

Table de matière

Introduction générale.....	1
Section I : Etude Préalable	4
Chapitre I : Cadre du projet.....	5
1. Introduction	5
2. Préliminaires.....	5
3. Présentation de la société d'accueil EL AMED	6
3.1. Historique du Groupe EL Amed	6
3.2. Activités	7
3.3. Relation internationale	8
3.4. Contact	8
3.5. Siège du groupe.....	9
4. Motivation	9
5. Mise en situation de notre projet	10
5.1. Architecture générique notre projet	11
5.2. Diagramme de Gantt	11
6. Spécification des besoins fonctionnels et non fonctionnels	13
6.1. Les acteurs	13
6.2. Les besoins fonctionnels	14
6.3. Les besoins non fonctionnels	15
7. Proposition d'une maquette d'un prototype	16
8. Conclusion.....	16
Chapitre II : Analyse du domaine.....	17
1. Introduction	17
2. Importance de l'analyse du domaine.....	17
3. Le choix de méthodologie de conception.....	18
4. Diagramme de navigation	18
5. Diagrammes des cas d'utilisation.....	18
5.1. Diagramme de cas d'utilisation Global.....	19
5.2. Diagramme de cas d'utilisation « Gestion des Modèles ».....	20
5.3. Diagramme de cas d'utilisation « Gestion des Destinataires ».....	20
5.4. Diagramme de cas d'utilisation « Planification des Tâches »	21

5.5.	Diagramme de cas d'utilisation « Envoi SMS»	21
6.	Description textuelle des Cas d'utilisation	22
6.1.	Scénario de cas d'utilisation « S'authentifier »	22
6.2.	Scénario de cas d'utilisation « choisir modèle »	23
6.3.	Scénario de cas d'utilisation « Ajouter Modèle »	24
6.4.	Scénario de cas d'utilisation « Envoyer SMS »	24
7.	Diagrammes d'activités	25
7.1.	Diagramme d'activité de « l'authentification »	25
7.2.	Diagramme d'activité de « Envoi SMS»	26
8.	Conclusion.....	27
Section II : Etude Conceptuelle.....		28
Chapitre III : Conception de l'ontologie		29
1.	Introduction	29
2.	Etat de l'art ontologique	29
2.1.	Etude de l'existant.....	29
2.2.	Ontologie et web sémantique	29
2.3.	Définition d'une Ontologie	34
2.4.	Composantes d'une ontologie	34
2.5.	Les trois sous langages de l'ontologie	35
2.6.	Types d'ontologies	37
2.7.	Cycle de vie d'une ontologie	39
2.8.	Etat de l'art des méthodes de conception des ontologies.....	40
2.9.	Un exemple de domaine d'application de l'ontologie	42
3.	Conception de notre ontologie	44
3.1.	L'ontologie vCard (Visit Card).....	45
3.2.	L'ontologie Nepomuk	45
3.3.	L'ontologie time.....	46
3.4.	Ontologie SMS.....	46
3.5.	Ontologie finale	46
3.6.	Notre modèle ontologique.....	46
4.	Diagramme de classes de l'ontologie	51
5.	Interprétation du diagramme de classes de l'ontologie	53
6.	Conclusion.....	53
Chapitre IV : Conception & Interfaçage du futur Système		54
1.	Introduction	54

2. Etude dynamique du système global	54
2.1. Diagramme de séquence	54
2.2. Diagramme de collaboration	56
3. Etude statique du système global	57
3.1. Diagramme de classes	57
3.2. Diagramme de déploiement	58
4. Architecture & interfaçage avec l'ontologie	59
5. Conclusion.....	60
Section III : Mise en Œuvre	61
Chapitre V : Développement & Intégration	62
1. Introduction	62
2. Architecture de l'application	62
3. Environnement de développement	63
3.1. Environnement Logiciel.....	63
3.2. Choix techniques.....	65
4. Les serveurs utilisés	66
4.1. Serveur Fuseki V2.....	66
4.2. Triple Store	66
4.3. Serveur Tomcat V8	67
5. Le Framework Bootstrap.....	68
6. Conclusion.....	68
Chapitre VI : Vérification & Tests	69
1. Introduction	69
2. Les interfaces réalisées	69
2.1. Interface de présentation	69
2.2. Interface d'authentification :.....	69
2.3. Interface de gestion des tâches :.....	70
3. Tableau de données de test.....	71
4. Conclusion.....	71
Section IV : Retour d'expérience d'une élève ingénieur	72
1. Introduction	73
2. Outils de travail collaboratif.....	73
2.1. Slack.....	73
2.2. Gmail et google Drive.....	74
2.3. SVN : Subversion Number	74

3. Les réunions	75
4. Problèmes rencontrés	75
5. Conclusion.....	76
Conclusion générale & Perspectives	77
References	78
Annexes	79

Liste des figures

Figure 1 : Proportion d'individus utilisant le SMS [12]	6
Figure 2 : Relation avec les sous-systèmes existants	10
Figure 3 : Architecture générique de notre projet	11
Figure 4: Diagramme de Gantt.....	12
Figure 5: Prototype d'une interface d'envoi d'SMS	16
Figure 6 : Sondage des outils de modélisation [11]	17
Figure 7: Diagramme de navigation.....	18
Figure 8 : Diagramme des cas d'utilisation Global.....	19
Figure 9 : Diagramme de cas d'utilisation « gestion modèle ».....	20
Figure 10 : Diagramme de cas d'utilisation « Gestion Destinataire »	20
Figure 11 : Diagramme de cas d'utilisation « gestion des tâches »	21
Figure 12 : Diagramme de cas d'utilisation « Envoi SMS ».....	22
Figure 13 : Diagramme d'activité de « l'authentification »	26
Figure 14 : Diagramme d'activité de l'envoi de SMS	27
Figure 15: Les différentes versions du Web.....	30
Figure 16: Evolution du web de 1.0 jusqu'à le web 4.0 [2]	31
Figure 17 : Pile de standardisation [7]	32
Figure 18 : Exemple de représentation graphique d'une information en RDF	33
Figure 19 : Les trois sous langages d'OWL.....	37
Figure 20: Classification des ontologies selon Guarino [2]	39
Figure 21 : Cycle de vie de l'ontologie [2]	40
Figure 22 : Exemple de code SWRL en XML [6]	44
Figure 23 : Les Ontologies réutilisés.....	45
Figure 24: Les ontologies importées sous protégé.....	46
Figure 25 : Graphe de l'ontologie SMS généré par protégé	47
Figure 26 : Graphe de l'ontologie généré avec OWLGrE	52
Figure 27 : Diagramme de séquence de cas utilisation d'ajout de modèle	54
Figure 28: Diagramme de séquence de cas d'utilisation « s'authentifier »	55
Figure 29: Diagramme de séquence de cas d'utilisation « modification Modèle »	55
Figure 30 : Diagramme de séquence de cas d'utilisation « Supprimer Modèle ».....	56

Figure 31: Diagramme de collaboration de cas d'utilisation « Modifier modèle »	56
Figure 32 : Diagramme de collaboration de cas d'utilisation « Supprimer modèle ».....	57
Figure 33: Diagramme de classe du système	58
Figure 34: Diagramme de déploiement.....	59
Figure 35: Outils d'intégration des Ontologies dans des applications [12]	60
Figure 36 : Architecture 3 tiers de notre Système.....	62
Figure 37 : Système d'exploitation	63
Figure 38 : logo eclipse	63
Figure 39 : Logo de protégé	64
Figure 40: Interface d'accueil de serveur apache Jena Fuseki V2	66
Figure 41: Un exemple de triplet	67
Figure 42 : Requête SPARQL.....	67
Figure 43: Résultat de la requête SPARQL	67
Figure 44 : Logo de Tomcat V8	68
Figure 45 : Logo de Bootstrap	68
Figure 46 : Interface de présentation de service web	69
Figure 47: Interface d'authentification.....	70
Figure 48 : Interface de gestion des tâches	71
Figure 49 : Compte Slack.....	73
Figure 50: Compte Gmail de stagiaires.....	74
Figure 51: Rapport de jour	74
Figure 52: Logo subversion	74
Figure 53 : Réponse de la part de Comment Ça Marche	75
Figure 54 : Fichier de configuration de serveur Fuseki	79
Figure 55: Démarrage du serveur Fuseki	79

Liste des Tableaux

Tableau 1 : Scénario de cas d'utilisation « S'authentifier »	23
Tableau 2 : Scénario de cas d'utilisation « choisir Modèle »	23
Tableau 3 : Scénario de cas d'utilisation « Ajouter modèle »	24
Tableau 4 : Scénario de cas d'utilisation « Ajouter modèle »	25
Tableau 5 : La désignation de RDF [2].....	33
Tableau 6 : Tableau comparatif OWL DL, Lite et Full [12]	36
Tableau 7: Tableau comparatif des quelques raisonneurs [6].....	43
Tableau 8 : Propriétés d'objets	49
Tableau 9 : Propriétés de donné.....	50
Tableau 10 : Identification ontologie-UML.....	51

INTRODUCTION GÉNÉRALE

Le marché des téléphones mobiles a connu une véritable explosion depuis plus d'une décennie et ne cesse d'élargir sa cible entre personnes physiques et personnes morales ou organismes. La diffusion de l'information utile dans les entreprises tel que les informations financières, les informations commerciales, les informations organisationnelles, les informations relevant des ressources humaines, les informations environnementales, en plus de l'information personnelle ou privée constituent une dynamique d'échange. Cet échange d'information est indispensable pour la vie de toute entreprise qui est un système ouvert. C'est pour cela que l'entreprise est vue comme un système d'information qui évolue dans et avec un méta-système d'information plus grand qui est son environnement (fournisseurs, clients, partenaires, concurrents, organismes législateurs, gouvernement, société civile, ...). L'évolution d'une entreprise dans son environnement est désormais fortement rythmée par des phénomènes de modes et de tendances technologiques et communicationnelles.

Le premier de ces tendances technologiques est le World Wide Web qui était perçu dans les années 90 comme étant un système d'information publique constitué d'un ensemble de documents multimédias liés par des hyperliens, il est connu sous le nom Web 1.0. Le rôle de l'internaute était limité à une simple navigation sur le web. Une nouvelle version du 3W appelé web 2.0 a vu le jour en 2005, la vision du web2.0 considère l'internaute comme un élément actif qui participe à la production, la diffusion, la consultation du contenu du web. Le web 2.0 se distingue par une panoplie de technologies, nous en citons les wikis, les blogs, les flux RSS, les réseaux sociaux...Mais cette amélioration a continué à révéler les interminables limites. En effet, le sens des informations, les déductions, les constatations et les inférences pour dire en gros –la sémantique– échappe aux serveurs et aux logiciels qui constituent le web 2.0, ils sont encore dans les cerveaux des humains. S'ajoute la redondance, l'inutilité de l'information, le manque de sécurité et le problème de droit d'auteur.

La nouvelle version du web qui est le web 3.0 est venue palier à ces faiblesses par l'ajout du concept d'interopérabilité vis-à-vis de tout type de matériel et de logiciel, en plus de la mise à disposition d'informations compréhensibles par l'utilisateur et par la machine en introduisant le concept de web sémantique qui se base sur la définition d'informations sur les informations,

celles-ci sont dites des métadonnées. Ces métadonnées tentent de rendre l'information compréhensible par la machine. Il va de soi que toutes les applications de cette nouvelle ère de web doivent être en conformité avec les recommandations du W3C. Ces recommandations tournent autour de l'ouverture aux bases de données hétérogènes, la portabilité, la mobilité et l'accessibilité.

C'est dans ce sens que s'oriente notre sujet réalisé au sein de la société EL AMED. Il consiste en la notification par SMS basé sur une ontologie. Il s'agit d'implémenter un système qui permet à un utilisateur de manipuler une liste de contacts, une liste de modèles de messages et une liste de tâches planifiées. Pour envoyer un SMS, une importation des destinataires est indispensable. Tout le système peut être hérité et réutilisé par divers acteurs de la société EL AMED.

A l'occasion de la conception et du développement de ce projet, divers outils et technologies ont été adoptés. La plateforme Java EE pour le développement de notre application, le Framework apache JENA pour la manipulation de l'ontologie sous eclipse, l'API d'un des opérateurs téléphoniques tunisiens (Tunisie Telecom, Ooredoo, Orange) pour servir comme interfaçage entre l'application et les abonnés, le JSP (Java Server Pages), le RDF (Ressource Description Framework), l'OWL (Web Ontology Language), protégé pour créer notre modèle ontologique et l'Atelier de Génie Logiciel Rational Rose supportant le langage de modélisation UML.

L'avancement dans les étapes du cycle de vie de ce projet est rythmé par trois sections chacune comprenant deux chapitres. Ces sections sont l'analyse du domaine, la conception et le développement, tests et vérifications. Ils sont davantage explicités dans ce qui suit :

Le premier chapitre est dédié à la mise en contexte de notre projet par la présentation de la société EL AMED. Le deuxième chapitre est dédié pour la rédaction détaillée du cahier des charges. Le troisième chapitre est un chapitre central, contient un état de l'art qui détaille les concepts de web sémantique en vue de concevoir notre nouvelle ontologie, ce chapitre lie aussi à la conception orientée objet de notre nouveau système d'envoi de SMS et la partie consacrée à notre ontologie. Le quatrième chapitre est dédié pour la conception UML de l'application dans sa globalité. Le cinquième chapitre décrit une expérience pratique par l'implémentation de notre nouveau système. Le sixième chapitre représente les vérifications et les tests nécessaires pour valider les fonctionnalités, la fiabilité et la robustesse de notre système. Enfin, le septième chapitre nous l'avons consacré pour relater notre expérience en tant qu'élève ingénieur qui débute son intégration dans la vie professionnelle, nous y avons cité les outils

utilisés pour travailler en équipe et les difficultés que nous avons rencontrés et comment nous les avons surmonté.

En guise de conclusion nous ne manquerons pas d'évaluer le degré d'atteinte des objectifs fixés dans le cahier de charge pour notre système de notification par SMS basé sur une ontologie et nous profiterons d'en lancer d'autres travaux.

SECTION I : ETUDE PRÉALABLE

CHAPITRE I : CADRE DU PROJET

1. Introduction

Dans ce premier chapitre nous mettons notre sujet dans son grand contexte qui est la communication par le biais d'envoi de SMS en montrant son importance sur le plan économique et social. Nous présentons notre entreprise d'accueil groupe EL Amed et nous formulerons dans le détail les objectifs de notre futur application.

2. Préliminaires

Différentes formes de communication ne cessent de se développer, de se diversifier et d'être perfectionnées chaque année, pour ne pas dire chaque jour, de nouvelles technologies sont inventées pour faciliter la transmission d'une information ce qui bouleverse sensiblement les moyens classiques de recherche, de persistance et de diffusion des informations. Ce qui bouleverse même l'adoption et la stabilisation d'une technologie, imposant un effort d'adaptation continu. Auparavant et actuellement, le téléphone reste le moyen de communication favorisé. Sous quelques aspects, il reste et restera probablement encore, le moyen le plus direct, le plus rapide et universel afin d'établir une communication ou d'avoir recours au service SMS (Short Messaging Service).

En 2011, le cabinet d'analyse américain, ABI RESARCH indique que 7000 milliards de SMS seront transmis dans le monde. Cette élévation du nombre d'SMS est d'autant plus liée à la démocratisation des forfaits mobiles intégrant des SMS en nombre illimité. En France, des statistiques montrent un décalage net par rapport à l'utilisation de l'internet et l'email, envoyer des messages écrits est une pratique plus ancrée dans les habitudes des Français qui utilisent de plus en plus ce moyen de communication.

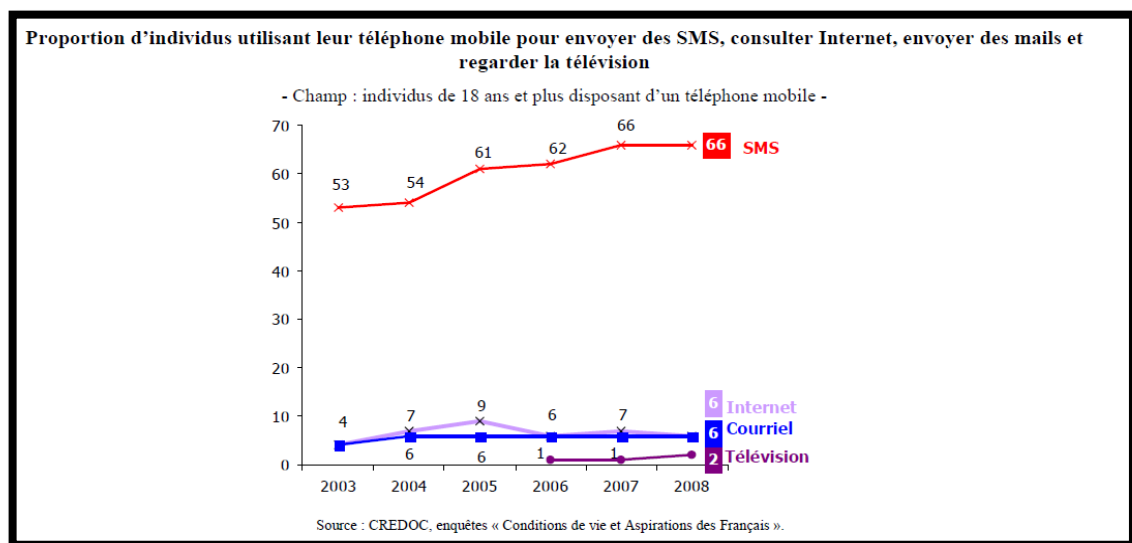


Figure 1 : Proportion d'individus utilisant le SMS [12]

En Tunisie, d'après un article publié dans La Presse de Tunisie le 17 décembre 2012, des millions de SMS sont échangés entre abonnés et les opérateurs de la téléphonie mobile. Un autre article publié dans Tunisie News le 05 Septembre 2009 indique que le taux de lecture des SMS est très important et varie entre 95% et 100%.

Le SMS utilisé dans le but de campagnes politiques, des campagnes de marketing (périodes de solde, de promotions, de liquidation, de jeux...), ou dans un but de sensibilisation de préservation de l'environnement, ou de diffusion de l'information éducationnelle (diffusion des résultats du bac, des concours nationaux, de relevés de notes, ...) EL AMED est en cours de profiter de cette technologie dont l'importance et la vitesse de ciblage est inouïe.

3. Présentation de la société d'accueil EL AMED

3.1. Historique du Groupe EL Amed

Une discussion entre un groupe d'enseignants en octobre 2004 a déclenché un rêve portant sur la promotion de la formation des sciences infirmières et la fondation d'une utopie crédible.

Le commencement de la réalisation de ce rêve a débuté par la création d'un institut de formation des cadres paramédicaux IFCP sous un slogan "Apprendre à soigner autrement".

La dernière réforme sanitaire en Tunisie a aidé à la concrétisation de ce projet par la naissance d'un Institut Supérieur des Sciences Infirmières EL AMED.

La formation académique a conféré un progrès de l'art infirmier et un changement radical de la qualité des soins prodigués en Tunisie et à l'extérieur.

Après une carrière pleine de succès, le groupe de promoteurs a décidé d'investir plus dans le secteur de l'enseignement privé et les services d'accompagnement social. Le groupe est présenté aujourd'hui par une gamme d'institutions de formation, des projets de service et d'accompagnement social.

3.2. Activités

Le groupe El Amed est constitué actuellement par plusieurs établissements d'enseignement privé ainsi que des entreprises spécialisées dans les domaines d'accompagnement social et d'agriculture.

3.2.1. El Amed pour l'enseignement supérieur

L'Institut Supérieur des Sciences Infirmières EL AMED a pour mission le développement et la promotion des connaissances théoriques et pratiques relatives à la discipline infirmière. Elle contribue à la formation des cadres paramédicaux capable de créer des pratiques de soins adaptées aux besoins de la société et capable de conquérir le marché international de l'emploi.

