ou bien accéder avec Google ou bien ouvrir un nouveau compte.

Le feature "RDV" nous montre que la gestion de rendez-vous est soit par le médecin soit par l'assistant.

Le feature optionnel "statistique" montre les statistique d'un médecin ou d'une clinique tell que le nombre de patient par mois,nombre de consultation en ligne...

Le deuxième diagramme : Diagramme de Features métier -2-, qui est la suite de premier diagramme.

Les features "gérer patient" et "gérer assistant" dans le but de gestion des patient et des assistant , qui comportent l'ajout, la validation, la modification, la suppression.

Le feature "dossier médical" pour contrôler et gérer les dossiers médicaux contenant l'ordonnance,les analyses,l'imagerie,et la fiche de consultation.

Le paiement présente par le feature "payement" qui est un feature obligatoire,et se fait soit par virement bancaire, soit par carte crédit.

Les features "recherche" et "contacter" sont optionnels, ils peuvent être dans l'application et ils ne peuvent pas l'être,Le contact se fait par chat ou par email.

Le troisième diagramme : Diagramme de Features technique, qui contient Les fonctionnalités techniques :

Le feature «authentification » : est un feature obligatoire et ce fait par soit le "mot de passe et le nom de l'utilisateur" de compte ou par un "code unique" (c'est un code envoyé par l'application à l'utilisateur)

Pour « l'IHM » on a le feature obligatoire "Langue" qui donne la possibilité de choisir n'importe quel langues et le feature optionnel. Le feature "thème" : Changement de l'interface des utilisateurs.

Le feature «gestion profile » pour gérer compte utilisateurs (ajouter, modéfier).

4.3 Conception du domaine :

4.3.1 Présentation d'UML

Définition d'UML :

Le langage UML (Unified Modeling Language, ou langage de modélisation unifié) a été pensé pour être un langage de modélisation visuelle commun, et riche sémantiquement et syntaxiquement. Il est destiné à l'architecture, la conception et la mise en œuvre de systèmes logiciels complexes par leur structure aussi bien que leur comportement. L'UML a des applications qui vont au-delà du développement logiciel, notamment pour les flux de processus dans l'industrie.

Il ressemble aux plans utilisés dans d'autres domaines et se compose de différents types de diagrammes. Dans l'ensemble, les diagrammes UML décrivent la limite, la structure et le comportement du système et des objets qui s'y trouvent.

L'UML n'est pas un langage de programmation, mais il existe des outils qui peuvent être utilisés pour générer du code en plusieurs langages à partir de diagrammes UML. L'UML a une relation directe avec l'analyse et la conception orientées objet. [37]

Diagrammes d'UML :

Les diagrammes d'UML représentent des processus et des données modélisés graphiquement, chacun représentant un concept spécifique d'un système logiciel. Les diagrammes UML représentent des processus et des données modélisés graphiquement, chacun représentant un concept spécifique d'un système logiciel.

Nous distinguons trois types de diagrammes.

- **Diagrammes Statiques** : Représentent l'aspect statique d'un logiciel.
 - Diagramme de classe.
 - Diagramme de composants.
- **Diagrammes dynamiques** : Représentent l'aspect dynamique du logiciel comme le comportement et les interactions.
 - Diagramme d'activité.
 - Diagramme de séquence.
 - Diagramme d'état de transition.
 - Diagramme de communication.
 - Diagramme de temps.
- **Diagrammes fonctionnels** : Représentent le fonctionnement d'un système, tel qu'il est vu par les utilisateurs externes.

- Diagramme de cas d'utilisation.
- Diagramme de collaboration.

UML est un très grand langage de modélisation uniforme. On se limite ici à la présentation des diagrammes utilisés pour la modélisation de notre système (diagrammes de cas d'utilisation, diagrammes de séquence, diagrammes de classe).

- **Diagramme de cas d'utilisation :**

En langage UML, les diagrammes de cas d'utilisation modélisent le comportement d'un système et permettent de capturer les exigences du système.

Les diagrammes de cas d'utilisation décrivent les fonctions générales et la portée d'un système. Ces diagrammes identifient également les interactions entre le système et ses acteurs. Les cas d'utilisation et les acteurs dans les diagrammes de cas d'utilisation décrivent ce que le système fait et comment les acteurs l'utilisent, mais ne montrent pas comment le système fonctionne en interne.

- **Diagramme de séquence :**

Les diagrammes de séquence permettent de modéliser un système de manière dynamique et ils s'attachent principalement à montrer la circulation et l'ordre chronologique des messages, autrement dit, ils décrivent la circulation de l'information. Ils sont directement basés sur les scénarios définis avec les cas d'utilisation [35]

- **Diagramme de classes :**

Le diagramme de classes représente la structure statique d'un système sous la forme de classes et de relations et ne contient pas d'informations temporelles. Une classe est une représentation abstraite d'un ensemble d'éléments similaires[35]

4.3.2 Phase d'analyse

La phase de l'analyse vise à définir et à justifier la solution optimale qui répondra aux exigences du projet et aux besoins identifiés ainsi que les acteurs du système et les tâches associées à chacun. [29]

- **1. Spécification des besoins :**

Cette étape représente tous les critères qui représentent les besoins réels et nous permet d'inclure des fonctions clés ainsi que des objectifs précis afin d'atteindre leur rendement interne et leurs interactions avec leur environnement.

- **2. Identification des besoins :**

Notre système doit répondre aux exigences suivantes :

Les besoins fonctionnels :

Ce sont les actions que le système doit l'exécuter, il ne devient opérationnel que s'il les satisfait, Cette application couvrir les besoin fonctionnels suivants :

- La brève recherche des médecins ;
- Gérer les patients ;
- Télé-charger et imprimer les ordonnances délivrées par chaque praticien ;
- Prise de rendez-vous ;
- Gérer les spécialité ;
- Gérer le payement ;
- Contacter les médecines ;
- Gérer les télé-consultation.

Les besoins non fonctionnels :

Ce sont des exigences qui ne concernent pas spécifiquement le comportement Du système mais plutôt identifient des contraintes externes et internes du système, dans notre projet les besoins non fonctionnelles sont :

- **Facilité d'utilisation :** Le logiciel doit être facile à utiliser et puissant.
- **Performance :** Ce besoin est lié aux temps de réponse et à la productivité des logiciels.
- **Portabilité :** Adaptabilité à d'autres environnements logiciels.
- **Sécurité :**Intégrité des données et protection de l'accès pour assurer la sécurité des données.
- Le code doit être clair pour la facilite d'améliorations (les mises à jours) ;

- **3. Acteurs et tâches du système :**

Un Acteur : les acteurs ou bien les participants représentent les rôles joués par des entités externes qui interagissent directement avec le système de recherche, C'est-à-dire que l'acteur peut directement vérifier et/ou modifier l'état du système par envoi. Et/ou la réception de messages pouvant contenir des données.

Une tâche : est une fonction parmi plusieurs fonctions attribuées aux participants qui peuvent accéder et réaliser.

Dans le cadre de notre étude nous avons distingué quatre acteurs principaux, le tableau ci-dessous nous citons les tâches associées à chaque acteur :

Acteur	Taches
Admin	<ul style="list-style-type: none"> - Authentification - Gérer les spécialité(ajouter, supprimer) - Gérer compte médecin(valider ,supprimer) - Gérer compte patient(valider ,supprimer) - Gérer base de données
Médecin	<ul style="list-style-type: none"> - Authentification - Gérer RDV - Gérer spécialité - Consulter Statistique - Gérer télé-consultation - Recherche patient - Consulter boîte de réception - Gérer patient (ajouter ,supprimer) - Consulter liste de patients - Gérer dossier médical - Consulter dossier patient - Créer ou participer à une réunion
Patient	<ul style="list-style-type: none"> - Authentification - Chercher médecin - Prendre RDV - Consulter boîte de réception - Remplir fiche patient - Contacter médecin - Gérer dossier médical - demande d'ordonnance - Passer une consultation enligne - Modifier profile
Assistant médical	<ul style="list-style-type: none"> - Authentification, - Consulter boîte de réception - Contacter médecin - Contacter patient - Gérer consultation enligne - Gérer les RDVs - Consulter le planning des RDVs

TABLE 4.1 – Acteurs et leurs tâches.

4.3.3 Phase de conception

La phase de conception complète la phase d'analyse en fournissant plus de détails pour chaque exigence spécifique et chaque processus par lequel l'application interagit avec les objets.

4.3.3.1 Diagramme de cas d'utilisation :

Le cas d'utilisation suivant illustre notre comportement fonctionnel de future Application. Chaque cas d'utilisation représente une tâche qu'on associe à un acteur du système le déroulement de ces tâches est représenté sous forme des scénarios.