La formation à EL AMED se termine par une Licence appliquée en sciences infirmières Suite à une décision du Monsieur le Ministre de l'Enseignement Supérieur de la Recherche Scientifique et de la Technologie, l'institut est impliquée dans le système LMD à partir de l'année universitaire 2008- 2009 et prochainement :

- Master Recherche en sciences infirmières
- Master Management du risque infectieux
- Master Gestion de soins
- Master en Santé communautaire

3.2.2. El Amed pour la formation professionnelle

El Amed pour la formation professionnelle a pour mission de former des techniciens capables de conquérir le marché international de l'emploi dans des domaines multiples :

- Préparateur en Pharmacie
- Auxiliaire de Vie
- Comptable d'Entreprise
- Assistant de direction
- Commerce
- Commerce international

3.2.3. El Amed pour l'enseignement de base et secondaire

El Amed pour l'enseignement de base et secondaire est présenté par trois collèges situés à Khezama ouest (Sousse), Sahline et beni Hassen (Mounastir). Ainsi que deux lycées secondaires à Khezama ouest (Sousse), Sahline (Mounastir).

3.2.4. El Amed pour l'enseignement primaire

Présenté par l'école Primaire à Beni Hassen localisée à la place d'Abderrahman Dahmani rue Elmadra 5314 Beni Hassen.

3.2.5. El BOUSTAINE

Production de l'huile d'olive, polyculture, viticulture, maraîchage, arboriculture et toutes les cultures spécialisées. Les élevages traditionnels (chevaux, bovins, ovins, caprins ...)

3.2.6. E-SAAAD

Le projet de service e-SAAD vu le jour pour rendre humage aux citoyens tunisiens qui ont beaucoup mérité.

Notre objectif est d'assurer un processus d'accompagnement de la personne pour atteindre un but de santé qui est le mieux-être.

3.3. Relation internationale

En vue de l'internationalisation de cette formation universitaire l'institut EL AMED tend à créer des relations de jumelage, d'échange des stagiaires, des formateurs et des connaissances avec des autres institutions dans le monde surtout à travers le BTEC au Canada.

3.4. Contact

- Adresse : 03 Avenue de la Poste Sahloul 4054 Sousse.

- Tel /Fax : 73.820.800 - 73.822.800 - 73.822.822
- E-mail : contact@universite-amed.com
- Site web : <http://www.universite-amed.com>

3.5. Siège du groupe

- Adresse : 03 Avenue de La Poste, Sahloul 4054 Sousse
- Tél: (+216) 73 820 800 --- (+216) 73 822 800
- GSM: (+216) 21 175 185 --- (+216) 97 882 937
- Fax: (+216) 73 820 826
- E-Mail: contact@universite-amed.com
- Site Web: www.universite-amed.com

4. Motivation

La communication et l'échange des messages téléphoniques demeurent une pratique quotidienne, malgré l'évolution technologique (notifications par applications mobiles, envoi de mails, messages et notifications à travers les réseaux sociaux...), il y a encore des personnes qui ne les utilisent pas. Le problème revient à plusieurs raisons partagées entre trois acteurs principaux : l'entreprise concernée émettrice de l'information (comme dans notre cas EL AMED), les intermédiaires d'émission de l'information à savoir les fournisseurs de services Internet et les opérateurs téléphoniques. Et enfin les récepteurs de l'information utile à savoir les citoyens ou consommateurs ou clients dépendant de leurs relation avec l'entreprise. Les entraves sont donc partagés entre eux et se déclinent par le coût des services internet, la couverture, la mentalité, l'expérience de l'utilisateur... C'est l'entreprise qui doit donc s'adapter et doit imposer.

Vu le nombre croissant de ses partenaires entre personnels, étudiants, élèves, parents d'étudiants, parents d'élèves, enseignants, fournisseurs, partenaires pédagogiques et scientifiques ... et vue sa dynamique relationnelle communicationnelle, groupe El AMED s'est trouvé dans l'obligation d'instaurer un système dynamique et évolutif pour supporter n'importe quel modèle d'information d'une manière automatique et rapide.

Notre système subviendra aux besoins de EL AMED en terme de la rapidité d'échange d'information diversifiée et de faciliter la communication entre les différents acteurs su-cités, pour une entreprise commerciale, contacter des clients par SMS pour dire par exemple « merci de payer la deuxième tranche de votre inscription » serait moins coûteux que de passer des appels téléphoniques, il serait plus rapide de transférer des messages urgents. Au fil des ans les notifications par e-mail se sont révélées lentes et peu efficaces, car le taux de connections Internet des gens est très fluctuant. La question était de fournir un outil de communication efficient.

5. Mise en situation de notre projet

Notre projet de notification par SMS en se basant sur la définition d'une ontologie est un module circonscrit au sein du système d'information global de la société EL AMED. On en cite les sous-systèmes de gestion de la scolarité, de la gestion des inscriptions, gestion des stages, gestion des formations, gestion des examens (voir la figure 2). Notre système de notification par SMS basé sur l'ontologie doit être capable d'échanger avec les sous-systèmes existants et doit s'apprêter à l'intégration de futurs nouveaux modules pour garantir l'évolutivité et respecter la dynamique du système d'information général.

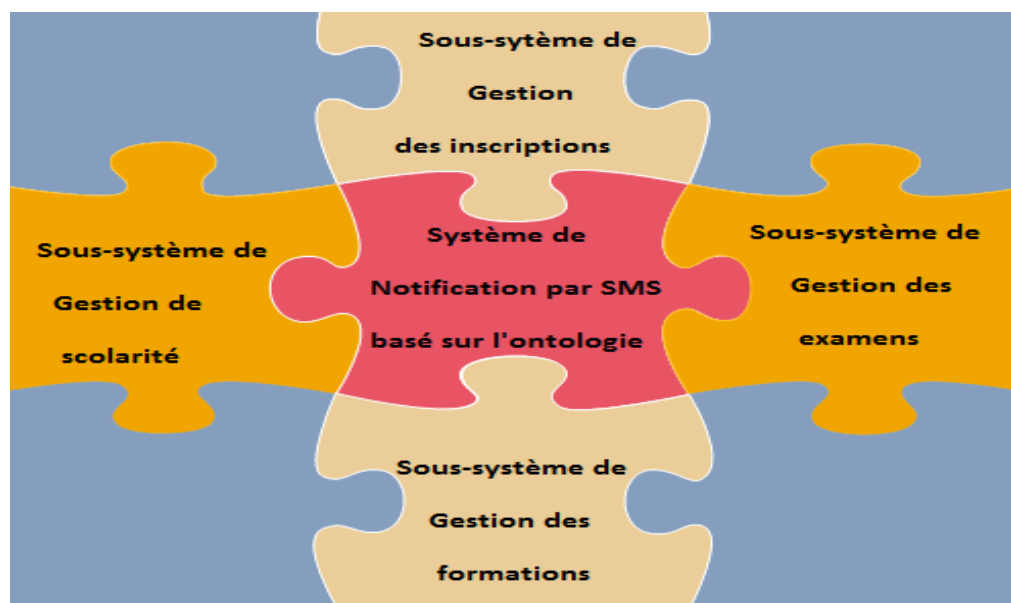


Figure 2 : Relation avec les sous-systèmes existants

5.1. Architecture générique notre projet

Notre système qui est « un système de notification basé sur une ontologie » permet à un administrateur d'envoyer un sms vers un destinataire (contact), Tunisie télécom est l'opérateur qui prendra charge pour nous fournir un interfaçage avec les contacts. La figure 3 illustre cette architecture.

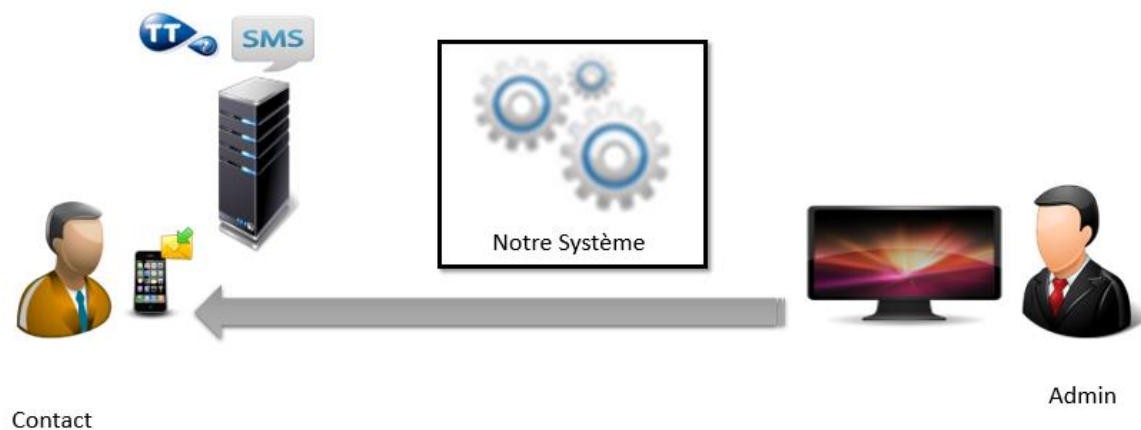


Figure 3 : Architecture générique de notre projet

5.2. Diagramme de Gantt

Le diagramme de GANTT dans la figure 4 représente la planification de différentes tâches durant la période du stage qui dure quatre mois.

Chapitre 1 : Cadre du projet

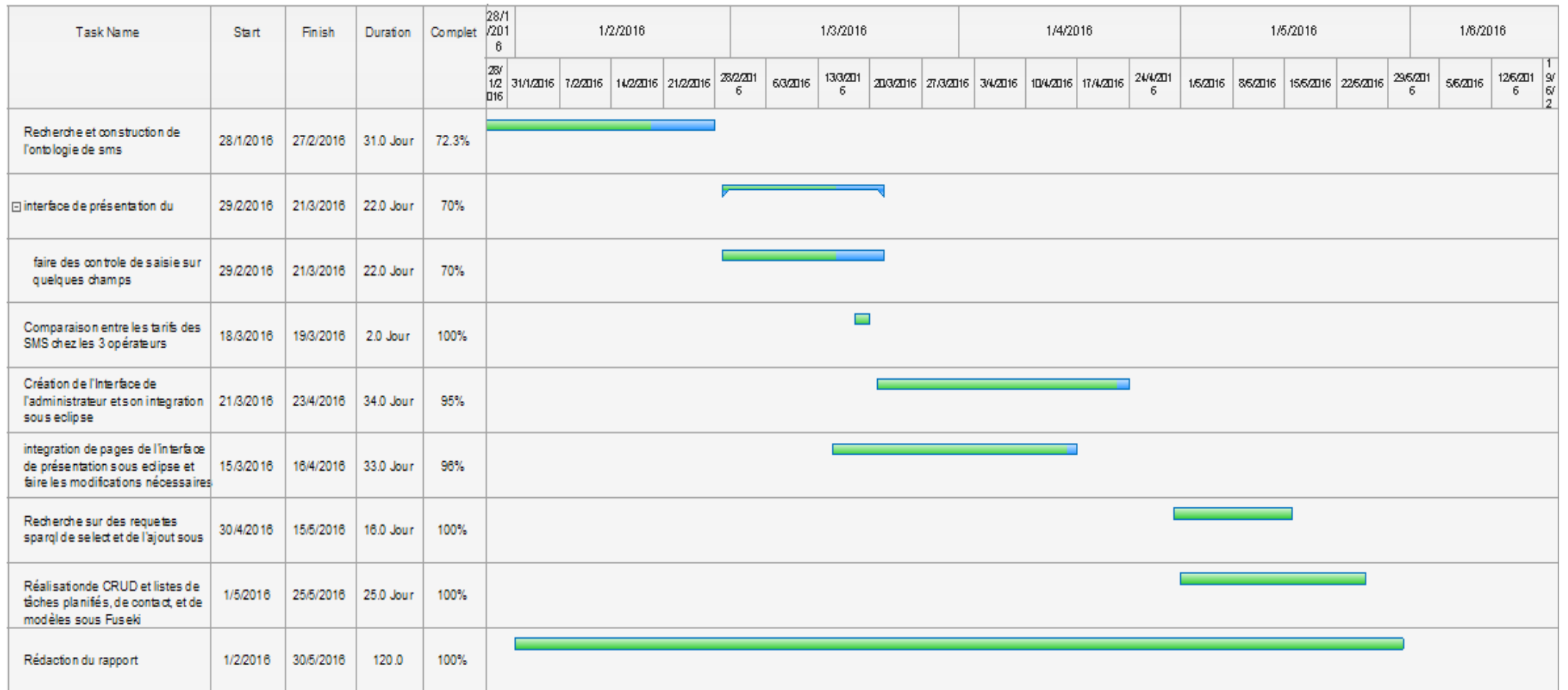


Figure 4: Diagramme de Gantt

6. Spécification des besoins fonctionnels et non fonctionnels

Pour transmettre une information, l'utilisation des SMS reste toujours indispensable, et ça revient à la facilité de son utilisation et la couverture de réseau 2G et 3G soit suffisante.

L'utilisateur de notre système en cas de service web, pour qu'il puisse envoyer des SMS, est invité à s'authentifier s'il a déjà un compte sinon il envoie une demande de création de compte au super-admin qui sera chargé de créer un compte au client demandeur de service. Dans le cas d'un client EL AMED, l'utilisateur n'aura pas besoin de posséder un compte. En cliquant sur le bouton « se connecter », une interface d'accueil s'affiche et il peut faire l'activité souhaitée.

Exemple : Pour faciliter la tâche de réception de documents dont l'étudiant a besoin, nous avons proposé de mettre en place une solution d'envoi par SMS à sa destination sans qu'il soit obligé de se déplacer.

6.1. Les acteurs

6.1.1. Le super-admin

Qui est le superviseur de toute activité sur le système (activité de gestion des modèles, des contacts, d'envoi d'SMS, etc.).

6.1.2. Le Client

Dont le rôle est de gérer son propre système, les mêmes fonctionnalités que le super-admin sauf il ne peut pas créer des comptes.

6.1.3. Un système externe

Qui consomme le service web. C'est un client autre que le client EL AMED.

Remarque :

- Le client après avoir créé un compte, devient à son tour un administrateur de son système. Donc, il peut avoir l'appellation d'un administrateur
- Lorsqu'on parle de contact c'est la même chose que destinataires, c'est-à-dire synonymes.

6.2. Les besoins fonctionnels

Notre application doit répondre à plusieurs besoins. Elle devra principalement permettre à nos acteurs de faire les opérations suivantes :

6.2.1. Super-administrateur

- **Gestion des modèles d'information :** Ajouter, modifier et supprimer des modèles d'information. On appelle modèle d'information un texte un message saisi par l'expéditeur contenant une information qui concerne des personnes bien déterminés on parle ici du publipostage ou standard qui ne contient aucune indication sur une personne précise.

Exemple : « Bonjour Mr, merci de payer votre dernière tranche avant le 04/09/2016 »

- **Gestions des tâches planifiées :** Ajouter des tâches planifiées, modifier des tâches planifiées, supprimer des tâches planifiées, prioriser des tâches. On désigne par tâche planifiée toute évènement ayant une date bien déterminé que notre système peut détecter cette date envoyer automatiquement les SMS aux destinataires concernés par cette évènement.
- **Gestion de différents destinataires :** Ajouter des destinataires, modifier des destinataires, supprimer des destinataires, importer et sauvegarder des destinataires. On désigne par destinataires les étudiants, les élèves de EL AMED, leurs parents ..., les destinataires propres au système externe (via un service web).
- **Elaborer des statistiques sur le nombre des SMS envoyés :** un administrateur peut élaborer des statistiques
- **Création d'un compte** pour un client (Système externe qui est un client consommateur de service web).

6.2.2. Client : client EL AMED

- Gestion des modèles d'information, des tâches planifiées, des destinataires
- Envoyer SMS en se basant sur un modèle.
- Le client importe un fichier (XML ou RDF : Ressource Description Framework) contenant une des données de notre base de connaissance.
- Le client doit choisir un modèle parmi les modèles existants, sinon créer un nouveau modèle.

- Le client doit activer un filtre qui permet de préciser les destinataires ciblés par un message bien déterminé.

6.2.3. Système externe

Le système externe fait les mêmes fonctionnalités que le client, en plus il doit s'inscrire pour qu'il puisse faire la gestion des Destinataires, des tâches planifiés et de modèles et l'envoi des SMS.

6.3. Les besoins non fonctionnels

Notre application future est une application web3.0 elle doit donc et tout d'abord être conforme aux recommandations W3C par le respect de l'accessibilité, par le respect de l'interopérabilité, par le respect d'un certain niveau de sécurité, par la divulgation et l'imposition du type de droit d'auteur.

Ensuite et vu que les besoins non fonctionnels représentent les exigences implicites auxquels le système doit répondre, l'application devra être cohérente au point de vue ergonomique, elle doit présenter une bonne architecture et une richesse informationnelle, elle doit être intuitive et offrir un help, elle doit de surcroit adhérer à la réglementation nationale. Nous détaillons dans ce qui suit certaines de ces exigences non fonctionnelles :

6.3.1. Contraintes ergonomiques

- Le système doit être lisible
- Le système doit être simple à utiliser par l'utilisateur.

6.3.2. Contraintes techniques

- L'application doit être fonctionnelle et disponible ;
- extensibilité du système ; notre système est évolutif à long terme.
- L'application doit garantir la sécurité : n'oublions pas que nous gérons dans notre base de connaissances les informations personnelles de nos divers acteurs, que la protection des informations personnelles est régie par des textes de loi à l'échelle nationale et internationale qui engendrent le déploiement de mécanismes de sécurité pour se conformer aux directives et réglementations ;

7. Proposition d'une maquette d'un prototype

L'interface ci-dessous représente une parmi les interfaces qu'on désire réaliser. Après l'importation du fichier et le choix de modèle, comme dans l'exemple suivant les champs civilité, nom et adresse seront remplacés par les données existantes dans le fichier contenant la liste des destinataires à l'instar d'un publipostage.

Cette proposition n'est qu'un scénario parmi d'autres que nous proposerons à la société qui choisira selon ses convenances la maquette qui sera implémentée. La figure 5 illustre cette maquette.

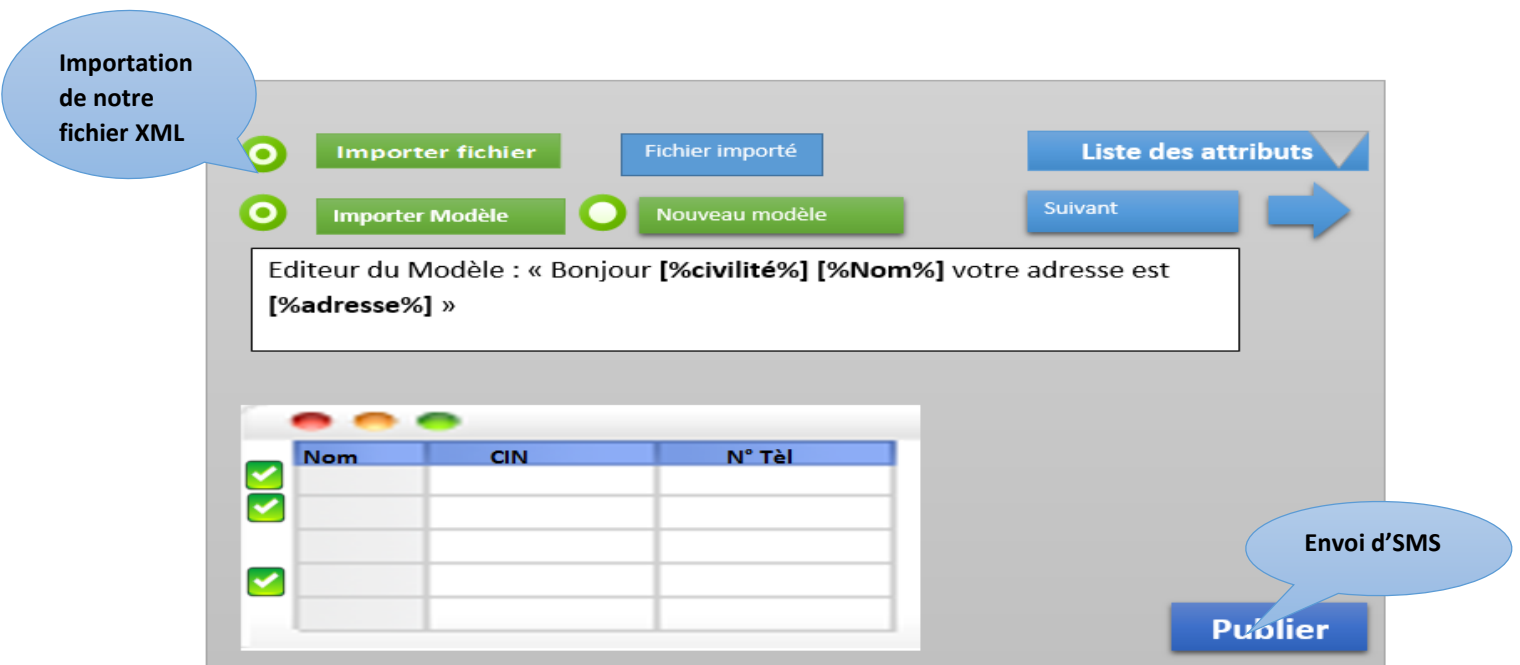


Figure 5: Prototype d'une interface d'envoi d'SMS

8. Conclusion

Tout au long ce chapitre nous avons présenté le cadre de notre projet de fin d'études. Nous avons pris conscience de son contexte général, nous avons identifié son intérêt pour le groupe EL AMED et sa localisation parmi le système d'information global de l'entreprise afin de bien le cerner d'une part, et afin de garantir son intégration et interfaçage avec les autres applications. L'identification des acteurs qui vont interagir avec notre futur système ainsi que la spécification de ses besoins fonctionnels et non fonctionnels nous illumineront le chemin pour proposer une analyse du domaine objet du prochain chapitre.

CHAPITRE II : ANALYSE DU DOMAINE

1. Introduction

Notre effort conceptuel requiert deux facettes importantes : Un premier effort conceptuel classique sera investi pour la totalité de l'application visée là où on aura recours au langage UML étant donné qu'elle est de nature interactive et relève du domaine du web. Et un deuxième effort conceptuel imposé par le fait que notre future application relève du domaine du web sémantique, nous aurons donc à lui modéliser une ontologie et on aura recours au langage dédié OWL. L'analyse du domaine tiendra compte donc de cette double facette de l'application.

2. Importance de l'analyse du domaine

L'analyse fonctionnelle d'UML est indispensable pour faciliter la compréhension du fonctionnement de notre système pour cela nous représentons dans ce chapitre quelques diagrammes à savoir le diagramme de navigation, de cas d'utilisation et d'activité, dans le chapitre suivant nous allons fournir les diagrammes de séquence, de collaboration, de déploiement. Le choix de l'outil Rational Rose émane d'une petite recherche sur le classement des outils de modélisation orientés objet. La figure ci-dessous montre que l'outil de notre choix accapare presque le quart du marché des utilisateurs.

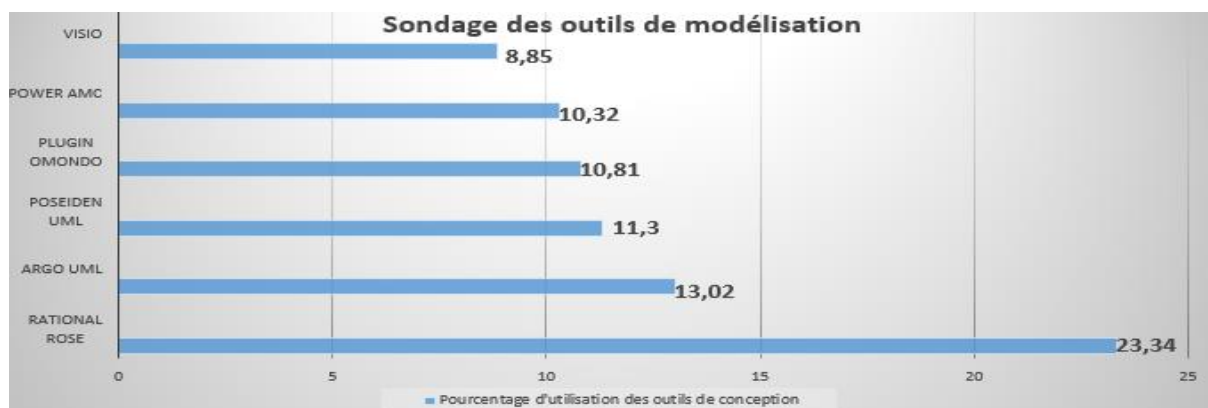


Figure 6 : Sondage des outils de modélisation [11]

3. Le choix de méthodologie de conception

Dans le cadre de la pratique, il est conseillé de choisir une méthodologie reconnue et efficace afin d'organiser les phases de développement et de garantir un maximum d'efficacité. Pour cela, nous avons adopté UML comme langage de modélisation, tout d'abord, puisque nous allons utiliser le concept de programmation orientée objet dans un Framework J2EE pour développer notre application.

4. Diagramme de navigation

Le diagramme de navigation permet d'aider à comprendre la manière de passage d'une page vers un autre. Il est directement déduit de la maquette proposée dans le chapitre précédent.

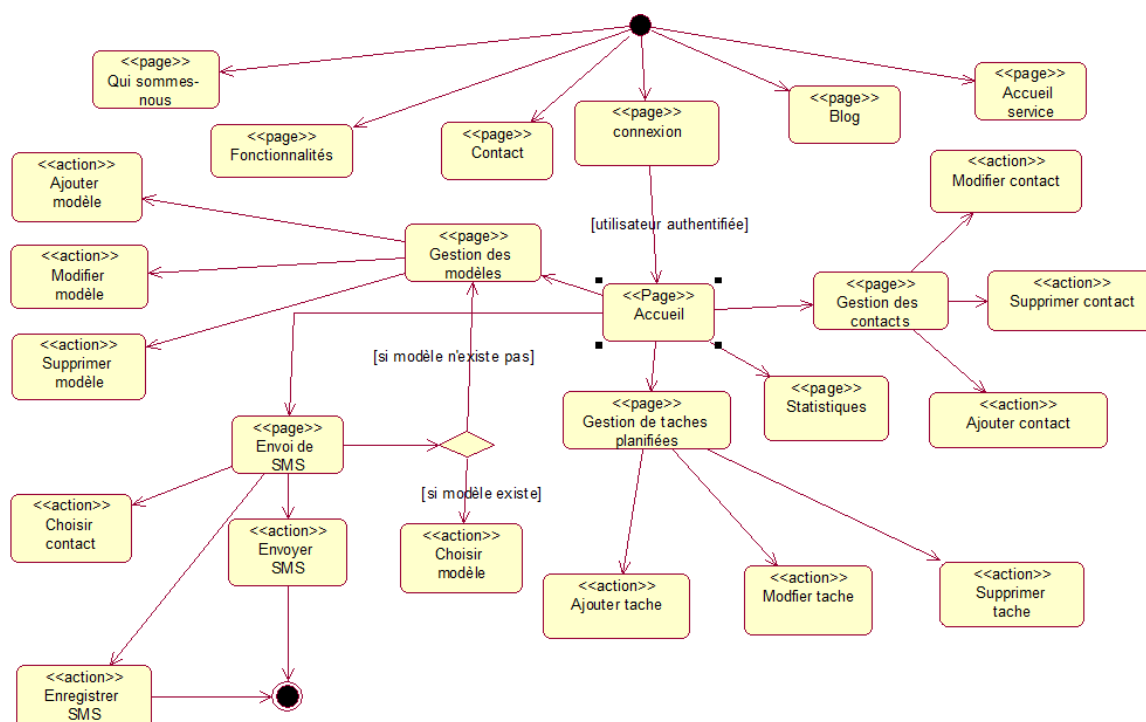


Figure 7: Diagramme de navigation

5. Diagrammes des cas d'utilisation

Les diagrammes de cas d'utilisation décrivent les principales fonctions d'un système et identifient les interactions entre le système et son environnement externe, représenté par des

acteurs. Ces acteurs peuvent être des personnes, des organisations, des machines ou d'autres systèmes externes. [3]

5.1. Diagramme de cas d'utilisation Global

Le diagramme ci-dessus représente le diagramme de cas d'utilisation global de notre système. Un utilisateur quel que soit un super Admin ou un client (client EL AMED ou système externe) sont capable de gérer des modèles, des destinataires et des tâches. Le super-admin est celui qui crée des comptes pour les clients de type système externe en plus il peut faire des statistiques pour bien estimer son système et alors apporter des améliorations. Pour envoyer des SMS, le client doit s'authentifier et doit choisir un ou plusieurs destinataires. Le message à envoyer peut être basé ou non sur un modèle : s'il est basé sur un modèle alors le client est invité à choisir un modèle sinon, le texte sera saisi dans l'éditeur de l'interface d'envoi des SMS.

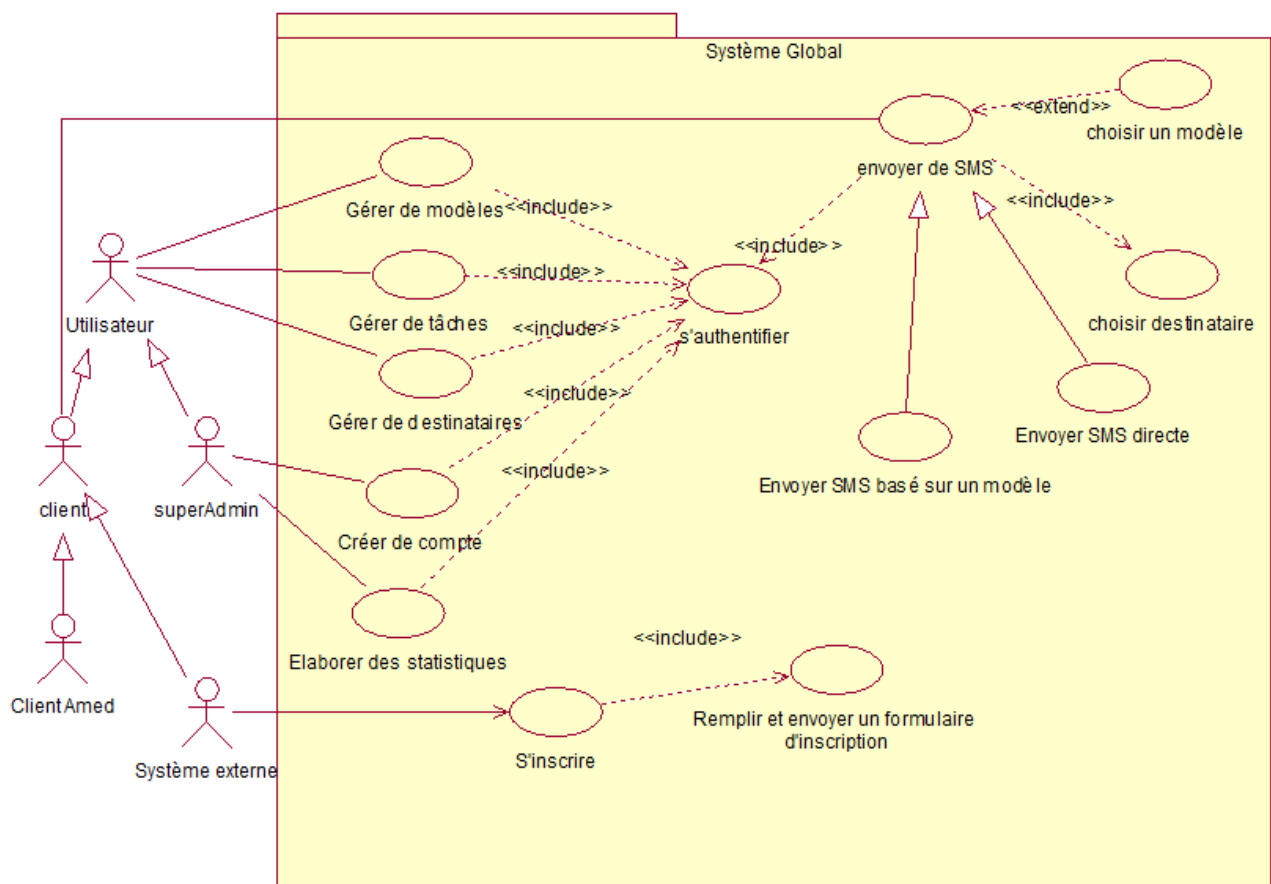


Figure 8 : Diagramme des cas d'utilisation Global

5.2. Diagramme de cas d'utilisation « Gestion des Modèles »

Un Super-administrateur ou un client peuvent gérer un modèle : ajouter, modifier, supprimer un modèle et consulter la liste des modèles. Pour faire toutes ces opérations, le super administrateur et le client doivent s'authentifier.

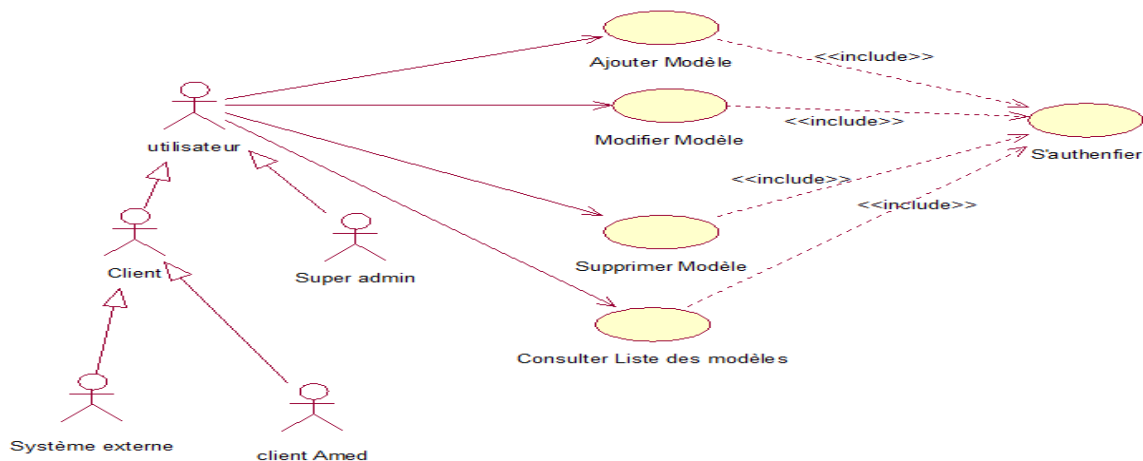


Figure 9 : Diagramme de cas d'utilisation « gestion modèle »

5.3. Diagramme de cas d'utilisation « Gestion des Destinataires »

Un super-administrateur ou un client peuvent gérer des destinataires : ajouter, modifier, supprimer un destinataire et consulter la liste des destinataires. Pour faire toutes ces opérations, le super administrateur et le client doivent s'authentifier.

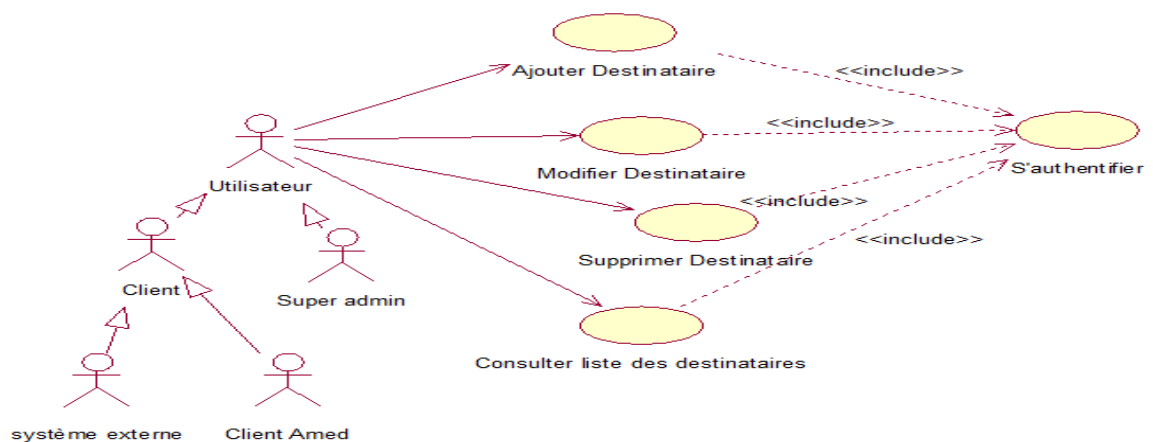


Figure 10 : Diagramme de cas d'utilisation « Gestion Destinataire »

5.4. Diagramme de cas d'utilisation « Planification des Tâches »

Un super-administrateur ou un client peuvent gérer des tâches planifiées : ajouter, modifier, supprimer une tâche et consulter la liste des tâches, en plus ils peuvent prioriser des tâches. Pour faire toutes ces opérations, le super administrateur et le client doivent s'authentifier.

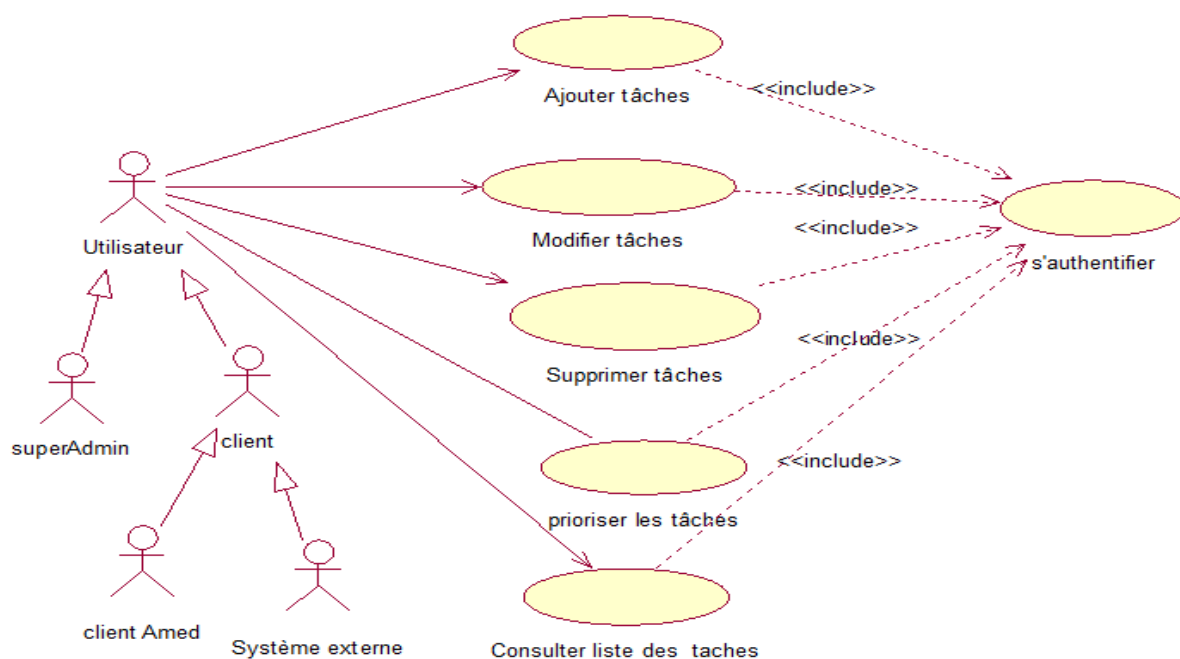


Figure 11 : Diagramme de cas d'utilisation « gestion des tâches »

5.5. Diagramme de cas d'utilisation « Envoi SMS »

Pour envoyer un sms le client ou le système externe doit choisir un ou plusieurs destinataires, si le destinataire existe déjà, il suffit de faire le choix de destinataires parmi les existants, sinon il crée un nouveau destinataire. Le message à envoyer peut être basé sur un modèle et dans ce cas l'utilisateur doit choisir un ou plusieurs modèles, sinon il édite directement le modèle dans l'éditeur. Pour envoyer un SMS, le client/le système externe doit d'abord s'authentifier.