4.3.3.2 Diagramme de cas " Admin " :

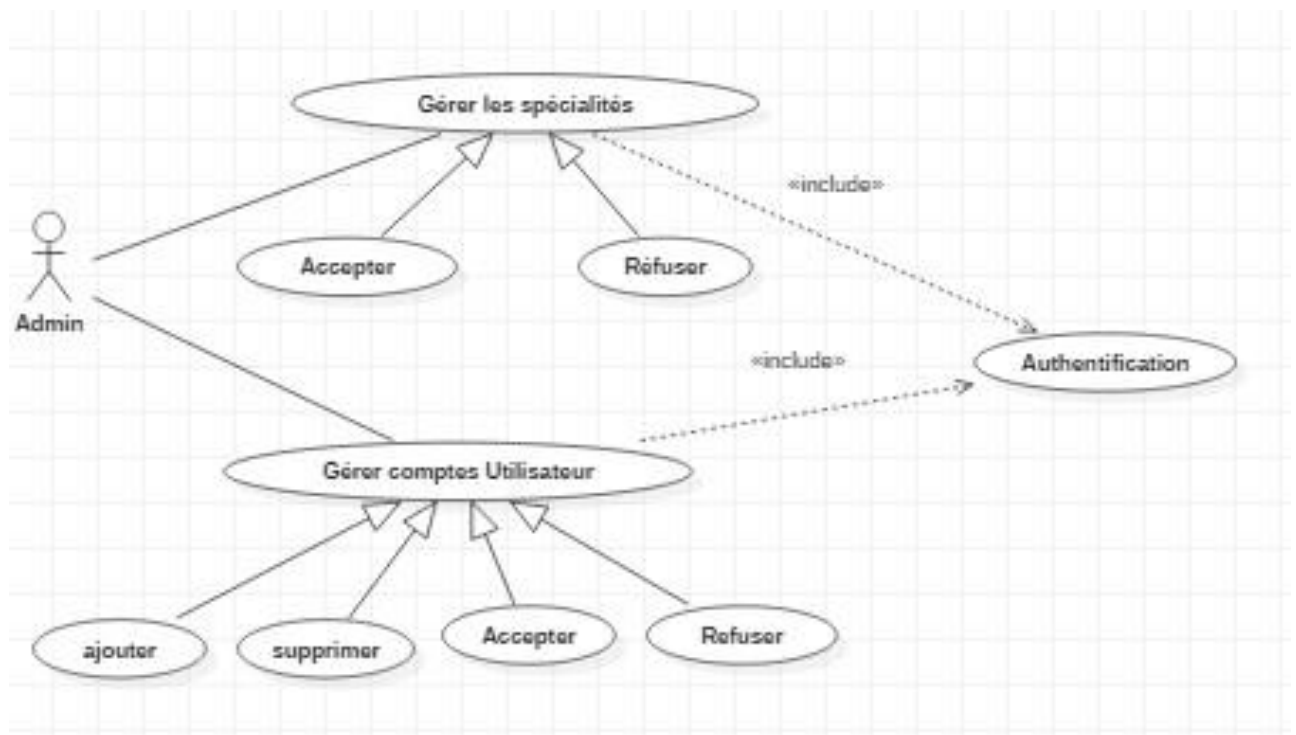


FIGURE 4.4 – diagramme de cas d'utilisation "Admin"

4.3.3.3 Diagramme de cas "Médecin " :

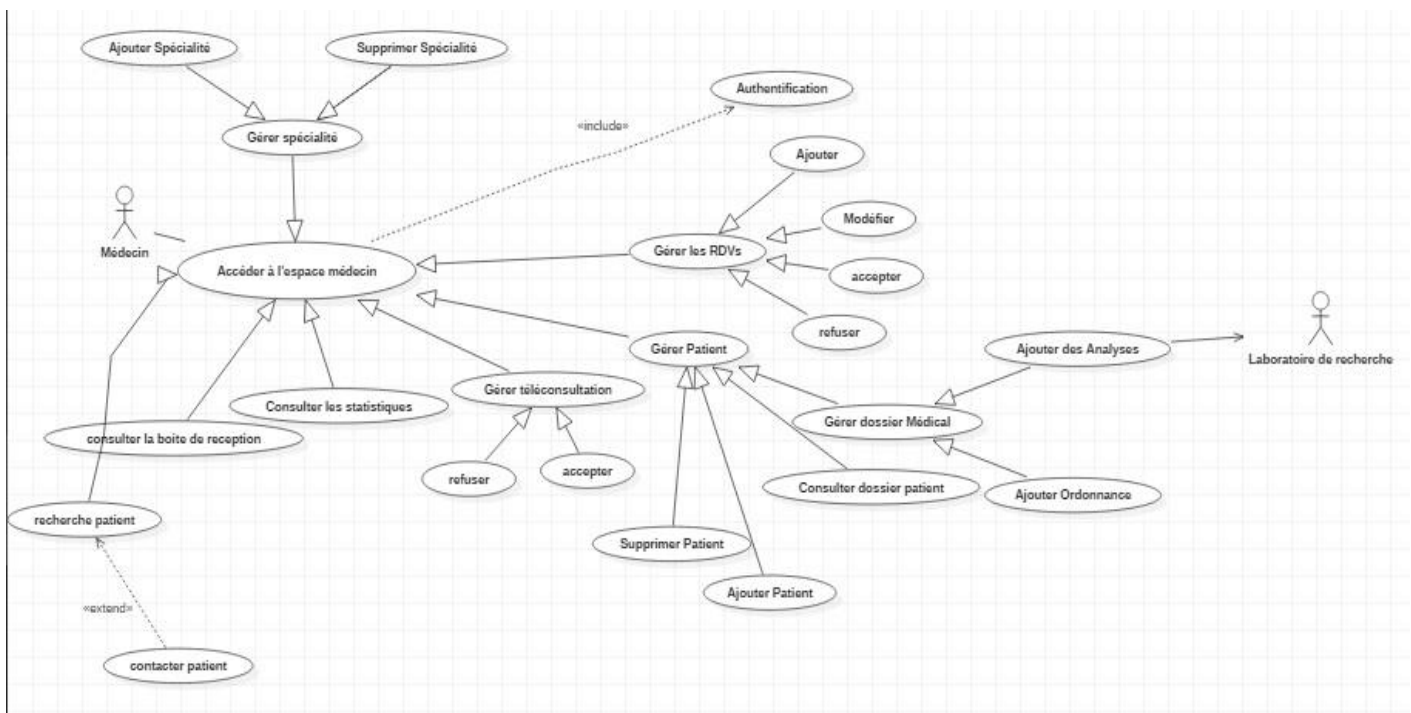


FIGURE 4.5 – diagramme de cas d'utilisation " médecin"

4.3.3.4 Diagramme de cas "Assistant médical" :

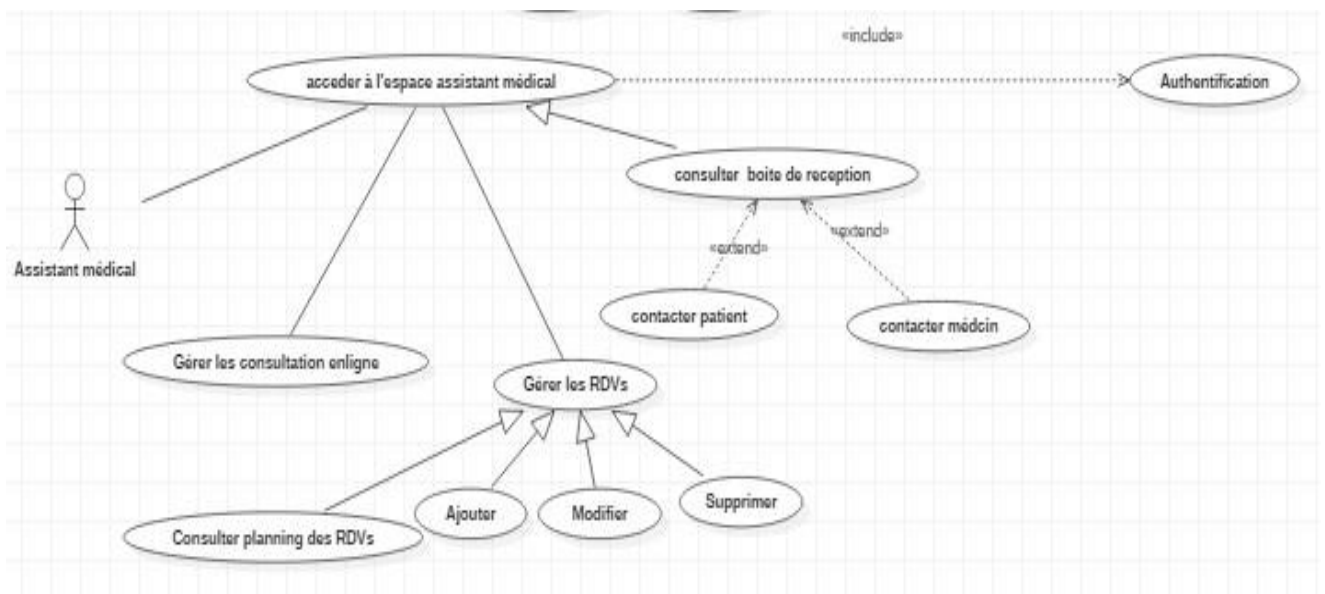


FIGURE 4.6 – diagramme de cas d'utilisation " Assistant médical"

4.3.3.5 Diagramme de cas "Patient" :

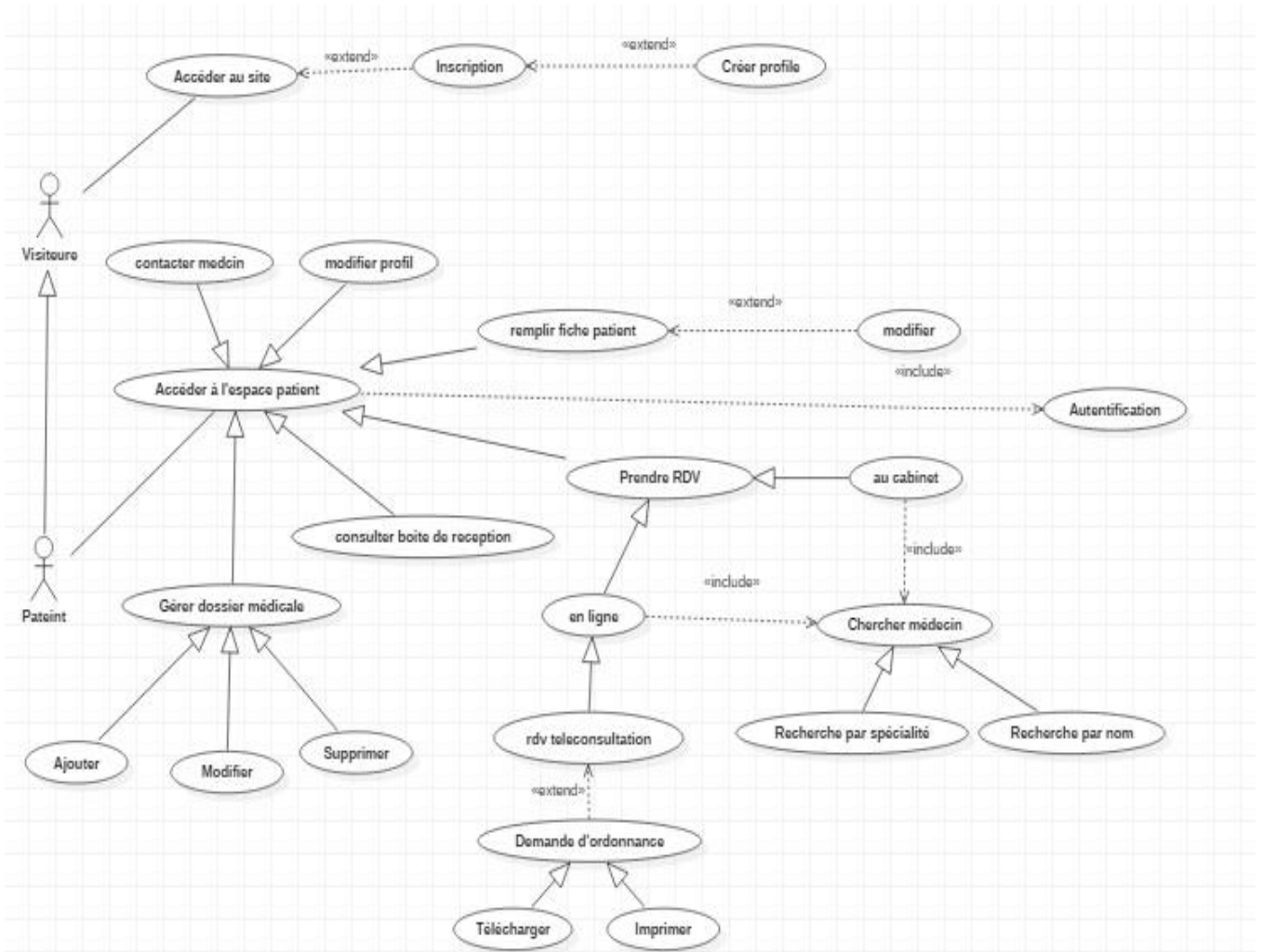


FIGURE 4.7 – diagramme de cas d'utilisation "patient"