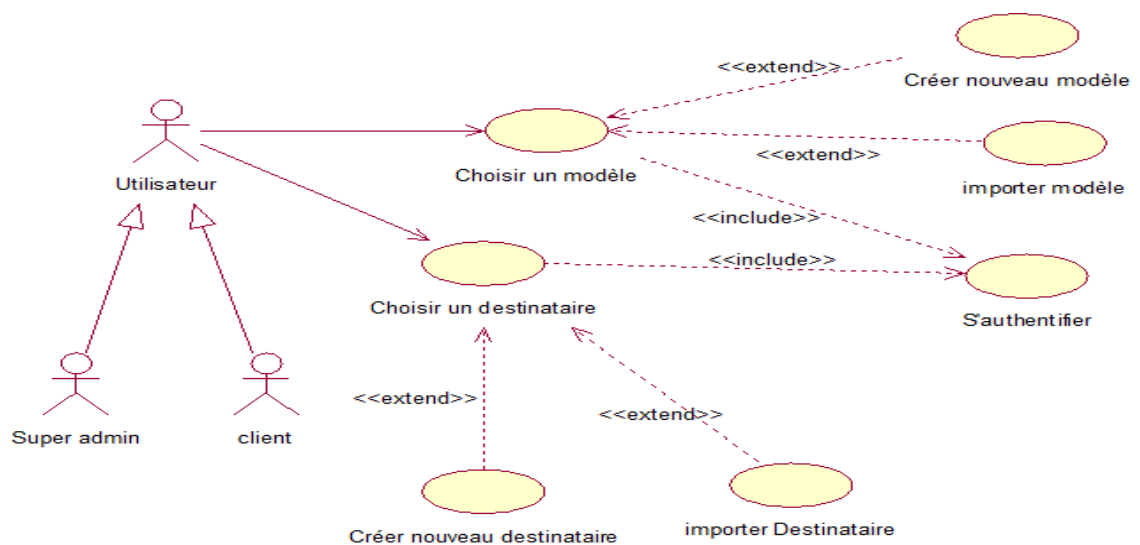


Figure 12 : Diagramme de cas d'utilisation « Envoi SMS »

6. Description textuelle des Cas d'utilisation

6.1. Scénario de cas d'utilisation « S'authentifier »

CU : S'authentifier
Objet : Saisie de login et de mot de passe
Acteurs : Utilisateur
Précondition : interface d'authentification affichée
Post-Condition : Utilisateur authentifié
Scénario nominal
Description Du Scénario Nominal « DEBUT » 1 : L'utilisateur saisit login et mot de passe. 2 : Le système vérifie les paramètres d'authentification (login et mot de passe). 3 : Le système renvoie la page d'accueil. « FIN »
Scénario alternatif

L'utilisateur saisit des paramètres erronés :

- Le système affiche un message d'erreur et reprend au point 1 du scénario nominal,
- L'utilisateur ressaisit les informations de connexion.

Tableau 1 : Scénario de cas d'utilisation « S'authentifier »

6.2. Scénario de cas d'utilisation « choisir modèle »

CU : choisir modèle
Objet : choisir un modèle
Acteurs : Client/Système externe
Précondition : interface de choix de modèles affichée
Post-Condition : Modèle choisi
Scénario nominal
Description Du Scénario Nominal « DEBUT » 1 : le client/système externe clique sur le bouton «Choisir modèle » ; 2 : Modèle choisi. « FIN »
Scénario alternatif
Le modèle souhaité n'existe pas : <ul style="list-style-type: none">• L'utilisateur clique sur le bouton nouveau modèle• Notre système renvoie l'interface de gestion des modèles• le client/système externe édite le modèle à insérer ;• Modèle ajouté avec succès
Scénario d'erreur
L'éditeur du modèle indique une erreur : <ul style="list-style-type: none">• Le système reprend au point 3 de scénario alternatif

Tableau 2 : Scénario de cas d'utilisation « choisir Modèle »

6.3. Scénario de cas d'utilisation « Ajouter Modèle »

CU : Ajouter Modèle
Objet : Permet d'ajouter un nouveau modèle
Acteurs : L'Administrateur/ le client
Précondition : interface de gestion des modèles affichée
Post-Condition : Modèle Ajouté
Scénario nominal
Description Du Scénario Nominal « DEBUT » 1 : L'Administrateur/ le client demande d'ajouter un modèle ; 2 : Le système affiche l'interface d'ajout du modèle ; 3 : L'Administrateur/ le client saisit le modèle souhaité et clique sur le bouton « ajouter Modèle » 4 : Le nouveau modèle est ajouté « FIN »
Scénario alternatif
L'utilisateur saisit des paramètres erronés : <ul style="list-style-type: none">• Le système affiche un message d'erreur et reprend au point 3 du scénario nominal,• L'Administrateur/ le client ressaisit le texte du modèle.

Tableau 3 : Scénario de cas d'utilisation « Ajouter modèle »

6.4. Scénario de cas d'utilisation « Envoyer SMS »

CU : Envoyer SMS
Objet : Envoyer SMS à un ou une liste des destinataires
Acteurs : client/Système externe
Précondition : interface d'envoi d'SMS affichée
Post-Condition : SMS envoyé

Scénario nominal
Description Du Scénario Nominal « DEBUT » 1 : Le client/ Le Système externe choisit un modèle ; 2 : Le client/ Le Système externe choisit un ou plusieurs destinataires ; 3 : Le client/ Le Système externe clique sur le bouton publier ; 4 : Le système indique que l'SMS est envoyé avec succès. « FIN »
Scénario alternatif
Le nombre de destinataires est peu nombreux : <ul style="list-style-type: none">• L'administrateur coche la liste des destinataires visés• L'administrateur édite directement le message ou choisit un modèle• L'administrateur clique sur le bouton « publier » ;• Le système indique que l'SMS est envoyé avec succès.
Scénario d'erreur
Aucune liste des destinataires n'est choisie : <ul style="list-style-type: none">• Le système reprend au point 2 du scénario nominal ;• L'administrateur clique sur le bouton publier ;• Le système indique que l'SMS est envoyé avec succès

Tableau 4 : Scénario de cas d'utilisation « Ajouter modèle »

7. Diagrammes d'activités

Dans le langage UML, un diagramme d'activité fournit une vue du comportement d'un système en décrivant la séquence d'actions d'un processus. Les diagrammes d'activité sont similaires aux organigrammes de traitement de l'information, car ils montrent les flux entre les actions dans une activité. Les diagrammes d'activité peuvent, cependant, aussi montrer les flux parallèles simultanés et les flux de remplacement. [sources.docx]

7.1. Diagramme d'activité de « l'authentification »

Le diagramme d'activité de l'authentification décrit toute la procédure avec lequel notre système se comporte. D'abord un utilisateur demande de l'authentification, lorsque l'interface est affichée, il saisit son login et mot de passe, si ses paramètres sont corrects alors il aura

navigation vers l'interface d'accueil sinon, il lui est demandé de ressaisir son login et mot de passe.

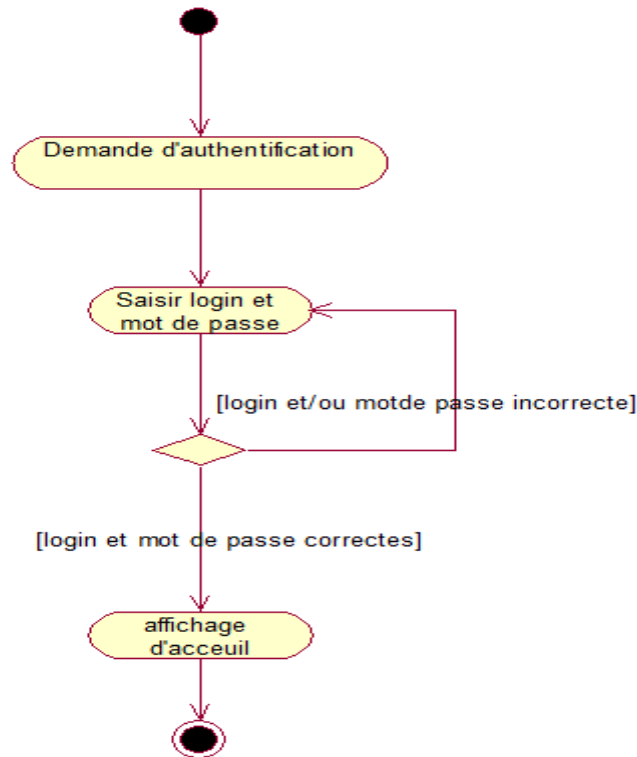


Figure 13 : Diagramme d'activité de « l'authentification »

7.2. Diagramme d'activité de « Envoi SMS»

Pour envoyer un SMS, l'utilisateur doit s'authentifier, et il demande l'interface de l'envoi de SMS, si l'envoi est basé sur un modèle, il doit choisir le modèle, si le modèle existe alors il faut choisir le modèle concerné sinon il doit l'importer à partir de la base. Le message à envoyer a bien sur son cible donc un expéditeur doit choisir au minimum un destinataire de l'SMS. En cas de l'envoi directe qui n'est pas basé sur un modèle l'utilisateur doit saisir un message dans l'éditeur de texte et toujours choisit un ou plusieurs destinataires et envoyer son SMS.

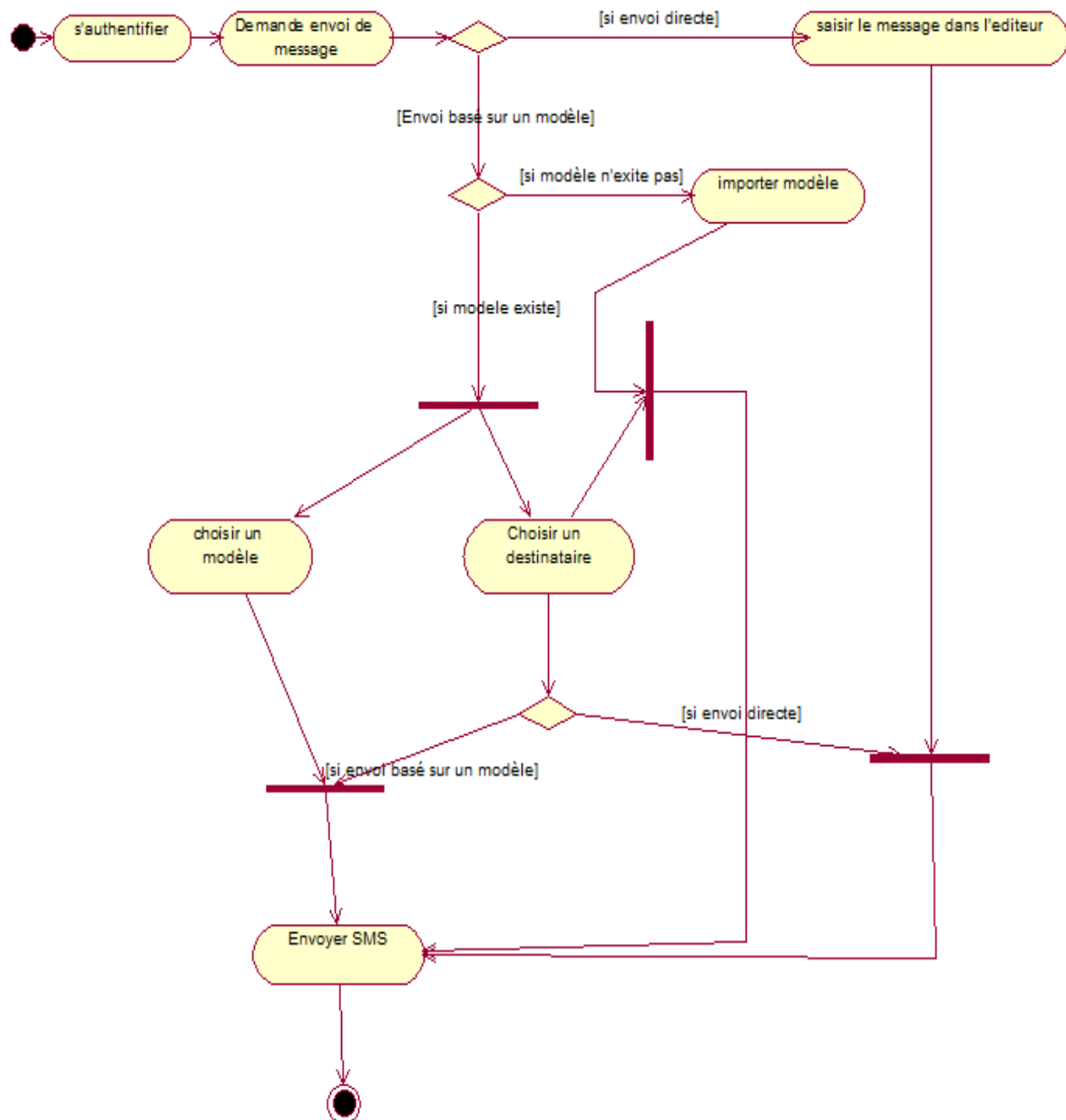


Figure 14 : Diagramme d'activité de l'envoi de SMS

8. Conclusion

Tout au long de ce chapitre, nous avons identifié les acteurs qui vont interagir avec notre application. Nous avons identifié les besoins fonctionnels et non fonctionnels et nous avons établi nos diagrammes des cas d'utilisation et les diagrammes d'activités. Dans le chapitre suivant nous présenterons la conception de l'ontologie.

SECTION II : ETUDE CONCEPTUELLE

CHAPITRE III : CONCEPTION DE L'ONTOLOGIE

1.Introduction

La première étape de notre conception consiste à créer une ontologie d'SMS en utilisant le logiciel protégé : nous créons des classes, des propriétés, des sous classes, des Domain et rang des propriétés, des cardinalités associés à ObjectProperty qui relient les différentes classes entre elles.

2.Etat de l'art ontologique

2.1. Etude de l'existant

Il existe plusieurs plateformes et logiciel qui sont construit pour envoyer des SMS, dans notre état de l'art nous abordons quelques exemple des travaux réalisés dans le même domaine que le nôtre par exemple « Octopush » qui est un service web permettant de réaliser plusieurs fonctionnalités : d'envoi des SMS, des statistique, de publipostage, sa dernière version 1.6.1 était le 10 Juin 2014.

Notre système

2.2. Ontologie et web sémantique

2.2.1. Evolution du web

Le Web (World Wide Web) a été développé par le Centre européen de recherche nucléaire (CERN), fin des années 1980. Le web supporte plusieurs améliorations en passant de web 1.0 qui concerne tout ce qui est Recherche de l'information, navigation sur le web etc.

Ensuite apparait ce qu'on appelle le web 2.0 ou web social qui fait naissance de la notion d'interaction entre le visiteur de page web avec le contenu publiée ou partagé sur internet, en fait il intervient la notion des web social comme les commentaires dans les blog , les partages et les commentaires sur Facebook, les wiki etc.

Chapitre 3 : Conception de l'ontologie

Le web 3.0 ou web sémantique est beaucoup plus récent, ayant pour but de rendre les ressources compréhensible par la machine accessible et utilisable par les programmes et agents logiciels grâce aux métadonnées d'où la notion de recherche sémantique qui a été absente auparavant avec le web 1.0 et le web 2.0. Concernant le web sémantique, la recherche est basée sur des concepts et non plus sur des mots clés. [2]

Le web 4.0 ou web intelligent, évoqué par certains comme le web intelligent, effraie autant qu'il fascine, puisqu'il vise à immerger l'individu dans un environnement (web) de plus en plus prégnant. Mais il pose par la même occasion de nombreuses questions quant à la protection de la vie privée, au contrôle des données. [7]

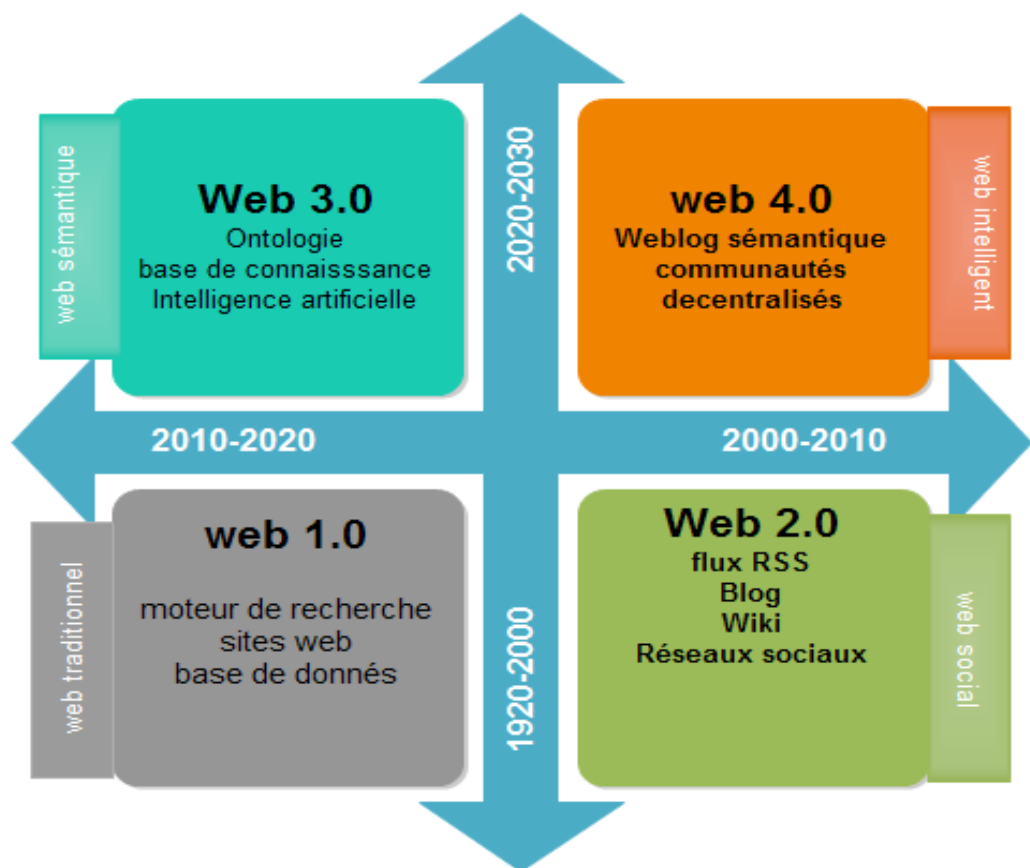


Figure 15: Les différentes versions du Web

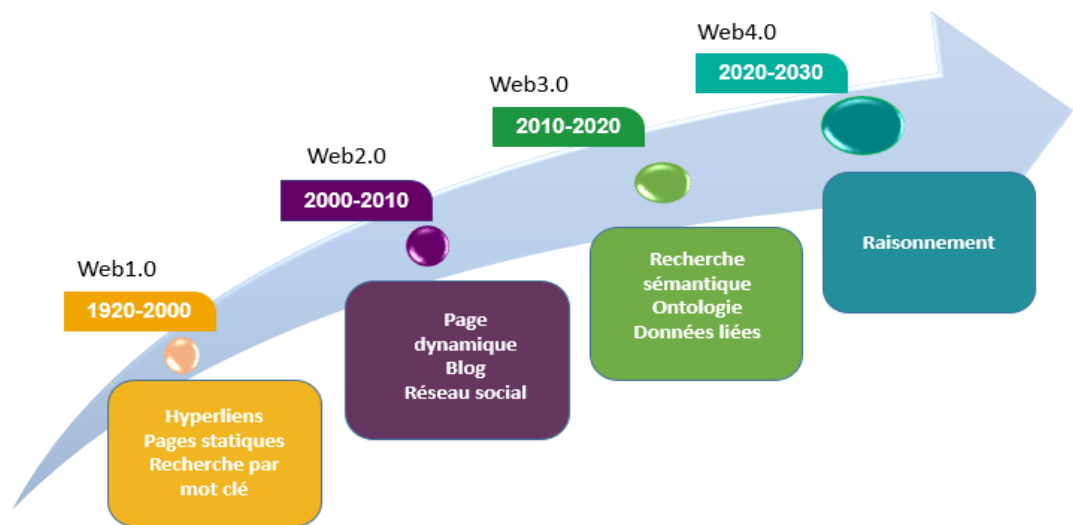


Figure 16: Evolution du web de 1.0 jusqu'à le web 4.0 [2]

2.2.2. Architecture du web sémantique

Le web sémantique est représenté sous forme de plusieurs couches : Couche d'URI et d'Unicode, couche XML, couche RDF, couche RDFS, couche OWL, couche SPARQL qui représente les règles et les requêtes, Couche Logique, Couche preuve, et enfin la couche confiance.

La couche la plus basse dans laquelle se trouve l'adressage et l'espace de nom, en fait chaque ressource existant sur le WWW doit avoir un identifiant unique pour être réutilisable et partageable à savoir l'URI (Unique Ressource Identifier) et l'IRI (Internationalized Resource Identifier).

Les normes sont utilisées pour renommer les ressources (données) dans le Web sémantique. Le web sémantique utilise aussi le XML (eXtensible Markup Language) et XML Schema car il représente des bons supports pour l'échange de donnés. XML fournit la syntaxe de base pour la sérialisation des donnés et XML Schema peut être utilisé pour ajouter des restrictions de ce syntaxe. L'IRI et l'XML sont connus comme des technologies qui permettent uniquement d'échanger les données nommés, mais qui ne sont pas utiles et suffisants pour modéliser la relation entre ces données.

Le modèle RDF ne fournit que la sémantique qui indique que certaines ressources A est en quelque sorte liée à une autre ressource B. Afin de marquer la signification des objets (ressources) dans un modèle, un vocabulaire avec des termes définis est nécessaire ; chaque

Chapitre 3 : Conception de l'ontologie

terme doit avoir un sens cohérent dans tous les contextes. RDF Schema est un vocabulaire RDF qui peut être utilisé pour définir des hiérarchies de classes et propriétés.