4.3.3.6 Diagramme de classe

Le diagramme des classes est un diagramme structurel (statique) qui permet de représenter :

- Les classes (attributs + méthodes)
- Les associations (relations) entre les classes.

Le diagramme de classes est le plus important des diagrammes UML, c'est le seul qui soit obligatoire lors de la modélisation objet d'un système.

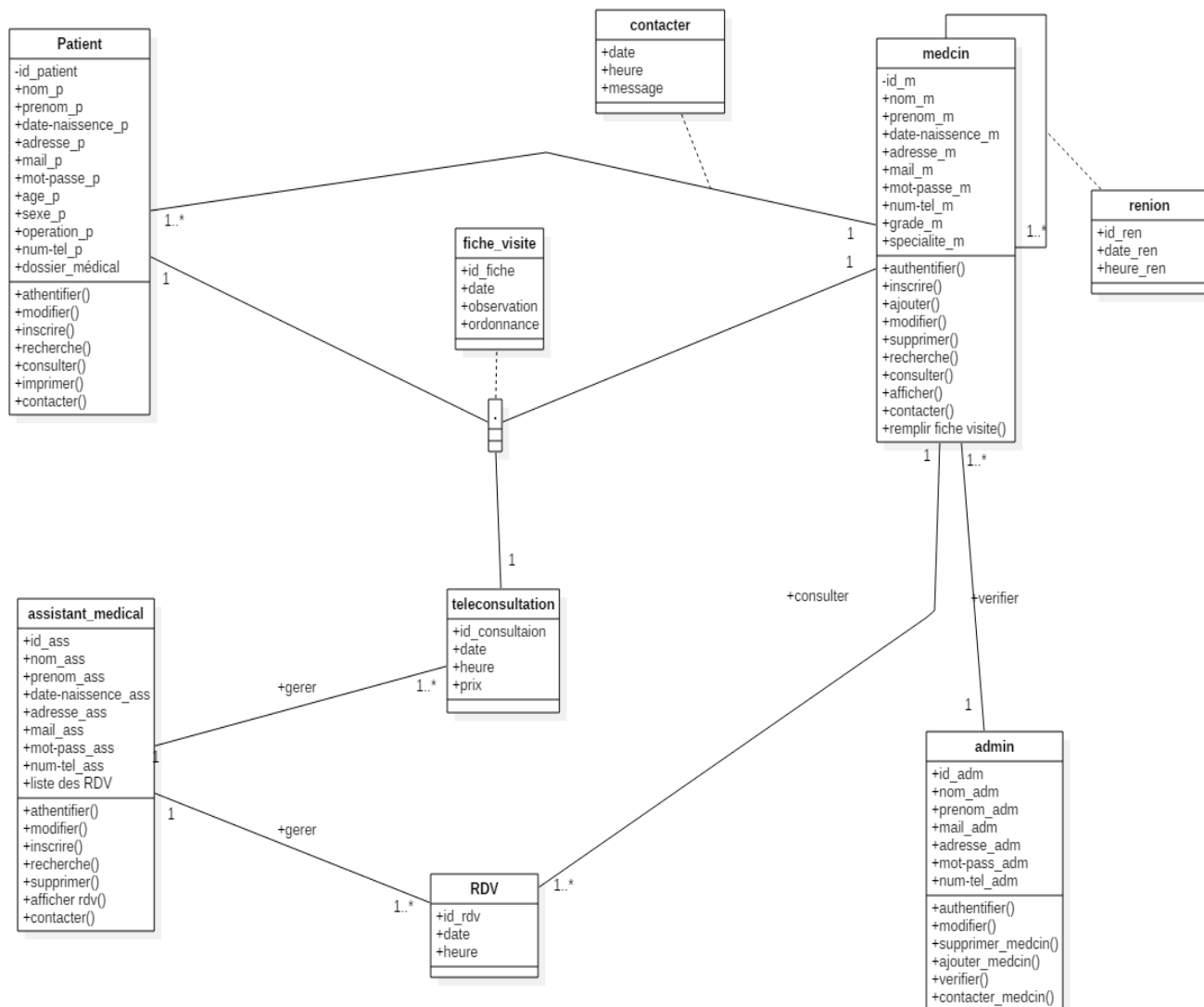


FIGURE 4.8 – Diagramme de classe de l'application e-health

4.3.4 Présentation SoaML(Service-oriented architecture Modeling Language)

4.3.4.1 Définition SoaML :

Service-Oriented Architect (SOA) nous permet de décrire et de comprendre comment les personnes, les organisations et les composants du système fonctionnent ensemble en utilisant des services pour atteindre des objectifs.

Le langage SoaML (Service oriented architecture Modeling Language) a été introduit par l'OMG (Object Management Group), afin de concevoir des architectures SOA. SoaML est une extension du langage UML pour supporter les exigences de modélisation de services SOA. Le langage SoaML fournit une norme d'architecture pour les modèles de solutions SOA. [38]

4.3.4.2 Diagrammes SoaML :

- **Diagramme d'interface de service :**

Le diagramme d'interface de service est l'un des types de diagramme SoaML les plus importants. Afin de comprendre ce qu'est le diagramme d'interface de service, vous devez d'abord connaître un concept clé de SoaML - le service

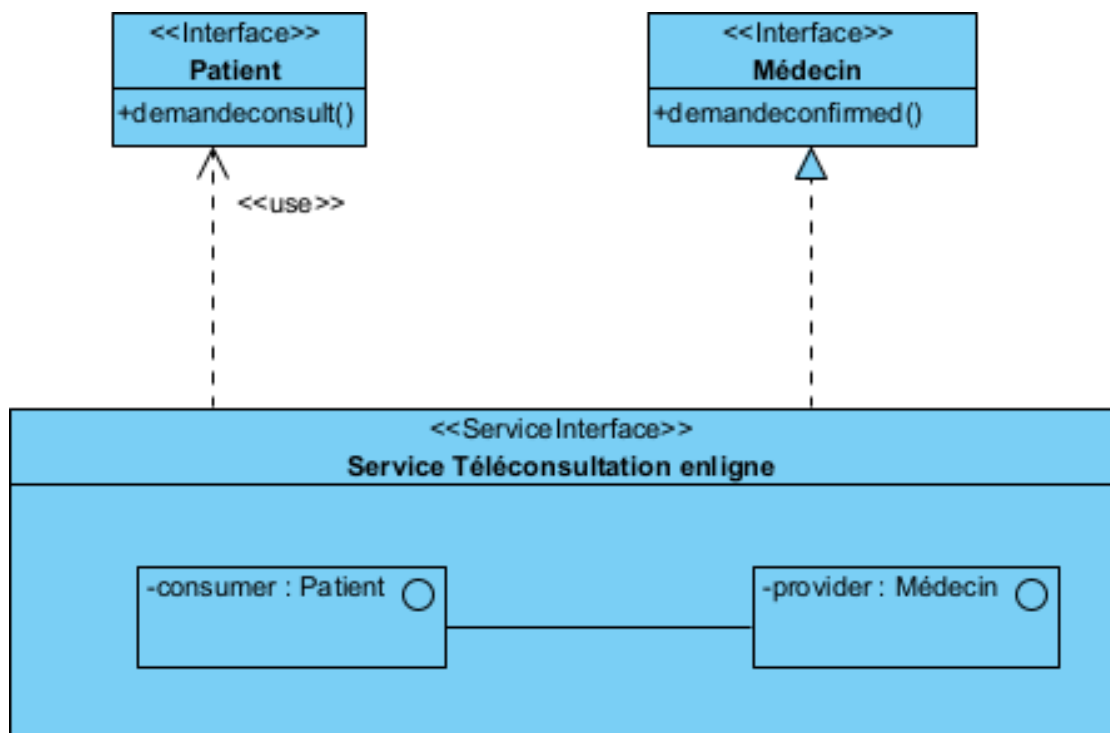


FIGURE 4.9 – Diagramme d'interface de service

- **Diagramme de contrat de service SoaML :**

Le contrat de service définit l'accord entre les parties sur la manière dont un service doit être fourni et consommé.

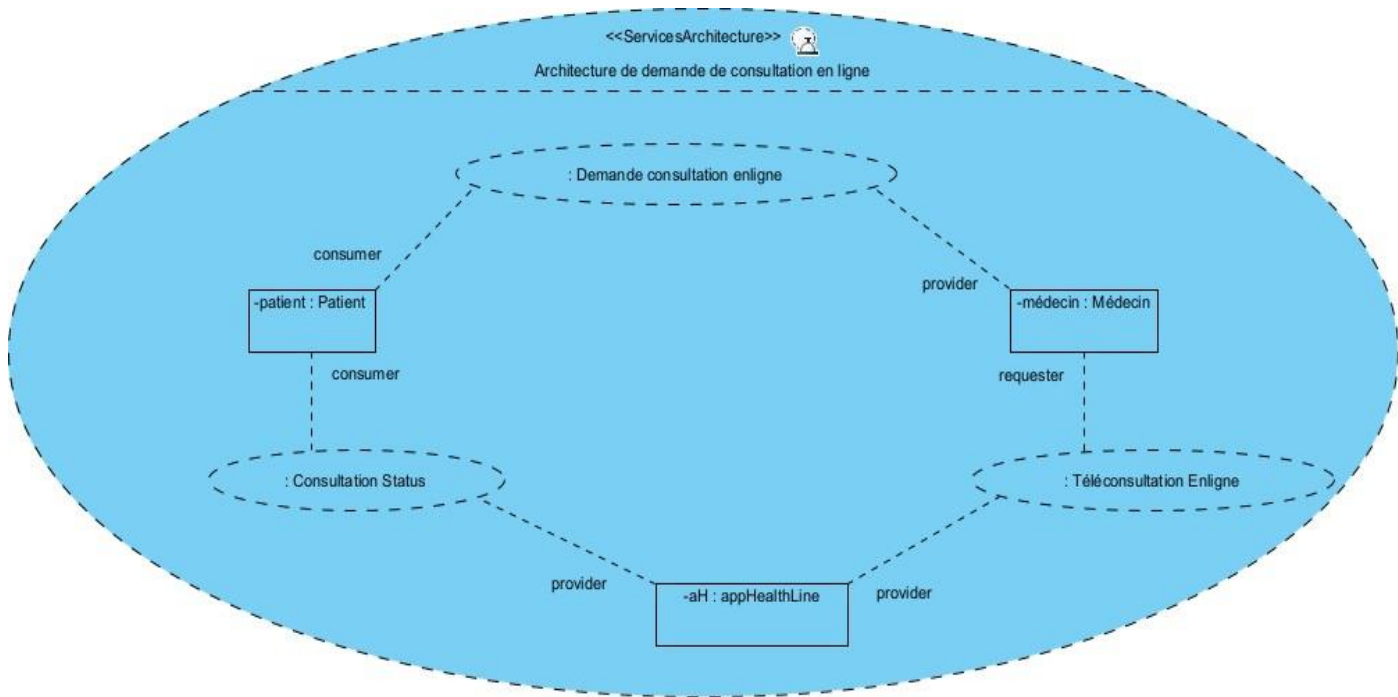


FIGURE 4.10 – Diagramme du contrat de service

4.3.5 Modélisation de l'architecture d'application

Le diagramme suivant présente l'architecture orienté service de l'application où chaque fonctionnalité est reliée (dépend)un ou plusieurs services :

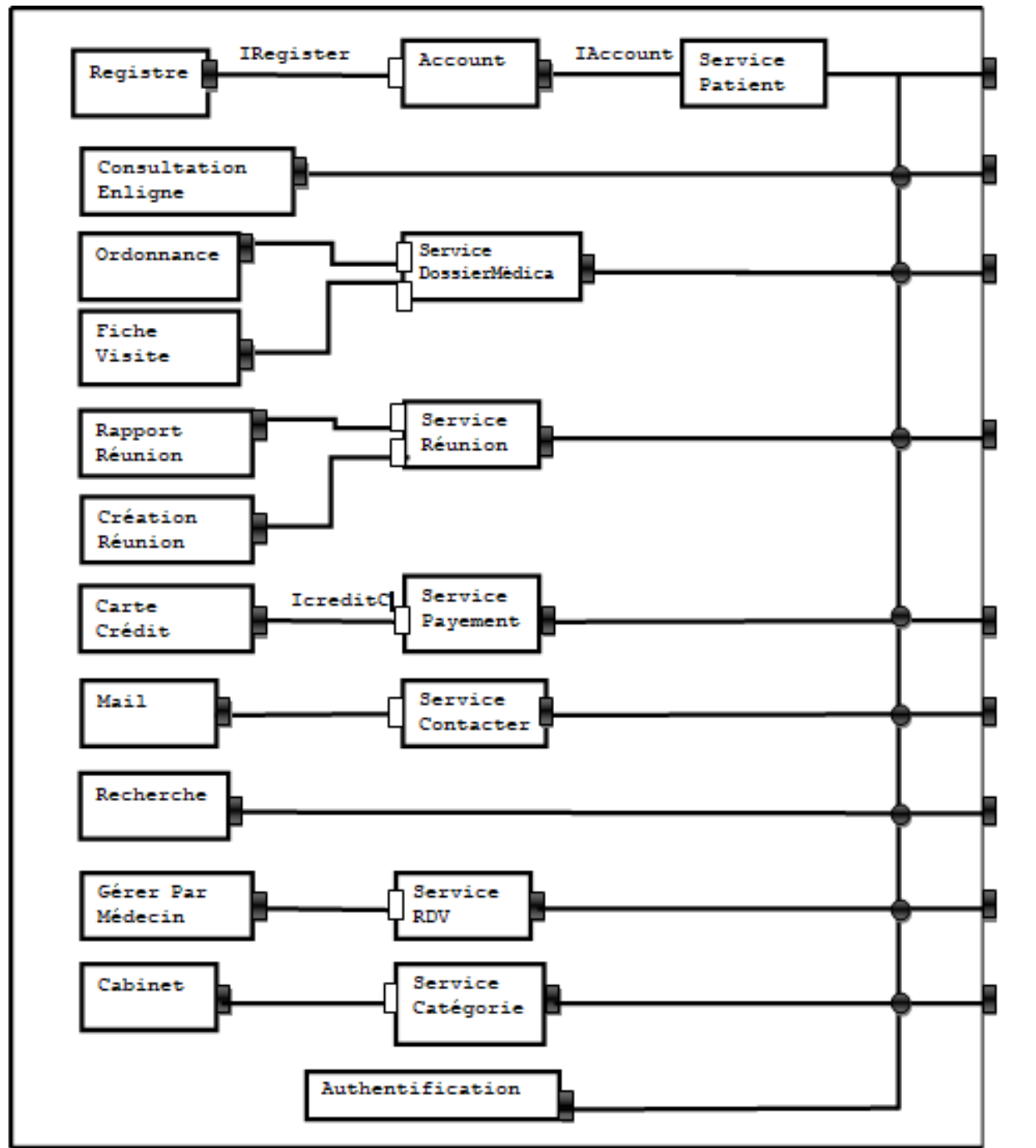


FIGURE 4.11 – Diagramme d'architecture des services web

4.3.6 Architecture de notre application

le feature model qui identifie les besoins des utilisateurs en sélectionnant des features au niveau de la configuration qui gère les services Web (activation/désactivation), puis les services Web stockent des données dans une base de données après qu'un client envoie des requêtes HTTP dans des méthodes (post , get ,delete ,put) et réception des données au format JSON ou XML.

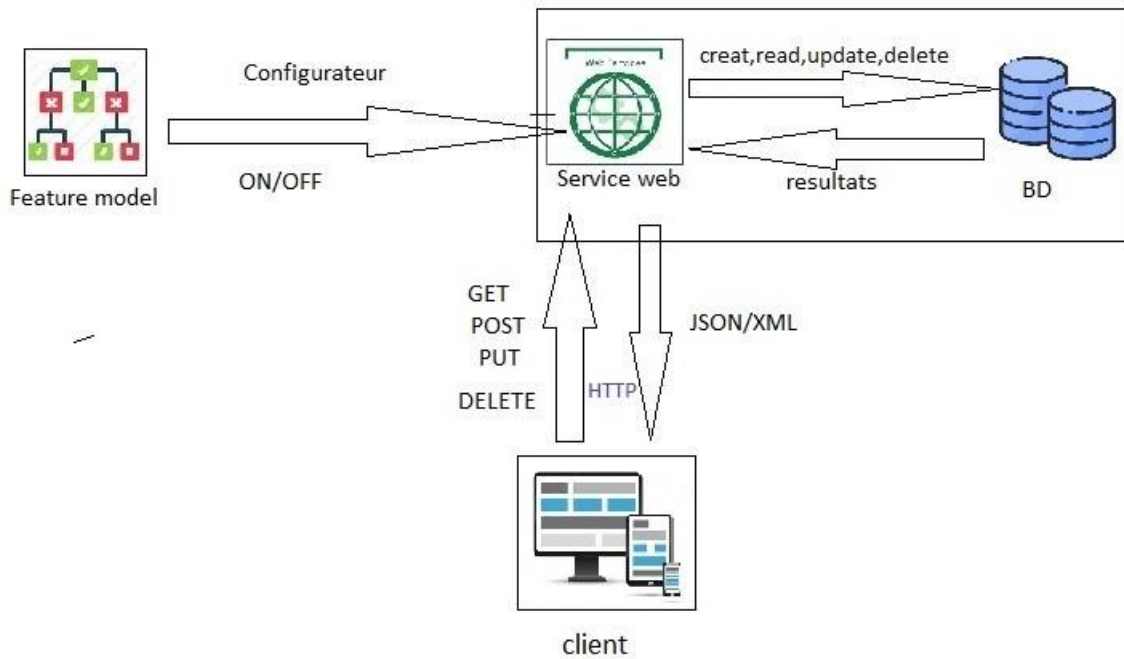


FIGURE 4.12 – Architecture de notre application

4.4 Conclusion

Dans ce chapitre, nous avons introduit les phases d'analyse (analyse de domaine) et de conception (conception de domaine) des exigences et planifié le fonctionnement de notre future plate-forme, dans lequel nous avons décrit notre feature model et sa description en détail , et ainsi la phase de conception et modélisation basée sur les diagrammes du langage UML, à savoir les diagrammes de cas d'utilisation, les diagrammes de séquence, les diagrammes de classes, pour faciliter la mise en place concrète de la plateforme, qui fera l'objet du chapitre suivant.

Ingénierie de l'application

5.1 Introduction

L'ingénierie de l'application consiste à utiliser les résultats de l'ingénierie de domaine pour la construction également appelée dérivation d'un produit particulier. C'est un développement par réutilisation. Les résultats d'ingénierie sur domaine (feature model, les diagrammes UML) contiennent une variabilité, de sorte que la dérivation d'un produit particulier nécessite des décisions (ou des choix) associées à ces points de variation. pour cela nous utilisant la sélection a partir de feature model pour fixer les décisions selon un cas particulier d'application de-health.

Nous allons dérivés une application pour montrer l'intérêt de l'utilisation de cette approche. Dans notre projet on se contente à un exemple d'application, détaillées dans la suite de ce chapitre.