Les ontologies sont des vocabulaires spécialisés qui fournissent plus haut degré de la sémantique pour un domaine d'intérêt spécifique, ou un domaine. W3C a proposé OWL (Web Ontology Language), qui est une extension de RDFS destiné à être utilisé dans le contexte de WWW. D'autres technologies, telles que SPARQL pour les bases de connaissances ou d'interrogation RIF (article Inter-change Format) qui permet d'utiliser des règles supplémentaires dans le processus de raisonnement.

Exemple des données liées : par exemple, les données qui décrivent la chaise sur laquelle je suis assis vont aussi pointer vers les données de la salle dans laquelle se trouve cette chaise, les données de ma chaise sont liées ainsi à ma salle (Gandon, 2016)

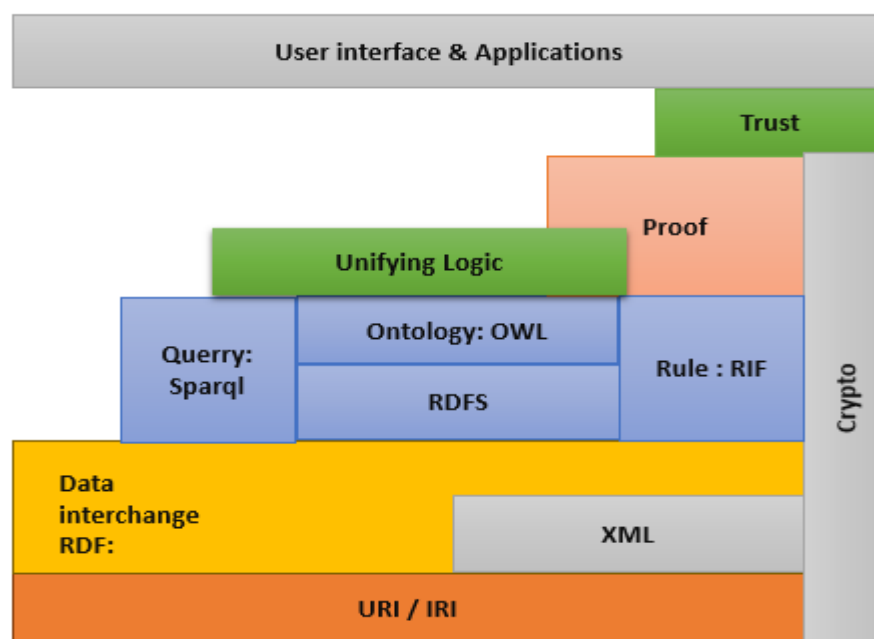


Figure 17 : Pile de standardisation [7]

2.2.3. Les Langages du web sémantique

Le Web sémantique est fondé sur les protocoles et langages standards du Web : le protocole HTTP ; les Uniform Resource identifier (URI) ; le langage XML (dans le cas, majoritaire, où RDF est sérialisé en XML, À ces standards s'ajoutent ceux qui sont propres au Web sémantique :

- **RDF : Ressource Description Framework**

RDF est le standard du Web sémantique pour la description de ressources. La syntaxe principale utilisé par RDF est RDF/XML. Le modèle RDF est un modèle de triplets (Sujet, Prédicat, Objet) et un modèle de graphes. Le modèle de triplet est donc l'atome de connaissance l'élément fondateur de modèle RDF, c'est en ce sens qu'on dit que le modèle RDF est un modèle de triplet.

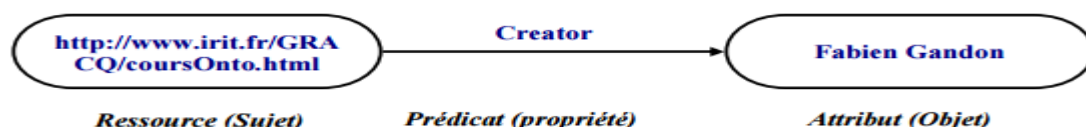


Figure 18 : Exemple de représentation graphique d'une information en RDF

Ressource	Description	Framework
Tout ce qui est peut être identifié sur le web : un téléphone, un sms, un contact etc.	On entend par description l'ensemble des attributs, des caractéristiques de ces ressources ainsi que les relations existantes entre les ressources.	Désigne le modèle RDF dans son ensemble, un langage avec une syntaxe abstraite, une sémantique et différents syntaxe concrètes.

Tableau 5 : La désignation de RDF [2]

- **RDFS : RDF Schema**

RDFS est une extension du RDF permettant de créer des ontologies simples en ajoutant la notion de classe, de sous classe, des propriétés et de sous propriétés.

- **Les ontologies**

Le langage OWL est une recommandation du W3C (World Wide Web Consortium), c'est un standard pour le Web sémantique qui fournit des primitives supplémentaires pour des ontologies plus complexes et plus évoluées. Il permet des définitions plus riches pour les classes et pour les propriétés et il permet de tirer plus de conclusions, de faire plus d'inférences (au biais du raisonneur). OWL permet de définir des classes et des propriétés grâce à des expressions logiques, la négation, la transitivité, l'équivalence, des restrictions. Par exemple, une classe

pourra être définie par l'union, la disjonction, l'intersection d'autres classes. . On l'utilise le préfixe owl : pour se référer à l'espace de nom (namespaces) du langage OWL. [2]

2.3. Définition d'une Ontologie

La communauté de l'Intelligence Artificielle et de l'Ingénierie des Connaissances ont proposé plusieurs définitions pour cerner ce qu'est une ontologie informatique et à quoi sert une ontologie. Une définition plus connu de l'ontologie est une représentation partagée et consensuelle entre les collaborateurs qui a pour but de se mettre d'accord sur un sujet particulier avec un objectif commun. Le but est de définir un ensemble de connaissances dans un domaine donné. Exploitée notamment sur le terrain du développement logiciel, elle explicite un vocabulaire en définissant les termes nécessaires pour partager la connaissance liée à ce domaine. [2]

2.4. Composantes d'une ontologie

Comme nous l'avons abordé, les ontologies fournissent un vocabulaire commun d'un domaine et définissent la signification des termes et des relations entre elles. La connaissance dans les ontologies est principalement formalisée en utilisant les cinq types de composants à savoir : concepts (ou classes), relations (ou propriétés), fonctions, axiomes (ou règles) et instances (ou individus). [12]

2.4.1. Les concepts

Les concepts, aussi appelés termes ou classe de l'ontologie, correspondent aux abstractions pertinentes d'un segment de la réalité (le domaine du problème) retenus en fonction des objectifs qu'on se donne et de l'application envisagée pour l'ontologie ;

2.4.2. Les relations

Les relations traduisent les associations (pertinentes) existant entre les concepts présents dans le segment analysé de la réalité. Ces relations incluent les associations suivantes :

- Sous classes de (généralisation-spécialisation) ;
- Partie de (agrégation ou composition) ;
- Associe à ;
- Instance de, etc.

Ces relations nous permettent d'apercevoir la structuration et l'interrelation des concepts, les uns par rapport aux autres ;

2.4.3. Les fonctions

Les fonctions constituent des cas particuliers de relations, dans laquelle un élément de la relation, (le nième) est défini en fonction des N-1 éléments précédents ;

2.4.4. Les axiomes

Les axiomes constituent des assertions, acceptées comme vraies, à propos des abstractions du domaine traduites par l'ontologie.

2.4.5. Les instances

Les instances constituant la définition extensionnelle de l'ontologie ; ces objets véhiculent les connaissances (statiques, factuelles) à propos du domaine du problème.

2.5. Les trois sous langages de l'ontologie

OWL fournit en fait trois sous-langages, d'expressivité croissante, nommés OWL Lite, OWL DL et OWL Full :

2.5.1. OWL Lite

Le langage OWL Lite peut être vu comme une extension du langage RDFS, mais auquel on aurait enlevé certaines fonctionnalités. Le principal intérêt de ce langage est de permettre la modélisation d'ontologies simples, d'une complexité formelle peu élevée, de sorte qu'il soit facile d'implémenter des raisonneurs corrects et complets.

OWL Light est destiné aux utilisateurs ayant besoin de principalement d'une hiérarchie de classe et de contraintes simples. Bien que ce langage gère les contraintes de cardinalités, il ne gère que la valeur 0 et 1 pour la cardinalité. [12]

2.5.2. OWL Full

Le langage OWL Full dispose des mêmes constructeurs qu'OWL DL mais il les interprète de manière plus large. Ainsi, une classe peut cette fois être vue comme un ensemble d'individus (définition extensionnelle) ou comme un individu à lui tout seul (définition intensionnelle) qui pourra, par exemple, donner une valeur à une propriété. À ce titre OWL Full devient clairement un sur-ensemble de RDF. Cette expressivité accrue est gagnée au détriment de la complexité : le langage OWL Full n'est plus décidable. Ce sous langage permet une expressivité maximale et une liberté syntaxique mais le raisonnement peut être incomplet. [12]

2.5.3. OWL DL

Ce troisième sous langage se base surtout sur la logique de description et les règles relatives à cette logique. Il est toujours avec un raisonnement complet. Le langage OWL DL contient des constructeurs supplémentaires, mais il ne peut être utilisé qu'avec certaines restrictions. Par exemple, une classe ne peut pas être une instance d'une autre classe. Il en résulte un langage un peu plus expressif mais toujours décidable, c'est à dire que les conséquences sont toujours calculables en un temps fini. [12]

	OWL Lite	OWL DL	OWL FULL
Compatibilité avec RDF	Théoriquement, aucun document de RDF ne peut être considérée comme compatible avec OWL Lite	Théoriquement, aucun document de RDF ne peut être considérée comme compatible avec OWL DL	Tous les documents de RDF valides sont OWL FULL
Restrictions sur la définition de la classe	Nécessite des classes, des instances, des propriétés	Nécessite la séparation des classes, des instances, des propriétés et des valeurs de données	peut y avoir des cas où les propriétés en même temps
Description des classes	La seule description disponible pour OWL Lite est « Intersection Of »	Plusieurs descriptions des classes sont possibles comme « UnionOf », « ComplementOf », « Intersection Of » et l'énumération	Plusieurs descriptions des classes sont possibles comme « UnionOf », « ComplementOf », « Intersection Of » et l'énumération
Contraintes de cardinalité	« Cardinality » : 0,1 « MinCardinality » : 0,1 « MaxCardinality » : 0,1	« Cardinality » : >=0 « MinCardinality » : >=0 « MaxCardinality » : >=0	« Cardinality » : >=0 « MinCardinality » : >=0 « MaxCardinality » : >=0

Tableau 6 : Tableau comparatif OWL DL, Lite et Full [12]

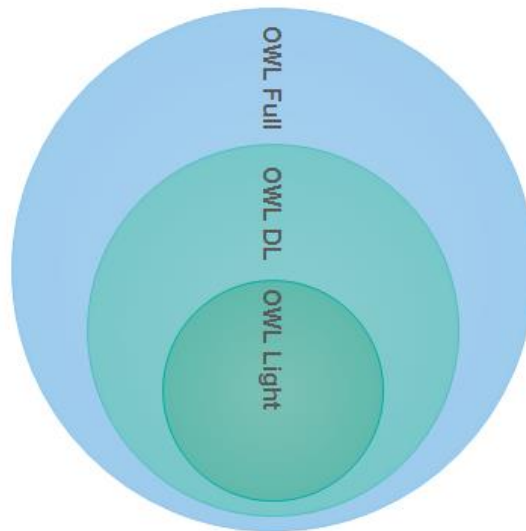


Figure 19 : Les trois sous langages d'OWL

Notre système est évidemment basé sur l'un de ces sous-langages qui est l'OWL DL. Notre choix est argumenté par notre vue à un futur système qui est basé sur un raisonneur qui représente une aide à la décision, En effet lui fournissant des informations (relations) règles que nous lui apprenons, il raisonne selon ces règles pour en déduire d'autres informations. Avec OWL lite ce n'est pas possible de raisonner, avec OWL Full le raisonnement est possible mais c'est plus compliqué, d'où nous considérons OWL DL comme le sous langage le plus adapté à nos futur besoins

2.6. Types d'ontologies

Il existe plusieurs types d'ontologie, les chercheurs dans ce domaine ont classifiées les ontologies en fonction des objets qui modélisent les ontologies pour répondre à un objectif précis. Dans notre cas nous allons considérer la classification d'A. Gómez-Pérez et N. Guarino qui sont : l'ontologie de haut niveau, l'ontologie d'application, ontologie des tâches et l'ontologie de l'application.

2.6.1. Ontologie de domaine

L'ontologie de domaine est orienté objet utilisé pour illustrer un domaine bien déterminé sous forme d'une base de connaissance. Elle renferme les concepts clés, les attributs et les instances en relation avec ce domaine. Les ontologies vont permettre de spécifier les connaissances d'un domaine, de façon aussi indépendante que possible du type de

manipulations qui vont être opérées sur ces connaissances. Ces ontologies sont appelées « ontologies de domaine », puisqu'elles sont construites sur un domaine particulier de la connaissance. De nombreuses ontologies de domaine existent déjà, telles que MENELAS dans le domaine médical (Zweigenbaum, 1999).

2.6.2. Ontologie de haut niveau

Ces sont des ontologies qui sont indépendants d'un domaine particulier, elle touche les concepts généraux comme le temps, les événements, etc. Une distinction est établie entre les ontologies de domaine portant sur des concepts renvoyant à des objets matériels ou à des concepts d'assez bas niveau (c'est-à-dire n'offrant que des possibilités limitées de raffinement) et les ontologies portant sur des concepts de haut niveau. Ces dernières décrivent des notions générales comme les notions d'objet, de propriété, d'état, de valeur, de moment, d'événement, d'action, de cause et d'effet (Sowa, 2000). En théorie, les ontologies de haut niveau doivent pouvoir être reliées au sommet des ontologies de domaine.

2.6.3. Ontologie de tâche

Cette ontologie permet d'illustrer un corpus de connaissances sous forme de compréhensible et utilisable par la machine. Elle représente un ensemble des concepts qui sont bien structurés et organisés dans un graphe dont la relation entre ces concepts peuvent être sémantiques ou d'héritage. L'ontologie de tâche décrit les connaissances portant sur des tâches et/ou des activités particulières (faire un diagnostic, planifier une activité . . .). Ces ontologies fournissent un ensemble de termes au moyen desquels on peut décrire, au niveau générique, comment résoudre un type de problème. [2]

2.6.4. Ontologie d'application

Cette ontologie est très spécifique concerne un domaine restreint pour exécuter une tâche. Ce sont les ontologies les plus spécifiques, elles contiennent les connaissances requises pour une application particulière. Les ontologies d'application étendent et spécialisent les connaissances contenues dans l'ontologie de domaine et dans l'ontologie de tâche pour une application donnée. Selon A. Maedche et S. Staab (2001), les concepts dans l'ontologie d'application correspondent souvent aux rôles joués par les objets du domaine tout en exécutant une certaine activité, par exemple : hypothèse, signe, diagnostic. [2]

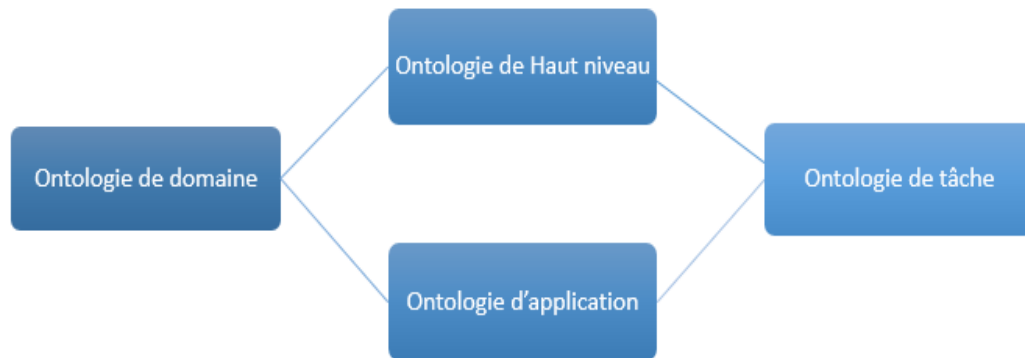


Figure 20: Classification des ontologies selon Guarino [2]

2.6.5. Le type de notre ontologie

Il est évident qu'il est difficile de faire la différence, pour une ontologie considérée, entre ces différentes classifications et de choisir celle qui correspond. Ainsi, l'ontologie de l'SMS réalisée dans le cadre de notre système de l'ontologie de domaine.

2.7. Cycle de vie d'une ontologie

Dans le cycle de vie d'une ontologie il existe des étapes liées à la construction de l'ontologie et des étapes liées à l'évolution ou la maintenance de l'ontologie. Afin de situer l'évolution. En effet, les ontologies sont destinées à être utilisées comme des composants logiciels dans des systèmes répondant à des objectifs opérationnels différents, leur développement doit s'appuyer sur les mêmes principes que ceux appliqués en génie logiciel. Ainsi, les ontologies doivent être considérées comme des objets techniques évolutifs et possédant un cycle de vie qui nécessite d'être précisé. Dans ce contexte, les activités liées aux ontologies sont, d'une part, des activités de gestion de projet (planification, contrôle, assurance qualité), et, d'autre part, des activités de développement (spécification, conceptualisation, formalisation) ; s'y ajoutent des activités transversales de support telles que l'évaluation. [2]

Nous avons adapté le cycle de vie à nos besoins et proposons notre vision du cycle de vie d'une ontologie. Il comprend une étape initiale de détection et de spécification des besoins qui permet notamment de circonscrire précisément le domaine de connaissances, une étape de

conception qui se subdivise en trois phases, une étape de déploiement et de diffusion, une étape d'utilisation, une étape, incontournable, d'évaluation, et enfin, une sixième étape consacrée à l'évolution et à la maintenance du modèle. [2]

Après chaque utilisation significative, l'ontologie et les besoins doivent être réévalués et l'ontologie peut être étendue et, si nécessaire, en partie reconstruite. La validation du modèle de connaissances est au centre du processus et se fait de manière itérative.

L'ontologie dès sa naissance passe par six étapes : d'abord des besoins qui représentent l'initiative de création de l'ontologie, après on fait sa conception (concepts, classes, propriétés, relations entre les classes, relation entre les propriétés), la phase de diffusion, la phase de l'utilisation de l'ontologie, la phase d'évaluation et enfin la phase de évolution bien sûr s'il est possible de l'améliorer La figure 20 représente le cycle de vie de l'ontologie :

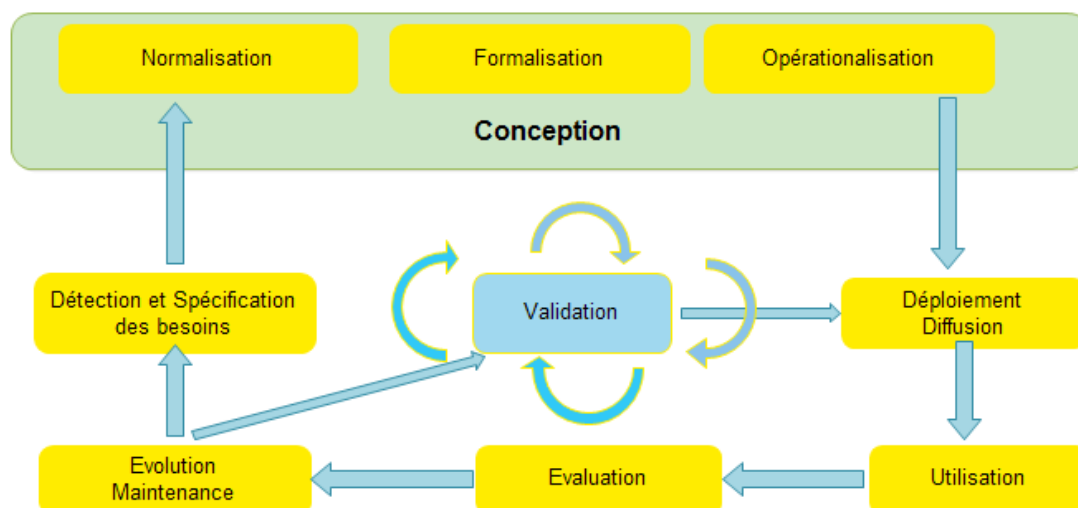


Figure 21 : Cycle de vie de l'ontologie [2]

2.8. Etat de l'art des méthodes de conception des ontologies

Il existe une multitude de méthodes d'ingénierie ontologique. On entend par méthodologie, les procédures de travail, les étapes, qui décrivent le pourquoi et le comment de la conceptualisation puis de l'artefact construit. Dans la suite nous allons présenter que deux méthodes de l'état d'ontologies :

2.8.1. La méthode de Bachimont

Cette méthode propose de contraindre l'utilisateur à un engagement sémantique en introduisant une normalisation sémantique des termes manipulés dans l'ontologie. La méthode de normalisation suit trois étapes.