5.2 Environnement logistique

5.2.1 Matériels utilisés

Notre application sera réalisée sur des machines dont les caractéristiques sont les suivantes :

- Deux pcs portables de configuration suivants :

DELL i3 4ème génération 64 bits, 6GB de Ram, disque dur HDD 500GB.

DELL i5 6ème génération 64bits, 8GB de Ram, disque dur SSD 256GB.

Deux Smart phone : Oppo A5s, Redmi note 9s.

5.2.2 Environnement de développement

5.2.2.1 Technologies de développement

Nous avons travaillé par plusieurs langages à savoir : Java pour le développement des services web (restfull service) et Java Script, HTML, CSS pour (restfull client) ces derniers appellent les "API" (Application Programming Interfaces) des services web et créer l'application web E-health.

Nous avons utilisé java swing pour le configurateur qui représente le Feature Model et qui gère (activer/désactiver) les services web.

Nous avons choisi le Java puisque est un langage de haut niveau et portable sur d'autres plate-formes.

5.2.2.2 Outils de programmation

NetBeans JEE est un IDE (Environnement de Développement Intégré) puissant pour Java. Il permet au développeur de gérer ses projets avec une interface riche, il est plus rapide, plus efficace et plus fiable que d'autre Java IDE.

Pour faire fonctionner une application web Java EE, nous avons besoin de un serveur d'applications. Nous avons choisi d'utiliser GlassFish server, car c'est un serveur léger, libre, multiplateforme et complet et nous avons utilisé une base de données MySQLWorkbench pour stocké les données.

5.2.3 Dérivation d'une application

L'application dérivée à partir de notre ligne de produit est une application web.

la dérivation d'un produit nécessite plusieurs étapes :

- 1- étude Objectifs et préférences des utilisateurs .
- 2- Analyse des besoins des utilisateurs.
- 3- sélection des features au niveau de la configuration.
- 4- gère les services Web (activation/désactivation) a partir de la configuration .
- 5- traitement et stockage des données dans une base de données avec des codes source et des méthodes et des opérations .
- 6- Plateforme dérivée.

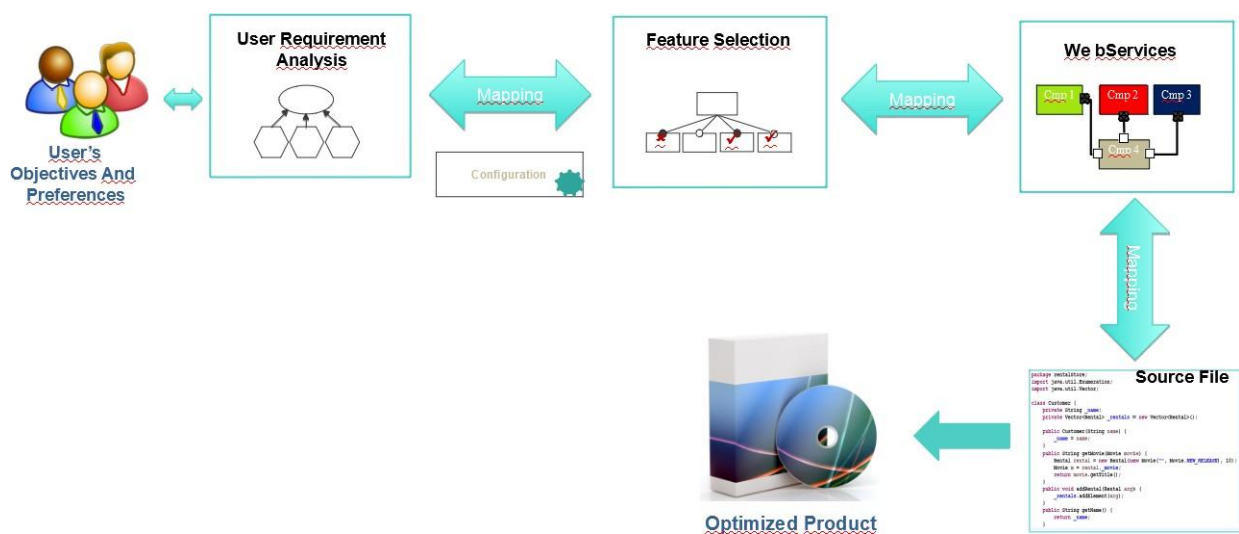


FIGURE 5.1 – Dérivation d'une application

5.2.3.1 Analyse d'application

Comme on a dit précédemment, il faut sélectionner les exigences de l'application depuis les exigences du domaine.

L'application dérivée à partir de notre ligne de produit s'occupe d'un cabinet médicale d'un diabétologie en ligne (CABINET DIABETOLOGIE HEALTHLINE) avec des interfaces graphiques en français et authentification des utilisateurs à l'aide de username et password, le system d'exploitation Windows et utilise une base de données MySQL.

5.2.3.2 Modélisation d'application

Le feature modèle de cette application après la sélection des fonctionnalités est le suivant :

1. Feature Model d'application (1) :

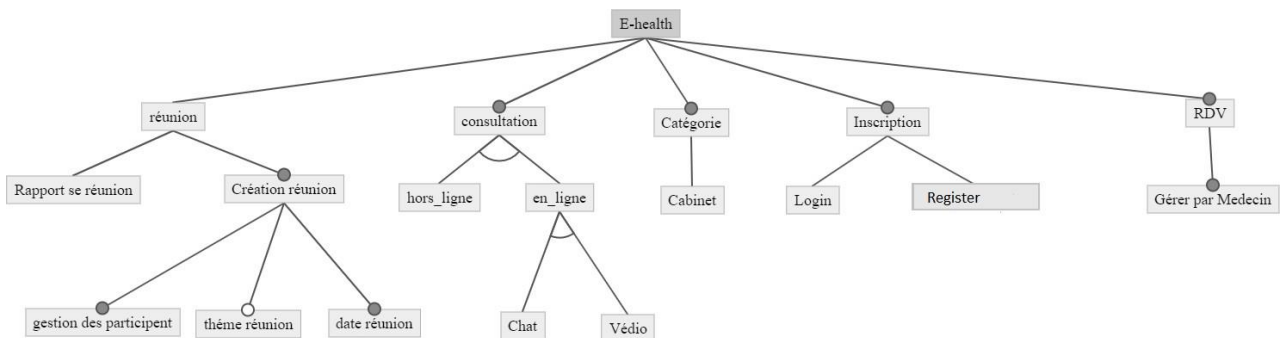


FIGURE 5.2 – Feature Model d'application (1)

2. Feature Model d'application (2) :

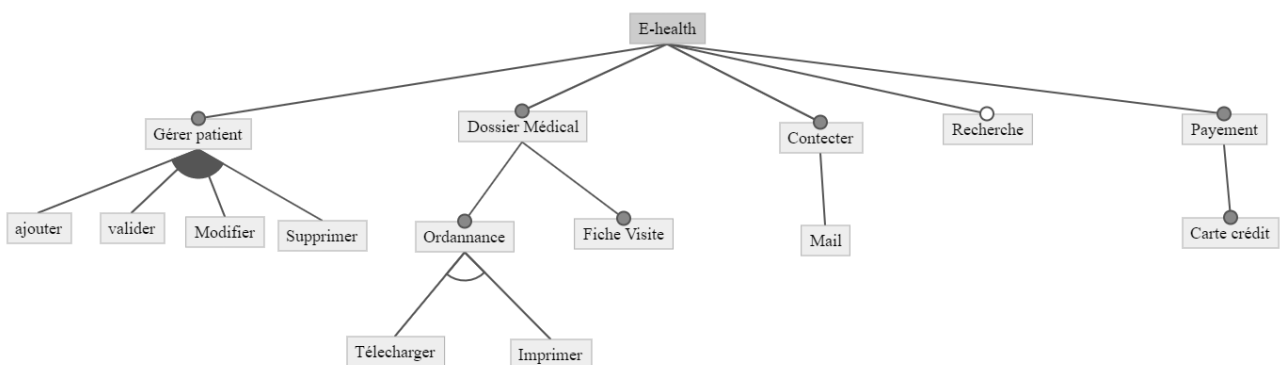


FIGURE 5.3 – Feature Model d'application (2)

3. Feature Model d'application technique :

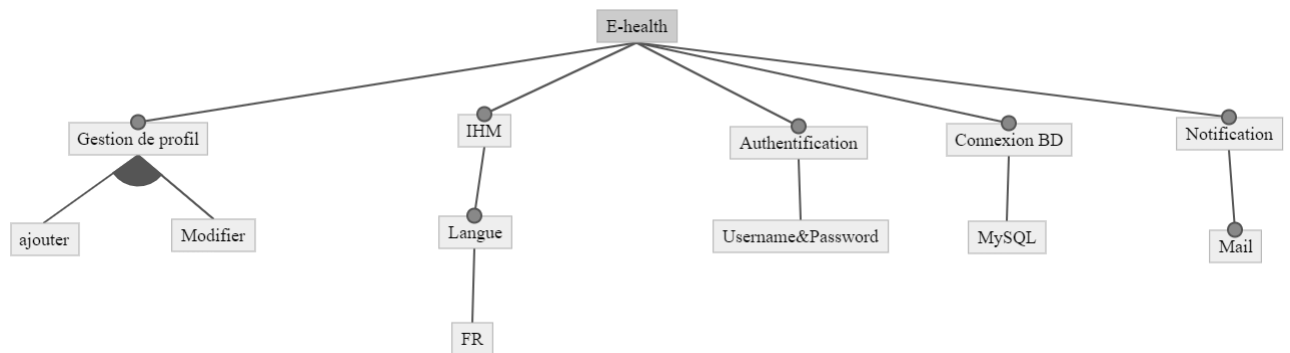



FIGURE 5.4 – Feature Model d'application technique)

5.2.3.3 Les interfaces

Les figures suivantes présentent quelques interfaces de Notre ligne de produits :

1 -Le configurateur de la ligne de produit :

sélection des features (fonctions) au niveau de la configurateur ,selon le choix des besoins de l'utilisateur .qui liee avec les servi ces Web (ON / OFF).

 Configurateur

E_HEALTH

<input checked="" type="radio"/> ON	Inscription	<input checked="" type="checkbox"/> Login
<input type="radio"/> OFF		<input type="checkbox"/> Par Google
		<input checked="" type="checkbox"/> Registration
<input checked="" type="radio"/> ON	Authentification	<input type="checkbox"/> Code unique
<input type="radio"/> OFF		<input checked="" type="checkbox"/> Username & Password
<input checked="" type="radio"/> ON	Catégorie	<input checked="" type="radio"/> Cabinet
<input type="radio"/> OFF		<input type="radio"/> Clinique
<input checked="" type="radio"/> ON	Consultation	<input checked="" type="checkbox"/> En_Ligne
<input type="radio"/> OFF		<input checked="" type="checkbox"/> Chat
		<input checked="" type="checkbox"/> Vidéo
		<input type="checkbox"/> Audio
		<input checked="" type="checkbox"/> Hors_Ligne

FIGURE 5.5 – Interface Configurateur "1"

Vidéo
 Audio

Hors_Ligne

ON OFF **RDV**

Gérer Par Médecin
 Gérer Par Assistant
 Calendrier

ON OFF **Contacter**

Chat
 Mail

ON OFF **Dossier Médical**

Ordonnance
 Télécharger
 Imprimer

Analyses
 imagerie
 Fiche consultation

FIGURE 5.6 – Interface Configurateur "2"

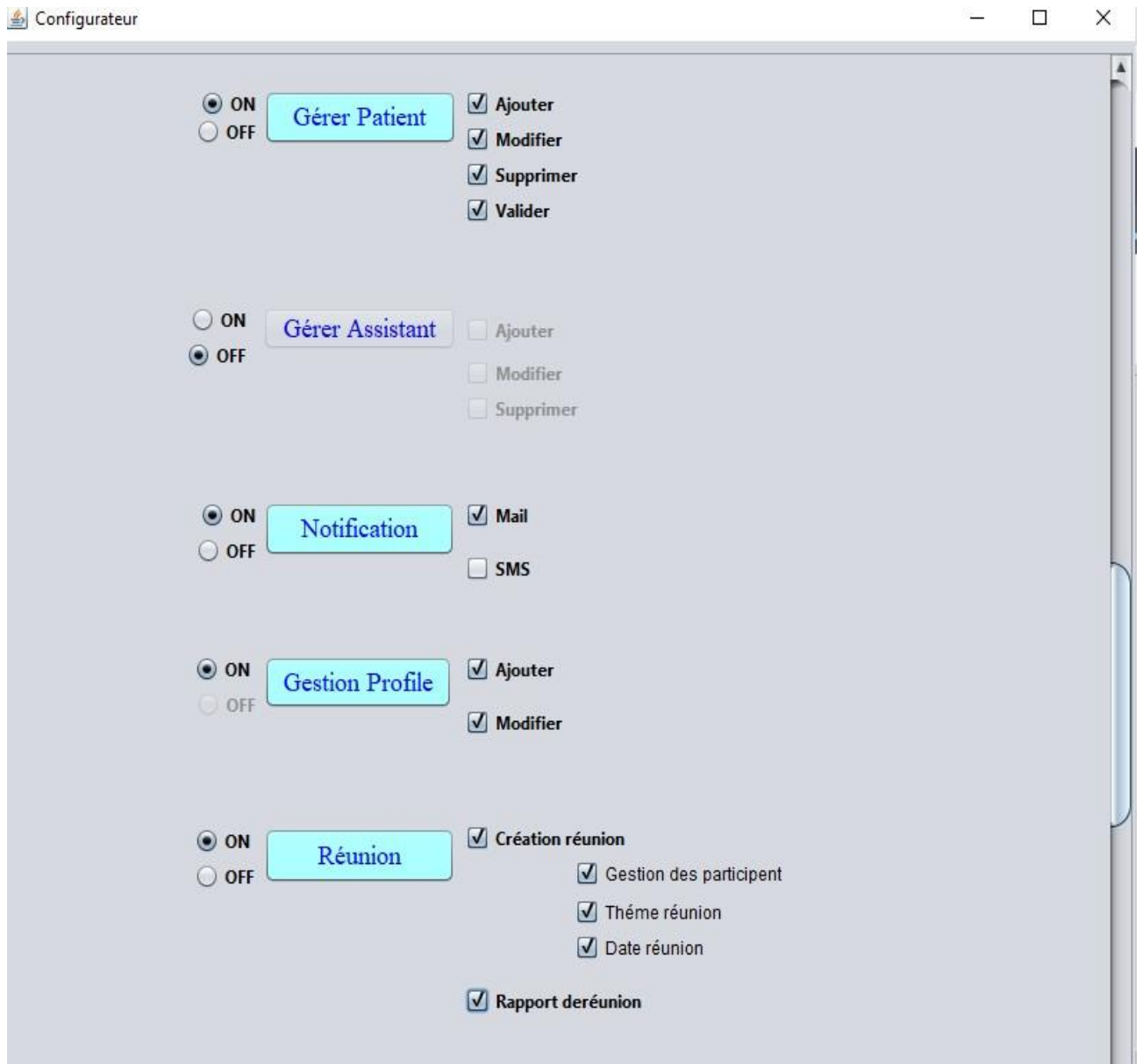


FIGURE 5.7 – Interface Configurateur "3"



FIGURE 5.8 – Interface Configurateur "4"

2 -Les services web (Rest FullService) de E-Health :

Code source de méthode "GET" des médecin :

Cette figure indique le code source de la méthode "GET" qui renvoie les informations du médecin.

```

@GET
@Path("/{id}")
@Produces({MediaType.APPLICATION_XML, MediaType.APPLICATION_JSON})
public Medecin find(@PathParam("id") Integer id) {
    return super.find(id);
}

@GET
@Override
@Produces({MediaType.APPLICATION_XML, MediaType.APPLICATION_JSON})
public List<Medecin> findAll() {
    return super.findAll();
}

@GET
@Path("/{from}/{to}")
@Produces({MediaType.APPLICATION_XML, MediaType.APPLICATION_JSON})
public List<Medecin> findRange(@PathParam("from") Integer from, @PathParam("to") Integer to) {
    return super.findRange(new int[]{from, to});
}

@GET
@Path("/count")
@Produces(MediaType.TEXT_PLAIN)
public String countREST() {

```

FIGURE 5.9 – Code source de méthode "GET" des médecin

Code source de méthode "POST" des médecin :

Cette méthode montre comment le médecin remplit ces informations .cette méthode utilise lorsque le médecin fait un Sign-in ou Sign-up.

```
48     @POST
49     @Path("insertion")
50
51     @Produces(MediaType.APPLICATION_XML)
52     @Consumes(MediaType.APPLICATION_FORM_URLENCODED)
53     public String insertmed(@FormParam("nommed") String nommed, @FormParam("prenommed") String prenommed ,
54         @FormParam("mailmed") String mailmed, @FormParam("numtelmed") String numtelmed,
55         @FormParam("grademed") String grademed, @FormParam("passwordmed") String passwordmed,
56         @FormParam("datenaissancemed") Date datenaissancemed, @FormParam("adressemmed") String adressemmed,
57         @FormParam("specialitemed") String specialitemed) {
58
59         System.err.println("hada nom : "+nommed+ "hada prenom : "+prenommed);
60         Medecin entity = new Medecin();
61         entity.setNommed(nommed);
62         entity.setPrenommed(prenommed);
63         entity.setMailmed(mailmed);
64         entity.setNumtelmed(numtelmed);
65         entity.setGrademed(grademed);
66         entity.setPasswordmed(passwordmed);
67         entity.setDatenaissancemed((java.sql.Date) datenaissancemed);
68         entity.setAdressemmed(adressemmed);
69         entity.setSpecialitemed(specialitemed);
70
71         super.create(entity);
72     }
73     /*
74     try {
```

FIGURE 5.10 – Code source de méthode "POST" des médecin

Ce figure présente l'exécution de la méthode "POST" :

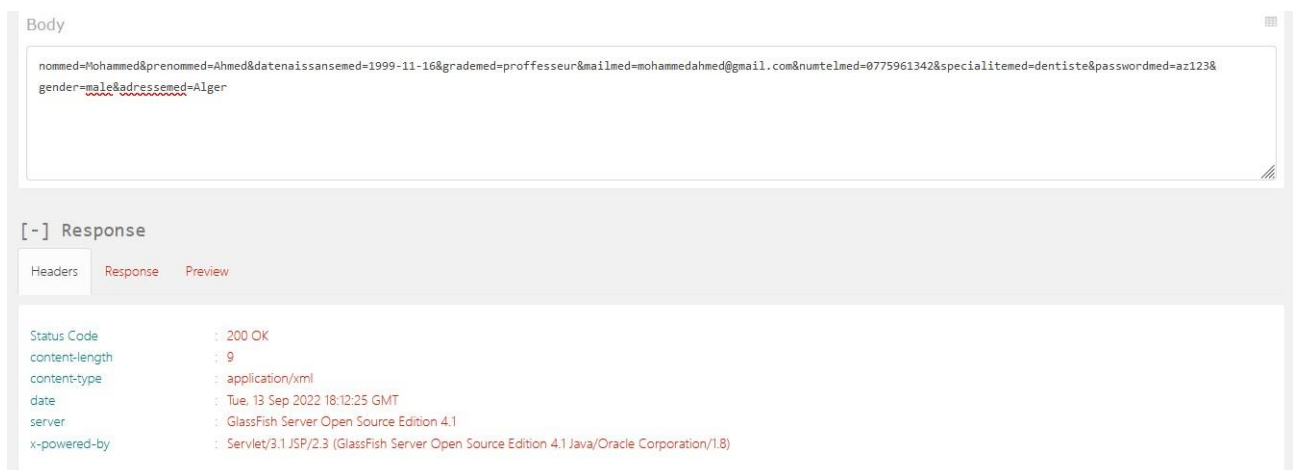


FIGURE 5.11 – Registrassions d'un médecin "OK"

Code source de méthode "PUT" modifier les information des médecin :

Cette méthode est utilisée lorsque le médecin souhaite modifier certaines de ses informations personnelles (exemple :l'email, password,...).

```
@PUT
@Path("edite/{mad}")
@Consumes("application/json")
public Medecin UpdateMed(@PathParam("mad") Integer idmed, Medecin mUpdated,
Medecin m =em.find(Medecin.class, idmed);

    if (m==null) {
        return null;
    }
    if(mUpdated.getNommed() !=null) {
        m.setNommed(mUpdated.getNommed());
    }

    if(mUpdated.getPrenommed() !=null) {
        m.setPrenommed(mUpdated.getPrenommed());
    }

    if(mUpdated.getMailmed() !=null) {
        m.setMailmed(mUpdated.getMailmed());
    }

    if(mUpdated.getPasswordmed() !=null) {
        m.setPasswordmed(mUpdated.getPasswordmed());
    }
}
```

FIGURE 5.12 – Code source de méthode "PUT"Code source de "Authentification" des médecin :

Ce code fonctionne lorsque l'utilisateur se connecte en appelant la méthode (isAuthenticated (auth)) qui vérifie l'email et le mot de passe et retourne succès ou erreur.

```
@GET
@Path("checkloginm")
//@Produces("MediaType.APPLICATION_FORM_URLENCODED")
@Produces("text/plain")
//@Consumes(MediaType.APPLICATION_FORM_URLENCODED)
public String checkloginm(@HeaderParam("authorization") String auth) {
    System.out.println(auth);
    if(isAuthenticated(auth)) {
        return"success";
    }
    else{
        return"error";
    }
}
```

FIGURE 5.13 – Code source de "Authentification" des médecin

Test des service web de E-health :

Cette figure montre l'interface de serveur Glassfish qui contient l'ensemble des services.