- **Normalisation sémantique** : l'utilisateur doit choisir les termes du domaine et les normaliser en explicitant leurs propriétés et en exprimant les identités et les différences dans leur voisinage proche. La place d'une notion dans l'ontologie doit être justifiée par rapport à la communauté et la différence avec le père et la fratrie.
- **Formalisation des connaissances** : Cette étape consiste à désambiguïser les notions de l'ontologie référentielle obtenue par l'étape précédente et choisir leurs sens pour un domaine spécifique. Cela peut nécessiter la création de nouveaux concepts, l'ajout de propriétés et d'axiomes.
- **Opérationnalisation des connaissances** : Le système utilise un langage opérationnel de représentation de connaissances qui possède les caractéristiques nécessaires pour répondre aux besoins exprimés lors de la spécification du système.

2.8.2. La méthode METHONTOLOGY

- C'est une méthodologie mise au point par l'équipe du laboratoire de l'intelligence artificielle de l'Université polytechnique de Madrid. Cette méthode inclut :
 - L'identification du processus de développement de l'ontologie.
 - Le cycle de vie basé sur l'évolution de prototypes.
 - Les techniques de gestion de projet (planification, assurance qualité), de développement (spécification, conceptualisation, formalisation, implémentation, maintenance) et des activités de support (intégration, évaluation, documentation).

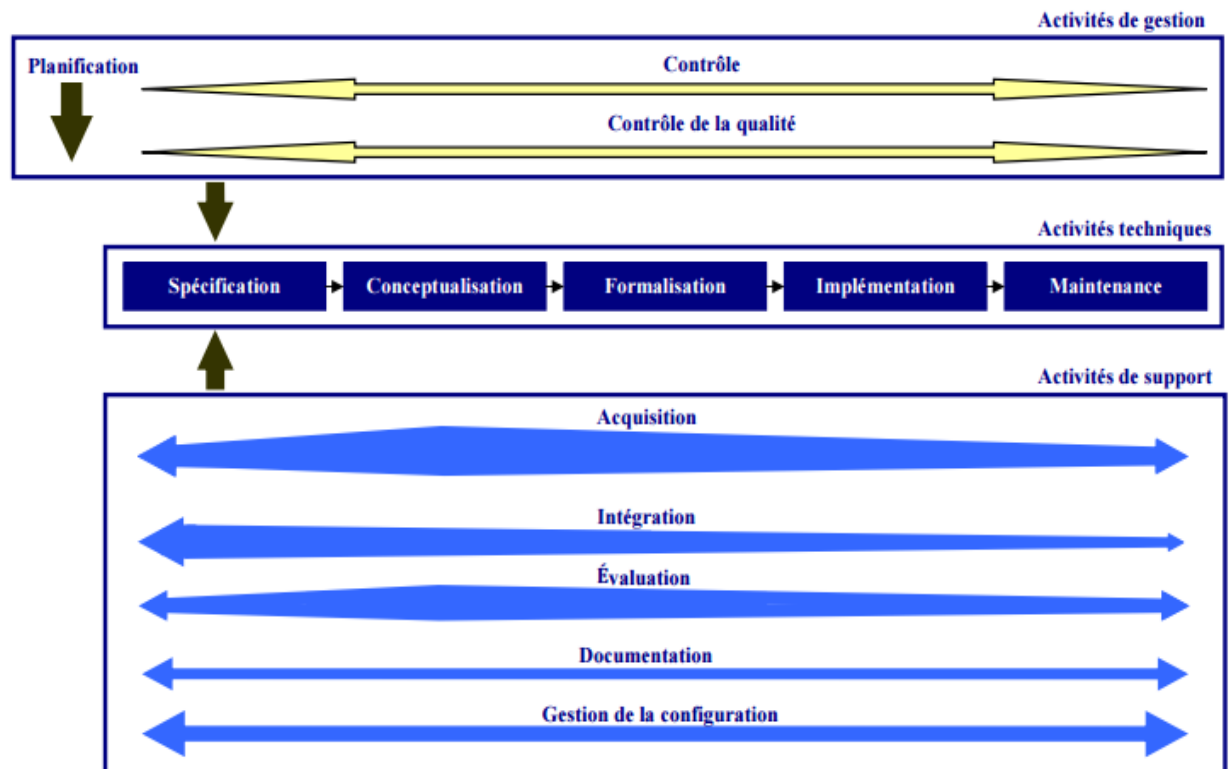


Figure : Cycle de vie d'une ontologie selon METHONTOLOGY

2.8.3. La méthode utilisée

La méthode METHONYOLGY en tenant compte de ses trois types d'activité (activités de support, activités technique et activités de gestion) nous semble la plus adapté à notre système.

2.9. Un exemple de domaine d'application de l'ontologie

2.9.1. Domaine de médecine

Le nombre des connaissances disponibles dans les domaines de la médecine, de la biologie et de la santé publique s'accroît. En médecine notamment, plusieurs raisons y concourent : L'augmentation du nombre de paramètres nécessaires à la prise en charge des patients pose le problème de la maîtrise de l'information. Face à l'augmentation des connaissances médicales et des paramètres de soins à prendre en compte, il apparaît nécessaire de recourir aux méthodes de traitement de l'information et à l'informatique. [2]

2.9.2. Le raisonneur

2.9.2.1. Définition

Un raisonneur est un programme informatique capable de comprendre les ontologies et de réaliser les inférences. En d'autres termes, il interprète les relations entre les concepts de l'ontologie de manière à créer de nouvelles relations valides. Si l'ontologie permet de représenter formellement les connaissances, c'est véritablement le raisonneur qui est capable d'en tirer le maximum. [6]

SWRL (Semantic Web Rule Language) est un langage basé sur RuleML qui permet de définir des règles sous la forme d'implications logiques entre des conditions et des conclusions.

2.9.2.2. Comparaison des différents types

Il existe une multitude de raisonneurs qui ont chacun leurs propres spécificités notamment en termes de compatibilité, d'expressivité et de performance. Parmi les plus connus, nous retrouvons Pellet, HermiT, Fact++ et RACER. Le choix d'un raisonneur doit se faire de manière réfléchie en fonction des fonctionnalités attendues. Dans le tableau ci-dessous nous considérons uniquement trois types de raisonneur. [6]

	Pellet	Racer	Fact++
Disponibilité	Open source	Commercial	Open Source
Profile	DL, EL	DL	DL
Support de règles	Oui (SWRL)	Oui (SWRL)	Non
Support Jena	Oui	Oui	Oui
Support protégé	Oui	Oui	Oui

Tableau 7: Tableau comparatif des quelques raisonneurs [6]

2.9.2.3. Exemple de code SWRL

```
<swrl:Variable rdf:about="urn:swrl#y"/>
<swrl:Variable rdf:about="urn:swrl#z"/>
<swrl:Variable rdf:about="urn:swrl#x"/>
<swrl:Imp>
  <swrl:body>
    <swrl:AtomList>
      <rdf:rest>
        <swrl:AtomList>
          <rdf:rest rdf:resource="#&rdf:nil"/>
          <rdf:first>
            <swrl:IndividualPropertyAtom>
              <swrl:propertyPredicate rdf:resource="#&Ontology;aPourPere"/>
              <swrl:argument1 rdf:resource="urn:swrl#x"/>
              <swrl:argument2 rdf:resource="urn:swrl#y"/>
            </swrl:IndividualPropertyAtom>
          </rdf:first>
        </swrl:AtomList>
      </rdf:rest>
      <rdf:first>
        <swrl:IndividualPropertyAtom>
          <swrl:propertyPredicate rdf:resource="#&Ontology;aPourFrere"/>
          <swrl:argument1 rdf:resource="urn:swrl#y"/>
          <swrl:argument2 rdf:resource="urn:swrl#z"/>
        </swrl:IndividualPropertyAtom>
      </rdf:first>
    </swrl:AtomList>
  </swrl:body>
  <swrl:head>
    <swrl:AtomList>
      <rdf:rest rdf:resource="#&rdf:nil"/>
      <rdf:first>
        <swrl:IndividualPropertyAtom>
          <swrl:propertyPredicate rdf:resource="#&Ontology;aPourOncle"/>
          <swrl:argument1 rdf:resource="urn:swrl#x"/>
          <swrl:argument2 rdf:resource="urn:swrl#z"/>
        </swrl:IndividualPropertyAtom>
      </rdf:first>
    </swrl:AtomList>
  </swrl:head>
</swrl:Imp>
```

Figure 22 : Exemple de code SWRL en XML [6]

3. Conception de notre ontologie

Au cours de l'étape de création d'une ontologie pour notre nouveau système, un des préceptes à respecter est celui de vérifier son existence et de la réutiliser. L'idée est d'investir le temps à créer de nouvelles métadonnées sur ce qui n'a pas été décrit auparavant que de perdre le temps à réinventer la roue. Notre ontologie visée sollicite les ontologies existantes suivantes :

- Ontologie vCard
- Ontologie SMS
- Ontologie Nepomuk
- Ontologie Time
-

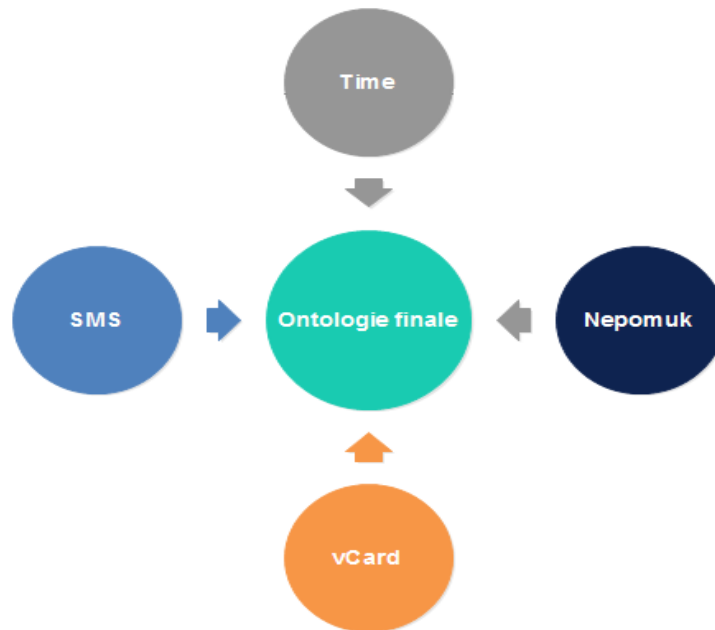


Figure 23 : Les Ontologies réutilisés

3.1. L'ontologie vCard (Visit Card)

VCard est créé par l'IETF (Internet Engineering Task Force) est conçue principalement pour décrire des personnes et des organisations et leurs adresses. Cette ontologie est utilisée par le carnet d'adresse d'Apple, ainsi que les logiciels de messagerie électronique tel que Mozilla Thunderbird et Outlook.

3.2. L'ontologie Nepomuk

L'ontologie Nepomuk est une ontologie de domaine du message, elle est standardisé et est créé pour un but de description des messages (emails, message instantané, etc.). Comme l'sms est considéré comme un type de message, nous avons besoin de réutiliser les classes, les propriétés de cette ontologie car il s'agit approximativement du même principe.

3.3. L'ontologie time

L'ontologie Time est aussi une ontologie standardisée par le W3C (World Wide Web Consortium). le but principale de l'utilisation de cette ontologie dans notre propre ontologie a été la manipulation des tâches planifiées.

3.4. Ontologie SMS

L'ontologie SMS est une ontologie qui décrit les différentes classes, propriétés qui sera utile pour notre ontologie. On utilise quelques classes et quelques propriétés.

3.5. Ontologie finale

C'est l'ontologie résultante de différentes ontologies cités dans les paragraphes ci-dessus. Il s'agit d'adapter les classes et les propriétés de l'ontologie existante en fonction de nos besoins. La figure ci-dessous représente l'importation sous protégé de l'ontologie time : les namespaces des ontologies TimeZone, Nepomuk, etc.

Default Namespace	
http://www.owl-ontologies.com/SMSOntology.owl#	
Namespace Prefixes	
Prefix	Namespace
Time	http://www.w3.org/2006/time#
vcard	https://www.w3.org/TR/vcard-rdf#
xsd	http://www.w3.org/2001/XMLSchema#
sms	http://www.semanticweb.org/ontologies/SMS.owl#
protege	http://protege.stanford.edu/plugins/owl/protege#
rdfs	http://www.w3.org/2000/01/rdf-schema#
Nepomuk	http://www.semanticdesktop.org/ontologies/2007/03/22/nmo/v1.1/#
rdf	http://www.w3.org/1999/02/22-rdf-syntax-ns#
owl	http://www.w3.org/2002/07/owl#

Figure 24: Les ontologies importées sous protégé

3.6. Notre modèle ontologique

Le schéma ci-dessous représente les classes et les sous classes de notre modèle ontologiques. Toutes les classes sont des sous classes de la classe Thing. Cette relation de hiérarchisation est traduite par le terme « has Subclass ».

3.6.1. Les propriétés d'objets

Dans une ontologie, les concepts (classes) sont reliés entre elles d'une façon hiérarchique, les autres relations correspondent à des relations de type ObjectProperty définie par le logiciel protégé. Le tableau ci-dessous montre les propriétés de type objet.

Propriété	Désignation	Domain	Range
personnalEmail	désigne l'email personnel d'un contact	contact	Email
References		Message	Message
inReplyTo	Indique s'il s'agit d'une réponse sur un message, c'est une sous propriété de la propriété « References »	Message	Message
HasDateTime	Indique la date et le temps d'une tâche planifiée	ScheduledTask	TemporalEntity
Recipient	Indique que le récepteur d'un message est un contact	Message	Contact
PrimaryRecipient	Le récepteur primaire d'un message, c'est une sous propriété de « Recipient »	Message	Contact
To	sous propriété de « PrimaryRecipient »	Message	contact
SecondaryRecipient	Le récepteur secondaire sous propriété de Recipient	Message	contact
Cc	Sous propriété de la propriété « SecondaryRecipient »	Email	contact
Bcc	Sous propriété de la propriété « SecondaryRecipient »	Email	contact
IsTransmittedThrough	Indique l'intermédiaire entre l'émetteur et le récepteur	SMS	Provider
HasAttachment	Indique que l'email contient des pièces jointes	Email	Attachment
inDateTime	Indique les entités de temps (jour, heure, minute,...)	Instant	DateTimeDescription

ReplyTo	Indique	Message	contact
From	Un message est envoyé par un utilisateur	Message	user
email	Un utilisateur envoie email à un contact	VCard	Contact
dayOfWeek	Liste les jours d'une semaine	DateTimeDescription	DayOfWeek
bdy	Un contact ayant une date de naissance	Contact	Date
Has	Un client possède un compte	Customer	Account
Has	Un modèle possède une ressource (entête et corps de message)	Model	Ressource
mobileTel	Un contact ayant un numéro de téléphone	Contact	cellPhoneNumber
IsBasedOn	Le message est basé sur un modèle	Message	Model
workTel	Le téléphone de l'entreprise	workTel	MessagingNumber
Send	L'utilisateur qui va envoyer le message	User	Message
HasPrice	Indique le prix d'un SMS	Sms	Price
MessageHeader	Indique l'entête du message	Message	Message Header
Workemail	désigne l'email d'une entreprise	Organization	EmailAddress
Manage	Le super-administrateur gère des comptes de clients	Super-administrator	Account
Manage	Un utilisateur gère un modèle	User	Model
Manage	Un utilisateur gère des tâches planifiées	User	ScheduledTask
IsComposedBy	Un message peut être un texte, un audio ou multimédia	Message	mediaContent

Tableau 8 : Propriétés d'objets

3.6.2. Les propriétés de donnés

En plus des ObjectProperty, on cite le DataTypeProperty qui permet de relier une classe à une valeur (chaîne de caractères, entier...). Le tableau ci-dessous représente les propriétés de donnés utilisés dans notre ontologie.

Propriété	Désignation	Domain	Range
Year	Indique l'année d'envoi de message	User	Date
SentDate	Indique la date d'envoi de message	Message	dateTime
ReceivedDate	Indique la date de réception d'un message	Email	dateTime
xsdDateTime		DateTimeDescription	DateTime
Organization-name	Indique le nom d'une entreprise	Organization	String
Hour	Indique l'heure d'envoi de message	DateTimeDescription	Int
Month	Indique le mois d'envoi de message	dateTimeDescription	Int
Minute	Indique la minute d'envoi de message	dateTimeDescription	Int
Second	Indique la seconde d'envoi de message	DateTimeDescription	Int
day	Indique le jour d'envoi de message	DateTimeDescription	Int
DayOfYear	Indique un jour de l'année	DateTimeDescription	String
messageSubject	Indique l'objet d'un email	Message	String
HasMaxLength	Indique la longueur maximale du message	Message	Int
Nickname	Indique le nom du contact	Contact	String
Familyname	Indique le nom de famille d'un contact	Contact	String
hasSold	Indique le solde d'un utilisateur	User	float

Tableau 9 : Propriétés de donné

4. Diagramme de classes de l'ontologie

De nombreux concepts de OWL sont très semblables à ceux des diagrammes de classes UML par exemple les classes de l'ontologie sont similaires aux classes de UML, les propriétés de données aux attributs des classes et les propriétés d'objet aux associations en UML, les individus aux objets etc. Donc il doit y avoir eu des tentatives pour définir un profil UML pour OWL. Plusieurs outils sont utilisés par exemple OWLGrEd est un outil qui permet de se concentrer sur la visualisation des ontologies OWL dans le style d'UML.

Ontologie	UML
Classes de l'ontologie	Classes en UML
Propriétés des données	Attributs des classes
Propriétés de données	Association entre les classes
Individus	objets
restriction	cardinalité

Tableau 10 : Identification ontologie-UML

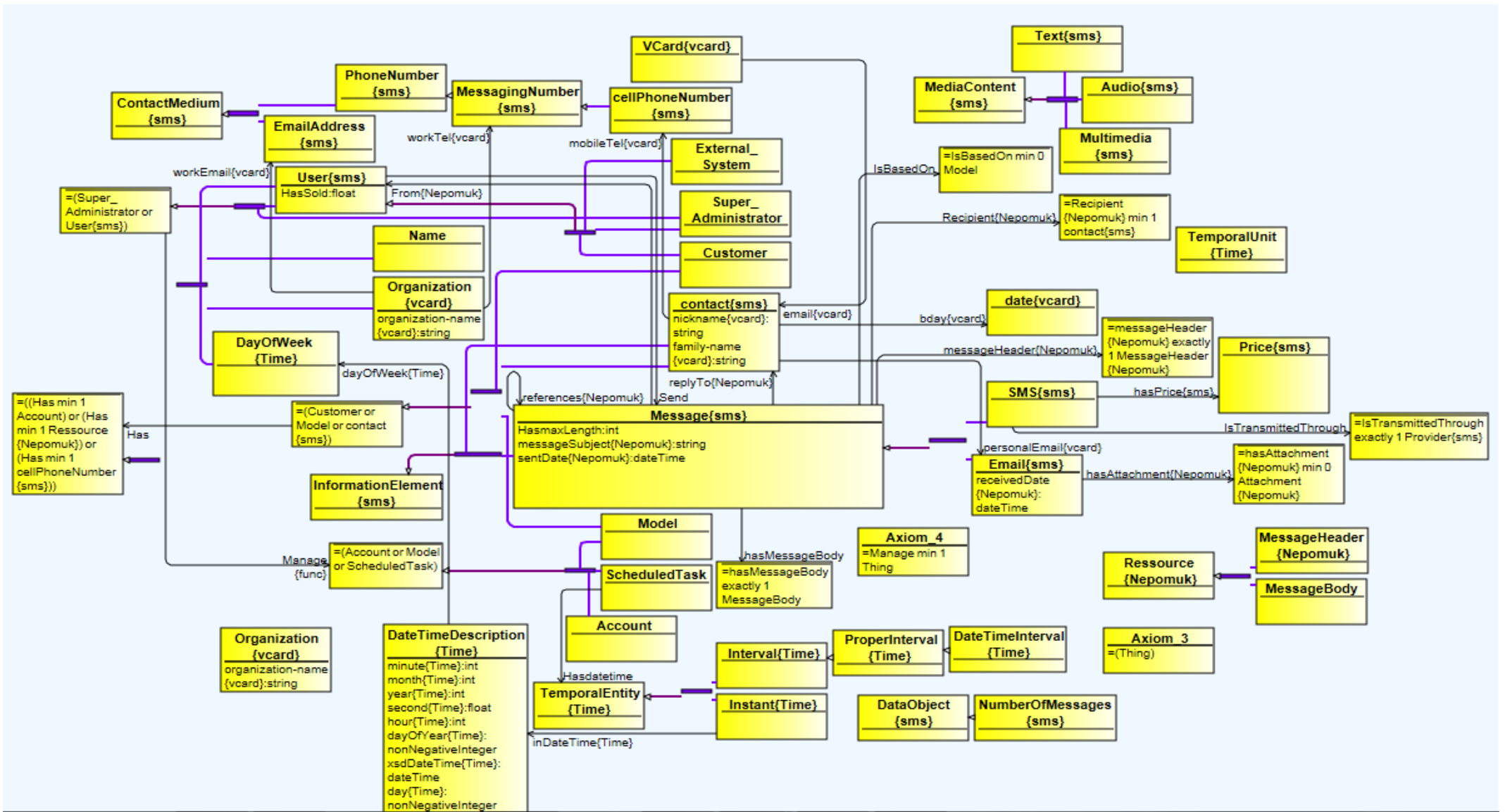


Figure 26 : Graphe de l'ontologie généré avec OWLGrE

5. Interprétation du diagramme de classes de l'ontologie

Le digramme ci-dessus est le résultat de l'exportation de notre ontologie finale à partir de protégé, il représente les différentes classes de notre ontologie SMS, Parmi ces classes il existe ceux qui appartiennent aux ontologies réutilisés et autres qui sont propres à notre ontologie. La relation entre les différents concepts doit se lire en triplets c'est à dire Sujet-Prédicat-Objet par exemple : Un message (sujet) « HasMessageBody » (prédicat = propriété) MessageBody (Objet). Le prédicat est soit un attribut d'une classe (propriété de donné) ou bien une association entre deux classes (propriété d'objet). OWL a plus fonctionnalités que les diagrammes de classes UML comme expressions de classes, classes anonymes, etc. Par exemple dans le digramme ci-dessus contient des propriétés qu'on appelle propriétés fonctionnels lorsque un sujet ayant plusieurs objet en utilisant la même propriété. Comme exemple la propriété « Manage » du sujet « super- Admin » : super- Admin Manage Account, super-Admin Manage ScheduledTask, super- Admin Manage Model.

6. Conclusion

Dans ce chapitre nous avons réalisé l'étude conceptuelle de notre ontologie. Cette étude étant subdivisée sous deux parties. La première partie parle de l'ontologie et le web sémantique de façon générale et la deuxième partie est consacrée pour un état de l'art ontologique qui présente les différentes ontologies réutilisés pour construire notre propre ontologie de SMS.

CHAPITRE IV : CONCEPTION & INTERFAÇAGE DU FUTUR SYSTÈME

1. Introduction

La phase de conception est une phase fondamentale dans le cycle de vie de tout projet. C'est la phase qui précède la phase d'implémentation, autrement, elle représente la dernière approche théorique du projet.

2. Etude dynamique du système global

2.1. Diagramme de séquence

Les diagrammes de séquences sont la représentation graphique des interactions entre les participants et le système selon un ordre chronologique dans la formulation Unified Modeling Language. Son utilité montrer les interactions d'objets dans le cadre d'un scénario d'un Diagramme des cas d'utilisation.

2.1.1. Diagramme de séquence de cas d'utilisation « Ajout Modèle »

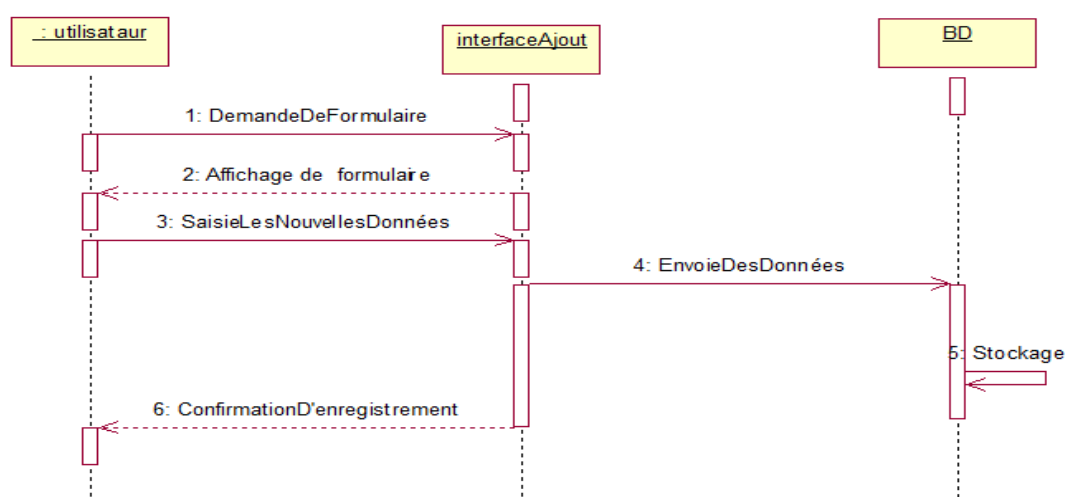


Figure 27 : Diagramme de séquence de cas utilisation d'ajout de modèle

2.1.2. Diagramme de séquence de cas d'utilisation « s'authentifier »

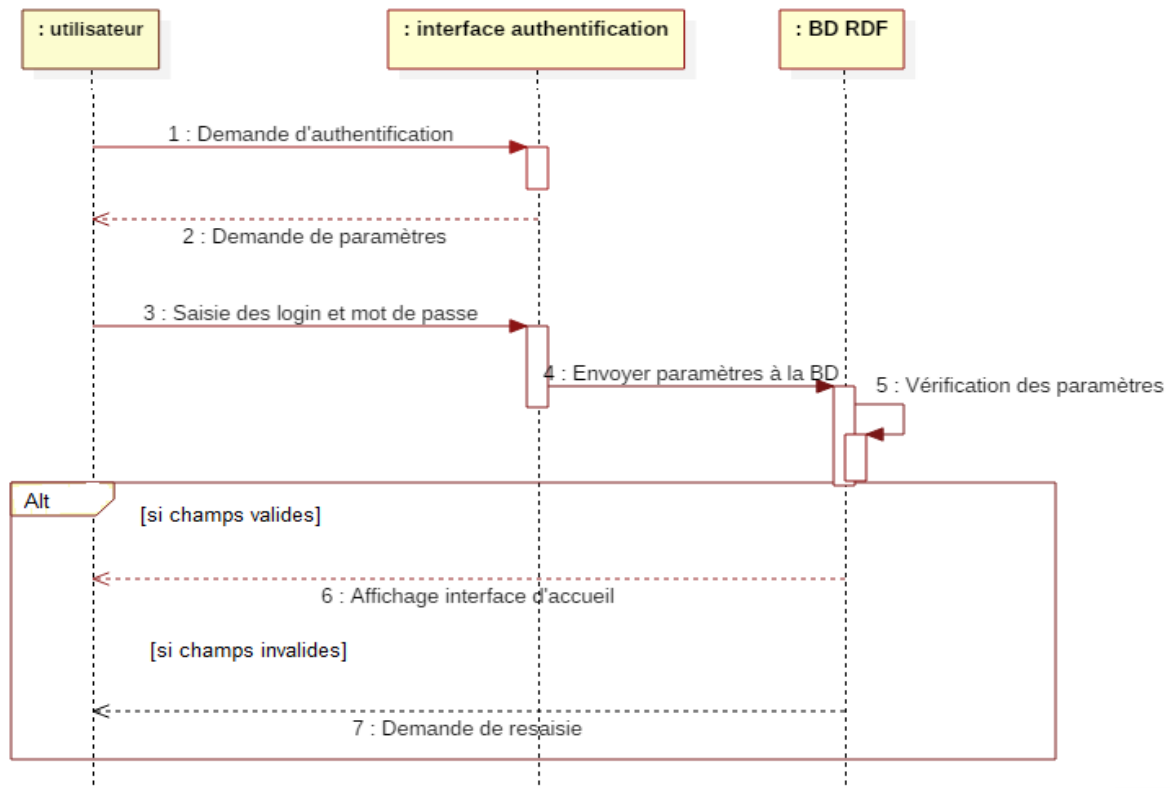


Figure 28: Diagramme de séquence de cas d'utilisation « s'authentifier »

2.1.3. Diagramme de séquence de cas d'utilisation « Modifier Modèle »

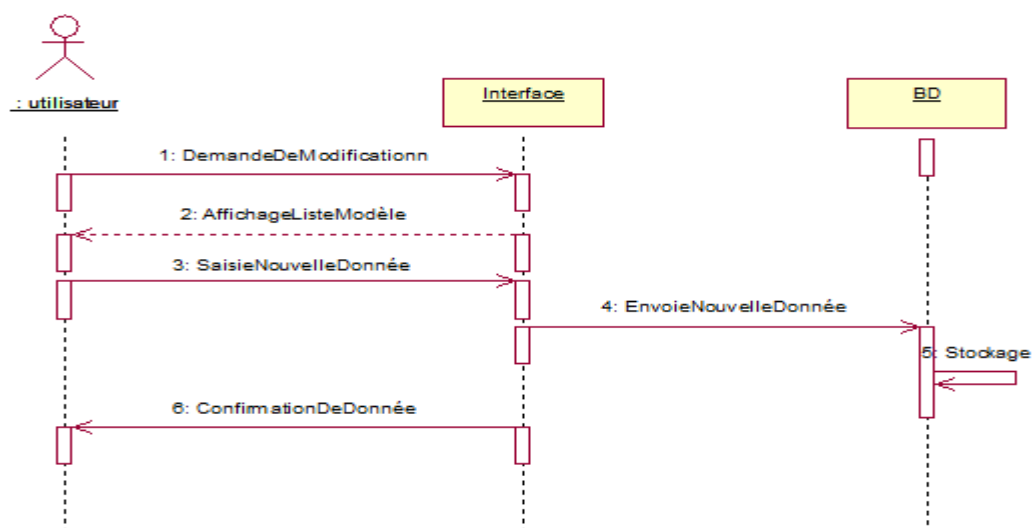


Figure 29: Diagramme de séquence de cas d'utilisation « modification Modèle »

2.1.4. Diagramme de séquence de cas d'utilisation « Supprimer Modèle »

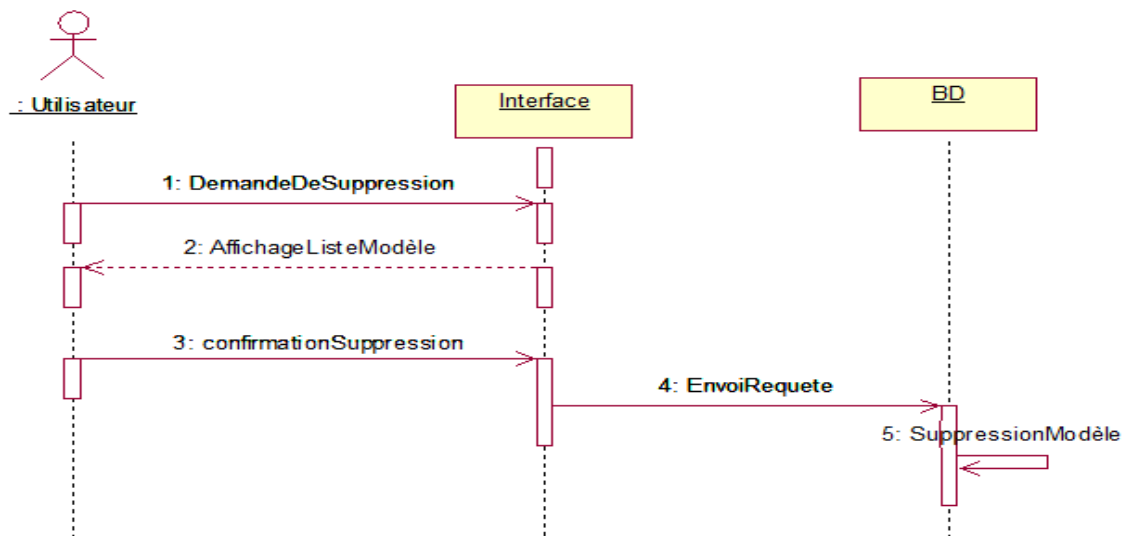


Figure 30 : Diagramme de séquence de cas d'utilisation « Supprimer Modèle »

2.2. Diagramme de collaboration

2.1.5. Diagramme de collaboration de cas d'utilisation « Modifier modèle »

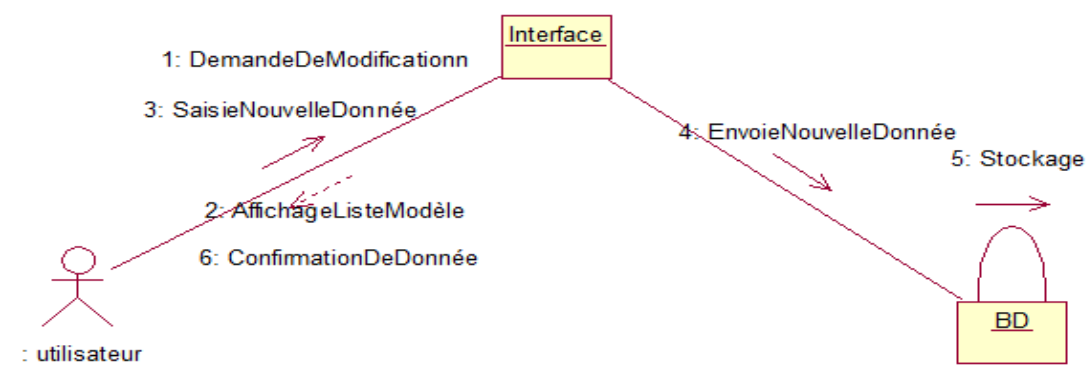


Figure 31: Diagramme de collaboration de cas d'utilisation « Modifier modèle »

2.1.6. Diagramme de collaboration de cas d'utilisation « Supprimer modèle »

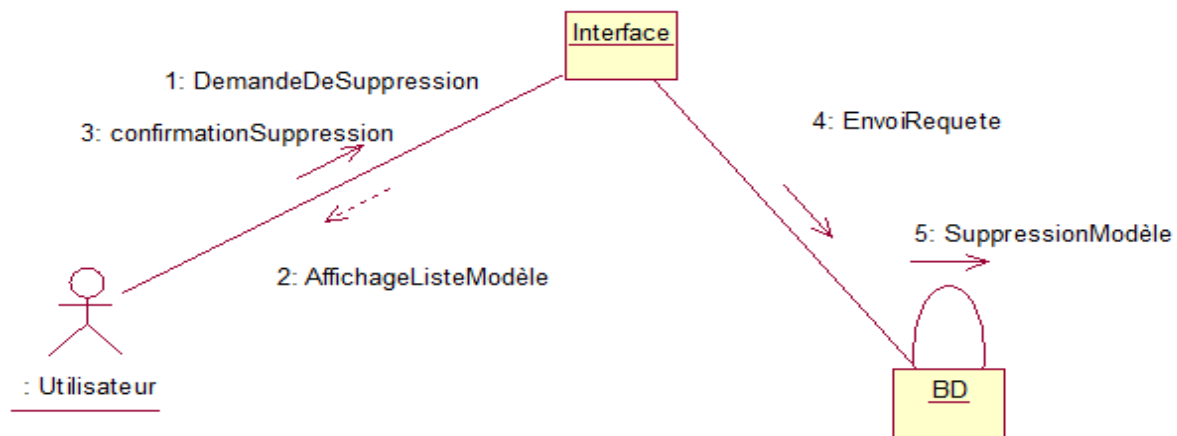


Figure 32 : Diagramme de collaboration de cas d'utilisation « Supprimer modèle »

3. Etude statique du système global

3.1. Diagramme de classes

Le diagramme de classe de notre système est composé de plusieurs classes ; une personne quel que soit un super-admin ou un simple utilisateur peut gérer des tâches planifiés, des modèles et des destinataires, le super-admin peut en plus de ces tâches créer des comptes des clients, un simple utilisateur quel que soit un client EL AMED ou un système externe en plus de gestion des tâches , des modèles et des destinataires, peut envoyer des sms. Un système externe doit avoir un compte pour qu'il puisse envoyer des messages.

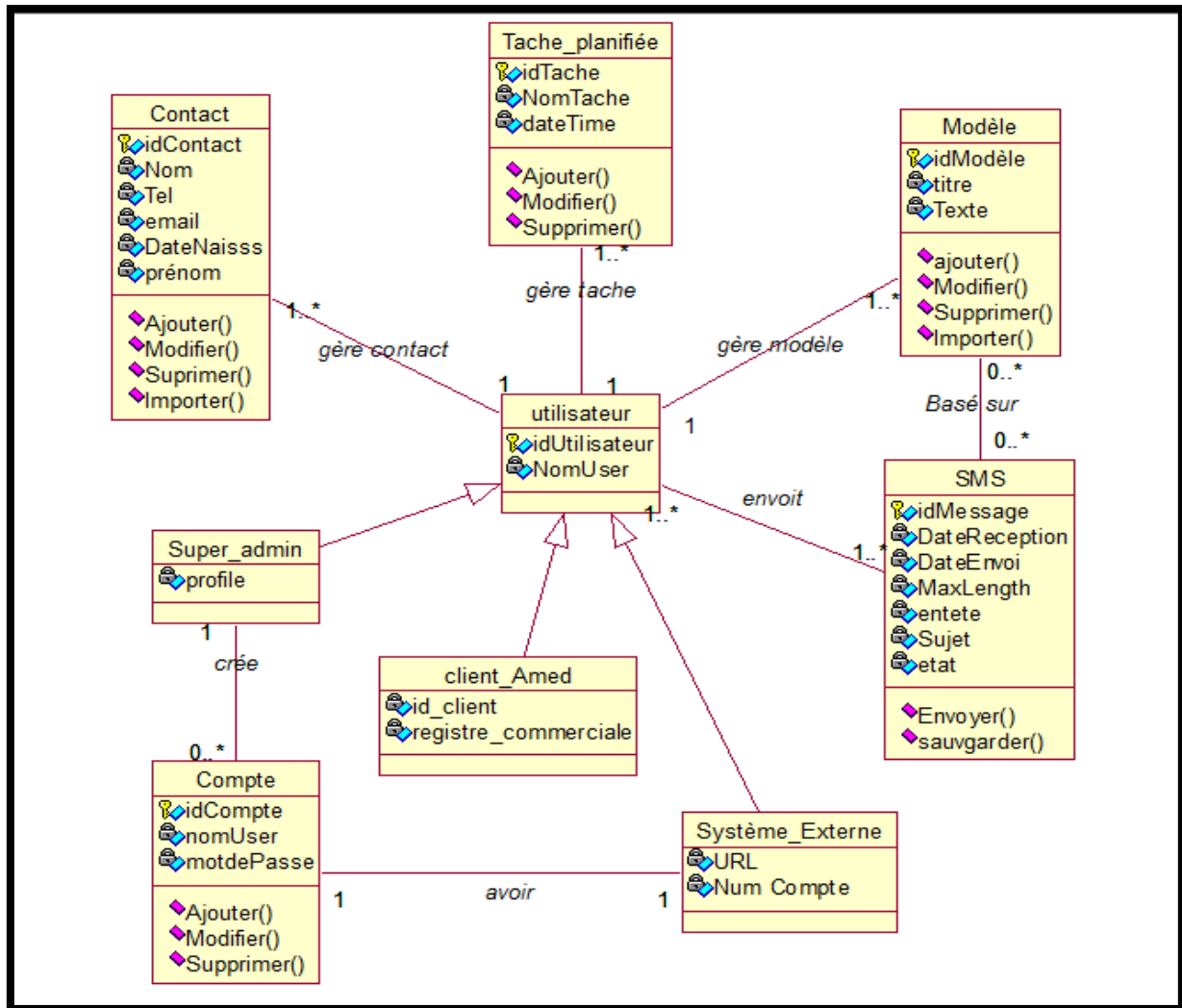


Figure 33: Diagramme de classe du système

3.2. Diagramme de déploiement

Le diagramme de déploiement sert à représenter l'utilisation de l'infrastructure physique par le système et la manière dont les composants du système sont répartis ainsi que leurs relations entre eux. Le diagramme de déploiement ci-dessous représente trois nœuds le client, le serveur d'application qui est représenté par Tomcat et le serveur de base de données nommé apache Fuseki.