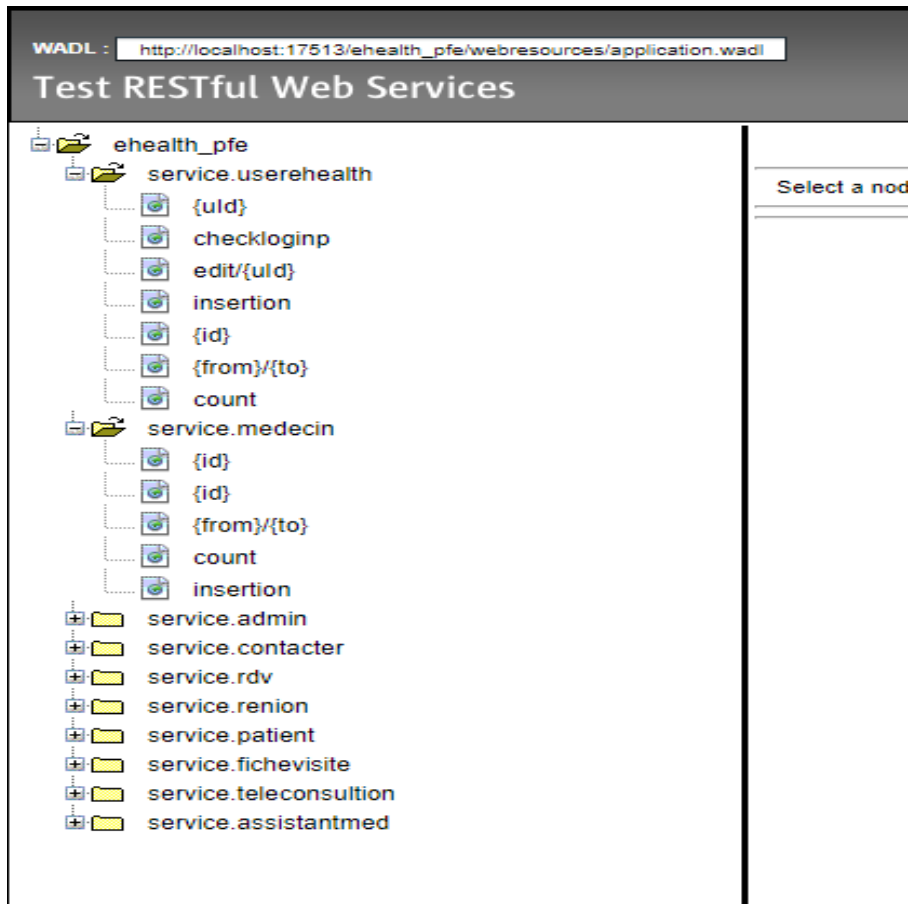


FIGURE 5.14 – Test Rest Ful Web Services

Afficher tous les médecins de service médecin avec la méthode "GET" dans un fichier XML ou JSON :

Cette figure montre le fonctionnement de la méthode "GET" au niveau de serveur et la réponse retournée sous format XML.

The screenshot displays a REST client interface with a tree view on the left and a main workspace on the right. The tree view shows a hierarchy of services, with 'service.medecin' selected. The main workspace shows the details of a GET request to the resource 'service.medecin/{id}' at the URL 'http://localhost:17513/ehealth_pfe/webresources/service.medecin/{id}'. The method is set to 'GET(application/xml)' and the status is '200 (OK)'. The response is shown in 'Raw View' as an XML document containing two doctor records.

```
<?xml version="1.0" encoding="UTF-8"?>
<medecins>
  <medecin>
    <grademed>doggtetf</grademed>
    <idmed>1</idmed>
    <mailmed>tenefyerg@</mailmed>
    <nommed>sohaib</nommed>
    <numtelmed>065675555</numtelmed>
    <passwordmed>99999</passwordmed>
    <prenommed>titouali</prenommed>
  </medecin>
  <medecin>
    <grademed>hbhgggyyg</grademed>
    <idmed>2</idmed>
    <mailmed>fdrrrrdr</mailmed>
    <nommed>uhuhhy</nommed>
    <numtelmed>05457788</numtelmed>
    <passwordmed>100000</passwordmed>
    <renommed>drtrtrtrtr</renommed>
  </medecin>
</medecins>
```

FIGURE 5.15 – Afficher les medecin de service Medecin avec la méthode "GET"

3 - Les interfaces de l'application «CABINET DIABETOLOGIE HEALTHLINE» :

Interface d'accueil de patient :

La page « d'accueil » est la première interface visualisée par le patients lors du lancement de la plateforme et s'il est intéressé, il doit s'inscrire par le patient.



FIGURE 5.16 – Interface d'accueil de patient

Interface de "créer un compte patient" :

La page « Inscription » est affichée selon le patient, lors de lancement de la plateforme. Il doit remplir un formulaire et le valider. C'est le système qui gère ensuite l'orientation de patient vers l'espace correspondant à ce dernier.

The image shows a registration form titled 'CRÉER UN COMPTE'. The form is set against a background of abstract green and blue geometric shapes. The form fields are: 'Nom' (text input), 'Prénom' (text input), 'Date de naissance' (date picker with 'jj/mm/aaaa' format), 'Gender' (radio buttons for 'Male' and 'Female'), 'Age' (text input), and 'Groupage' (dropdown menu with the text 'choisir votre groupage').

FIGURE 5.17 – Interface de "créer un compte patient"(1)

FIGURE 5.18 – Interface de "créer un compte patient"(2)

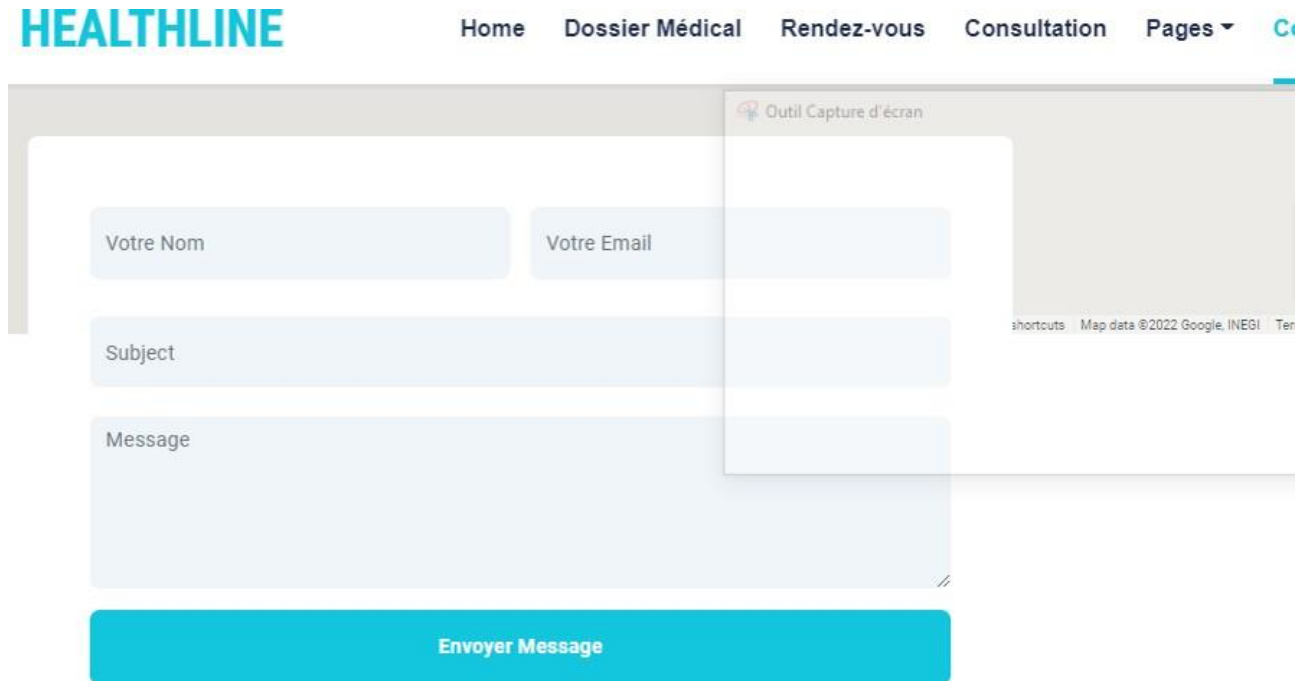
Interface de "prendre RDV" :

La page « prendre RDV » est affichée selon le patient, quand il veut prendre un rendez-vous pour la consultation. en attente de l’approbation du médecin.

FIGURE 5.19 – Interface de "prendre RDV"

Interface de "Contacter le médecin" :

La page "Contacter le médecin" s'affiche selon le patient, quand il veut contacter le médecin ou poser une question.. ect. en attente la réponse du médecin.



The screenshot shows the 'HEALTHLINE' website interface. At the top, there is a navigation menu with the following items: 'Home', 'Dossier Médical', 'Rendez-vous', 'Consultation', and 'Pages'. The main content area is a form for contacting a doctor. It includes the following elements:

- A header 'HEALTHLINE' in blue.
- Navigation links: 'Home', 'Dossier Médical', 'Rendez-vous', 'Consultation', 'Pages'.
- Form fields: 'Votre Nom', 'Votre Email', 'Subject', and 'Message'.
- A blue button labeled 'Envoyer Message'.
- A semi-transparent 'Outil Capture d'écran' (Screenshot Tool) window overlaid on the form.
- Footer text: 'shortcuts | Map data ©2022 Google, INEGI | Ter'.

FIGURE 5.20 – Interface de "Contacter le médecin"

5.3 Conclusion

Dans ce chapitre nous avons vu le deuxième processus de développement d'une Ligne de Produit, ingénierie d'application, ce processus se répète pour chaque nouvelle application (membre de la LdP). Dans notre projet de fin d'étude nous avons choisi le cas d'une application d'E-Health pour une cabinet de diabétologie.

Conclusion Générale

Le travail proposé dans Ce mémoire de fin d'étude est ""Une ligne de produits orientée service pour les plateformes de Télé-consultation Médicale", un produit d'environ quatre mois de recherche et de travail. Nous avons donc mis en évidence trois approches : l'approche de la ligne de production de logiciels, l'approche orientée service et, bien sûr, l'approche e-santé. Nous avons donc effectué une revue bibliographique des trois approches de la maîtrise des concepts de base.

Après avoir examiné l'étude bibliographique, nous avons commencé le processus de développement de notre ligne de produit pour le domaine d'application «e-health». Ce processus est divisé en deux processus : ingénierie de domaine et l'ingénierie de l'applications. Le deuxième processus de l'application d'E-Health pour une cabinet diabetologie «CABINET DIABETOLOGIE HEALTHLINE», et ce processus a été simplifié par les résultats du premier processus, tout en utilisant des services web.

Il est à noter que ce travail nous a permis, d'une part, de lancer un nouveau domaine de recherche appelé les lignes de produit logiciel. D'autre part, approfondir et appliquer nos connaissances dans le développement d'applications orientées service.

Enfin, au cours de ce travail, nous avons examiné certains points de vue particuliers :

- Test des services web réutilisables en dérivants d'autres membres de la LdP ;
 - Implémentation des variantes qui peuvent être utilisés par d'autres membres de notre LdP ;
- Le développement d'une méthode de dérivation des membres de la LdP qui pourra être automatique ou semi automatique.

Bibliographie

- [1] Ammiche, T. (2019). *La télémédecine comme levier de lutte contre les déserts médicaux Cas : CHU de TO*. PhD thesis, Université Mouloud Mammeri.
- [2] Bakhta, A. (2015). *MÉTHODE DE RECHERCHE DE SERVICES WEB BASÉE SUR L'ANALYSE FORMELLE DE CONCEPTS*. PhD thesis, Université de La Manouba.
- [3] Baujard, C. and Hamouda, I. B. (2015). La gestion de projet à l'hôpital : dossier patient informatisé et qualité de soins. *Recherches en sciences de gestion*, 109(4) :147–164.
- [4] Benboudriou, M. and Kouah, S. (2019). Conception et réalisation d'un systèmes d'iot (internet of things) basé agents pour le suivi des patients dans le cadre e-santé.université oum el bouaghi.
- [5] Benmerzoug, D. (2009). *Modèles et outils formels pour l'intégration d'applications d'entreprises*. PhD thesis.
- [6] Berry, G. (2017). *L'hyperpuissance de l'informatique : algorithmes, données, machines, réseaux*. Odile Jacob.
- [7] Bosch, J. (2000). *Design and use of software architectures : adopting and evolving a product-line approach*. Pearson Education.
- [8] Clements, P. (2001). *Northrop. Software Product Lines : Practices and Patterns*. Addison-Wesley, August.
- [9] Cousin, T. (2021). Déploiement de la télémédecine en france, des opportunités pour l'officine. *Actualités Pharmaceutiques*, 60(607) :20–25.
- [10] de Santé, H. A. (2013). Efficience de la télémédecine : état des lieux de la littérature internationale et cadre d'évaluation. *Rapport d'évaluation medico-economique*. Paris : Haute Autorite de Sante.
- [11] DEHMECHE, A. (2016). *Service web Embarqué (Technologie SOAP sous Android) pour la gestion d'une Bibliothèque MI*. PhD thesis, FACULTE DES MATHÉMATIQUES ET DE L'INFORMATIQUE-UNIVERSITE MOHAMED BOUDIAF-M'SILA.
- [12] Dufour Coppelani, D. (2016). *E santé, étude de la politique informationnelle et communicationnelle du public et du privé dans le cadre et l'enjeu des informations médicales*

personnalisées. PhD thesis, Aix-Marseille.

- [13] EDOUARD, A. (2014). Comprendre l'architecture des web services rest.[consulté le 28.06. 2016]. *Disponible à l'adresse : http://miage.unice.fr/@api/deki/files/2369/=webser-vice_REST.pdf*.
- [14] Foudil, I. (2017). *L'expérience de télémédecine en Algérie : état des lieux et perspectives Cas du CHU de Tizi-Ouzou*. PhD thesis, Université Mouloud Mammeri.
- [15] Gambarotto, P. (2009). Technologies pour web services faciles : Rest, json. *Les Journaux Réseaux JRES 2009*, page 43.
- [16] Group, I. A. W. et al. (2000). Ieee recommended practice for architectural description of software-intensive systems. ieee standard ieee std 1471-2000. *IEEE Standards Association*.
- [17] Guo, Y., Deng, F., and Yang, X. (2018). Design and implementation of real-time management system architecture based on graphql. *IOP Conference Series : Materials Science and Engineering*, 466 :012015.
- [18] Jean, C. (2009). Les lignes de produits logiciels réutilisation et variabilité. *Publication périodique de Smals*.
- [19] Krueger, C. W. (1992). Software reuse. *ACM Computing Surveys (CSUR)*, 24(2) :131–183.
- [20] Larab, A., Leloup, J., Gaucher, P., and Martineau, P. (2003). Description des conséquences du développement des tic dans l'habitat. *Laboratoire Informatique de l'Université de Tours, Rapport Interne*, (265x).
- [21] Lasbordes, P. and de l'Essonne, D. (2009). *La télésanté : un nouvel atout au service de notre bien-être*. La Documentation Française.
- [22] Le Calvar, T., Chhel, F., Jouault, F., Saubion, F., and Syriani, E. (2021). Génération d'ensembles de modèles explorables par couplage de contraintes et de transformation de modèles. In *Journée Francophones de Programmation par Contraintes (JFPC) 2021*.
- [23] Levy, F. (2010). How technology changes demands for human skills. oecd education working papers, no. 45. *OECD Publishing (NJI)*.
- [24] Liang, Z. H., Rongwong, W., Liu, H., Fu, K., Gao, H., Cao, F., Zhang, R., Sema, T., Henni, A., Sumon, K., et al. (2015). Recent progress and new developments in post-combustion carbon-capture technology with amine based solvents. *International Journal of Greenhouse Gas Control*, 40 :26–54.
- [25] Malki, A. et al. (2015). *Modélisation sémantique du cloud computing : vers une composition*

de services DaaS à sémantique incertaine. PhD thesis.

- [26] Mohammed, A. (2009). Interopérabilité des services web hétérogène. *Mémoire de projet de fin d'étude, université de Tlemcen*.
- [27] Morillon, C. and Plait, L. (2020). Dynamique de la communication en chirurgie robotique., *published by Elsevier*.
- [28] Muneene, D. (2020). ehealth solutions in global health. *Handbook of Global Health*, pages 1–15.
- [29] Neggab, M. and Aissani, N. (2020). *Conception et Réalisation d'une Plateforme Web de Téléconsultation Médicale*. PhD thesis, Université Mouloud Mammeri.
- [30] Nesrine, L. and Bennouar, D. (2018). On the use of model transformation for the automation of product derivation process in spl. *Acta Universitatis Sapientiae, Informatica*, 10 :43–57.
- [31] Obbink, H. and Pohl, K. (2005). *Software Product Lines : 9th International Conference, SPLC 2005, Rennes, France, September 26-29, 2005, Proceedings*, volume 3714. Springer.
- [32] Papazoglou, M. P. (2003). Service-oriented computing : Concepts, characteristics and directions. In *Proceedings of the Fourth International Conference on Web Information Systems Engineering, 2003. WISE 2003.*, pages 3–12. IEEE.
- [33] Petersen, K., Zaha, J., and Metzger, A. (2007). Variability-driven selection of services for service compositions. pages 388–400.
- [34] Simon, P. and Lucas, J. (2014). La télémédecine n'est pas du e-commerce mais de la médecine clinique. *European Research in Telemedicine/La Recherche Européenne en Télémédecine*, 3(1) :27–34.
- [35] Sini, G. (2013). *Méthodes et outils pour la gestion des workflow-Modélisation ontologique des processus pour l'analyse*. PhD thesis, Université Mouloud Mammeri.
- [36] Sperling, E. A. (2016). *Etude de la relation entre médecins généralistes et infirmiers libéraux dans l'Oise pour la prise en charge des plaies chroniques*. PhD thesis, Éditeur inconnu.
- [37] Taleb, A. and Amrouche, Y. (2020). *Conception et Réalisation d'Une Application Mobile de Gestion des absences des étudiants*. PhD thesis, Université Mohamed el-Bachir el-Ibrahimi Bordj Bou Arréridj Faculté de ...
- [38] TIELLE, C. E. P. and BLAL, R. (2018). Identification des services ti à partir de modèles de processus d'affaires : une approche par la modélisation d'affaires.

- [39] Voilmy, D., Guichard, R., Olivier, M., Tellier, V., Schoevaerds, D., and Novella, J. L. (2021). Quelles pratiques de télésurveillance de maladies chroniques dans les espaces à dominante rurale peu densément peuplés? In *JETSAN 2021-Colloque en Télésanté et dispositifs biomédicaux-8ème édition*.
- [40] Wahabi, M. (2011). *REALISATION D'UN SITE WEB DYNAMIQUE COMMERCIALE*. PhD thesis, Université Virtuelle de Tunis.
- [41] Yu, J. (2010). *Ligne de produits dynamique pour les applications à services*. Theses, Université Joseph-Fourier - Grenoble I.
- [42] Yu, J. (2023). *Ligne de produits dynamique pour les applications à services*. Article, Université Joseph-Fourier - Grenoble I.