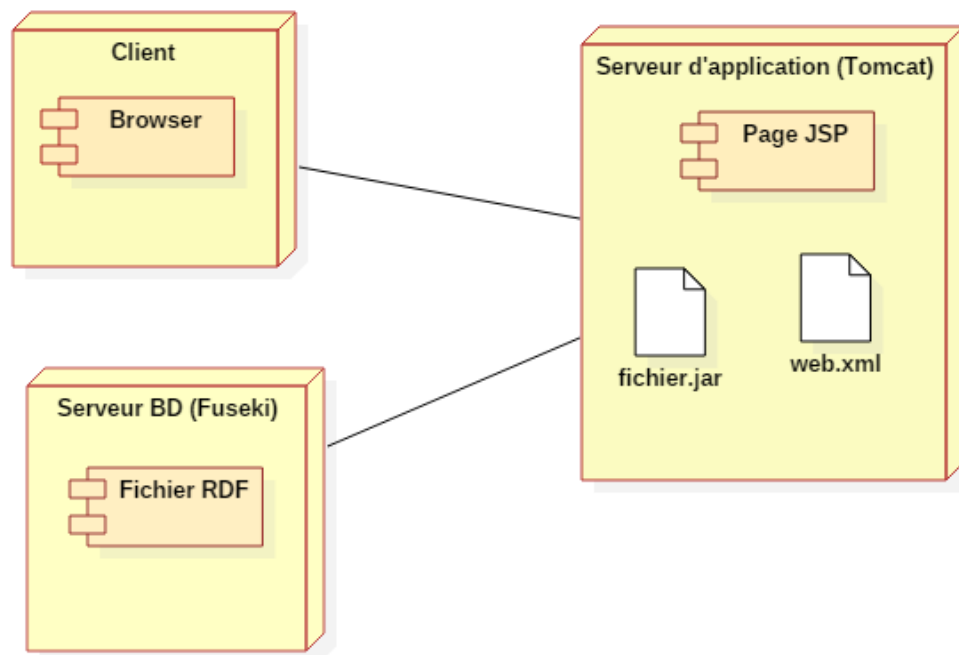


Figure 34: Diagramme de déploiement

4. Architecture & interfaçage avec l'ontologie

La mise en œuvre d'un système basé sur l'ontologie nécessite la définition d'une architecture qui permet de clarifier la relation entre les différentes entités intervenant dans la mise en place de notre application.

Dans la figure ci-dessous, nous avons besoin d'un serveur avec lequel notre système va communiquer soit pour extraire des données persistantes ou bien pour y stocker des nouvelles données. Plusieurs API (RDF API, Ontology API, SPARQL API) servent comme des outils pour que notre système puisse communiquer avec une base de connaissance TDB qui est un composant Java pour Jena destiné au stockage d'informations RDF, d'où la persistance de nos données. L'API d'inférence sert pour déduire à partir d'un ensemble de données des nouvelles informations qui améliorent notre modèle ontologique en fournissant d'autres données.

- L'API RDF est le noyau API RDF à Jena
- SPARQL sert pour interroger et mettre à jour les modèles RDF
- Fuseki est un serveur SPARQL qui peut présenter des données RDF et répondre aux requêtes sur HTTP
- TDB est un triple magasin persistant rapide qui stocke directement sur le disque

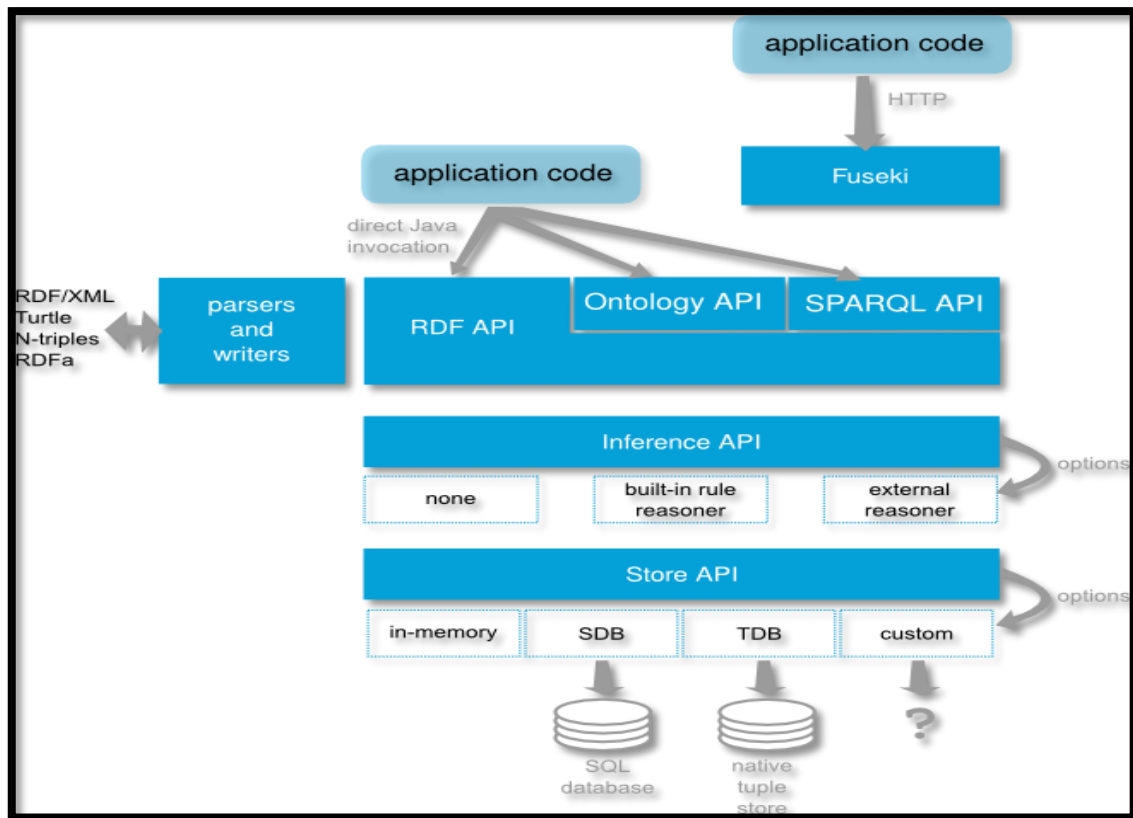


Figure 35: Outils d'intégration des Ontologies dans des applications [12]

5. Conclusion

Dans ce chapitre nous avons réalisé l'étude conceptuelle UML de notre système, nous avons fait une étude dynamique traduite par les diagrammes de séquence et de collaboration de quelques cas d'utilisation et une étude statique en représentant le diagramme de classe et le diagramme de déploiement pour mieux comprendre l'architecture de notre application. Enfin nous avons donné une architecture permettant de représenter l'interfaçage avec l'ontologie.

SECTION III : MISE EN ŒUVRE

CHAPITRE V : DÉVELOPPEMENT & INTÉGRATION

1.Introduction

Dans ce chapitre, nous présentons la partie réalisation et mise en œuvre de notre travail. Pour cela, nous présentons, en premier lieu, l'environnement de travail et les outils de développement utilisés. En second lieu, nous élaborons une présentation des différentes interfaces utilisateur

2.Architecture de l'application

Notre système suit une architecture 3 tiers : la couche présentation, la couche Business et la couche de données c'est le modèle MVC : Model, Vue, Controller. Le Vue représente les pages JSP, le contrôleur est désigné par la couche métier qui est la servlet et enfin le modèle représente notre base RDF.

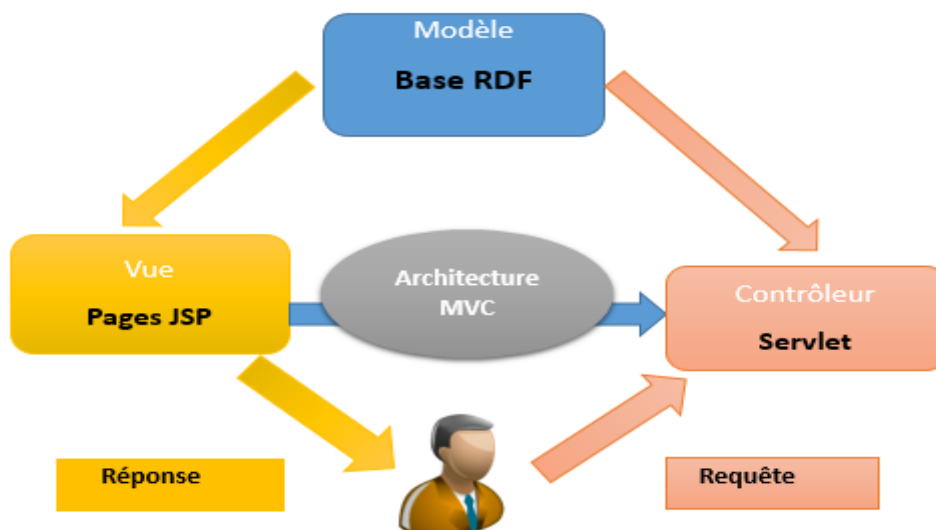


Figure 36 : Architecture 3 tiers de notre Système

3. Environnement de développement

3.1. Environnement Logiciel

3.1.1. Système exploitation : Windows 7

Système	
Évaluation :	Obtention de l'évaluation de l'ordinateur
Processeur :	Genuine Intel(R) CPU T2080 @ 1.73GHz 1.74 GHz
Mémoire installée (RAM) :	2,00 Go
Type du système :	Système d'exploitation 32 bits
Stylet et fonction tactile :	La fonctionnalité de saisie tactile ou avec un stylet n'est pas disponible sur cet écran
Paramètres de nom d'ordinateur, de domaine et de groupe de travail	
Nom de l'ordinateur :	Ghada-PC
Nom complet :	Ghada-PC

Figure 37 : Système d'exploitation

3.1.2. Eclipse Java EE IDE

Eclipse IDE permet potentiellement de créer des projets de développement mettant en œuvre n'importe quel langage de programmation. Il est principalement écrit en java. Son rôle est de produire et fournir des outils pour la réalisation de logiciels, englobant les activités de programmation (notamment au moyen d'un environnement de développement intégré) mais aussi de modélisation, de conception, de test et de reporting.



Figure 38 : logo eclipse

3.1.3. Protégé

Protégé est un environnement graphique de développement d'ontologies développé par le SMI de Stanford. Dans le modèle des connaissances de PROTEGE, les ontologies consistent en une hiérarchie de classes qui ont des attributs, qui peuvent eux-mêmes avoir certaines propriétés. L'édition des listes de ces trois types d'objets se fait par l'intermédiaire de l'interface graphique.



Figure 39 : Logo de protégé

3.1.4. API d'un opérateur Tunisien

Afin d'envoyer des SMS via une plateforme, nous avons besoin d'une API qui sert comme interfaçage entre notre application et les abonnés. Après des recherches réalisées sur les trois opérateurs tunisiens (Tunisie télécom, Ooredoo, Orange), Nous avons constaté que Tunisie Telecom était l'opérateur le plus adapté de point de vue coût. Framework JENA. Des technologies dont nous sommes besoin pour servir comme interfaçage entre l'expéditeur et le destinataire : SMPP et Kannel.

3.1.4.1. SMPP (Short Message Peer-to-Peer)

Le SMPP (Short Message Peer to Peer) est un protocole standard d'échange qui permet le transfert des SMS entre le SMSC et l'ESME. Il utilise en général deux connexions TCP/IP, l'une pour l'envoi de données (Transmitter) et l'autre pour la réception (Receiver). Il existe un autre mode (Transceiver) où l'envoi et la réception de données sont faits sur la même connexion TCP/IP.

3.1.4.2. Kannel :

Kannel est une passerelle WAP et SMS Open Source (source libre). Lancé en mars 1999, le projet est à l'initiative de la compagnie Finlandaise WAPIT. La passerelle possède trois interfaces chacune ne pouvant communiquer qu'avec un type d'équipement spécifique :

- **Les centres SMS (SMSC)**, utilisant divers protocoles.
- **Les serveurs http**, pour les contenus WAP et SMS et fournissent le contenu des « WAP push ». HTTP est utilisé pour le “pull”, et PAP pour le push.
- **Les terminaux WAP**, implémentant la pile de protocole WAP et (pour le « push ») WAP Push client. [(BAO, 2015)]

Kannel est conçue pour pouvoir communiquer avec plusieurs SMSC utilisant des protocoles différents. Chaque compte SMSC est acheté auprès d'un opérateur mobile. A chaque compte correspond un numéro expéditeur lorsqu'un message est envoyé via ce compte. Amed doit avoir un compte SMSC pour pouvoir communiquer ses SMS. El Amed à son rôle doit avoir un compte auprès de Tunisie Télécom pour envoyer ses SMS.

Mode de fonctionnement du Kannel :

Le client se connecte au Centre SMS appelé SMSC

- Lorsqu'un message est arrivé à partir d'un téléphone, le SMSC l'envoie au client qui par la suite devra l'acquitter.
- Lorsque le client voudra envoyer un message, il émet une requête au SMSC qui l'acquittera.
- Lorsque le client est servi il se déconnecte.

3.2. Choix techniques

- OWL (Web Ontology Language)
- RDF (Ressource Description Framework)
- Plateforme J2EE (Java 2 Enterprise Edition)

4. Les serveurs utilisés

4.1. Serveur Fuseki V2

Apache Jena Fuseki est un serveur SPARQL. Il peut fonctionner comme un service de système d'exploitation, comme une application Web Java (fichier WAR), et en tant que serveur autonome et indépendant. Il assure la sécurité et dispose d'une interface utilisateur pour l'administration du serveur. Il fournit des protocoles SPARQL 1.1 de recherche et de mise à jour, ainsi que le protocole SPARQL Graph Store. Fuseki est étroitement intégré à TDB pour fournir une couche de stockage robuste, transactionnelle et persistante. Il peut être utilisé pour fournir le moteur de protocole pour d'autres systèmes RDF d'interrogation et de stockage. Notre choix de Fuseki est argumenté par le fait que ce serveur offre une grande sécurité de telle sorte qu'il est par défaut verrouillé d'où la création de Data Set sera limitée en local host.

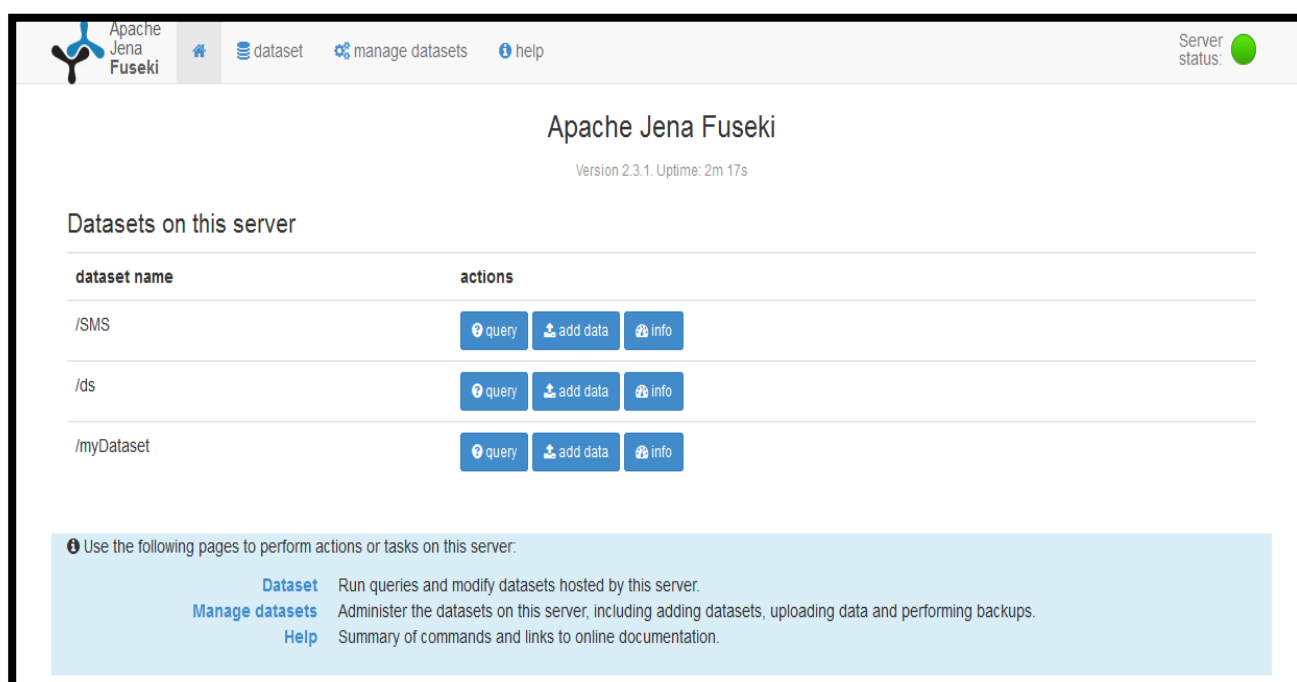


Figure 40: Interface d'accueil de serveur apache Jena Fuseki V2

4.2. Triple Store

Un nombre croissant des applications web sémantique stocke un très grand nombre de (**sujet, prédicat, objet**) dans les moteurs de stockage spécialisés appelés triple- store. Souvent , triples- stores sont principalement utilisés comme des magasins de données simples , à savoir , pour

l'insertion et l'extraction de grandes quantités de triplets , mais ne pas utiliser des fonctionnalités plus avancées telles que l'inférence logique , etc.[6]



Figure 41: Un exemple de triplet

```

1 PREFIX j.0: <http://www.owl-ontologies.com/SMSOntology.owl#>
2
3 SELECT *
4 WHERE
5 {
6   ?x j.0:hasMessageBody ?nom.
7   ?x j.0:hasTitle ?titre.
8   ?x j.0:IdModele ?idd.
9 }
10

```

Figure 42 : Requête SPARQL

x	nom	titre	idd
1	j.0:a3681cd3-18d8-4af3-a274-cde8f791642b	"Mr MERCI d'avoir assister un réunion des parents"	"a3681cd3-18d8-4af3-a274-cde8f791642b"
2	j.0:6f68161f-1cd1-46f4-99b5-317efdfb71ca	"merci d'avoir payer la dernière tranche de votre inscription avant le 2/9/2016"	"6f68161f-1cd1-46f4-99b5-317efdfb71ca"

Figure 43: Résultat de la requête SPARQL

4.3. Serveur Tomcat V8

Apache Tomcat est un conteneur web libre de servlets et JSP Java EE. Issu du projet Jakarta, c'est un des nombreux projets de l'Apache Software Foundation. Il implémente

les spécifications des servlets et des JSP du Java Community Process², est paramétrable par des fichiers XML et des propriétés, et inclut des outils pour la configuration et la gestion. Il comporte également un serveur HTTP.



Figure 44 : Logo de Tomcat V8

5. Le Framework Bootstrap

Dans notre projet, nous avons utilisé le framework Bootstrap qui est une collection d'outils utile à la création de sites et d'applications web. C'est un ensemble qui contient des codes HTML et CSS, des formulaires, boutons, outils de navigation et autres éléments interactifs, ainsi que des extensions JavaScript en option. C'est l'un des projets les plus populaires sur la plate-forme de gestion de développement GitHub.



Figure 45 : Logo de Bootstrap

6. Conclusion

Dans ce chapitre nous avons présenté l'architecture de notre application, ainsi l'environnement matériel et logiciel. En plus nous avons cité les différentes technologies utilisées ainsi que les types de serveurs utilisés : serveur d'application tel que Tomcat et serveur de la base de données tel que Fuseki.

CHAPITRE VI : VÉRIFICATION & TESTS

1.Introduction

Ce chapitre est consacré pour représenter nos interfaces et nos tests de différentes fonctionnalités de notre système comme la gestion des modèles, la gestion des tâches planifiées et la gestion des contacts, l'envoi des SMS.

2.Les interfaces réalisées

2.1. Interface de présentation

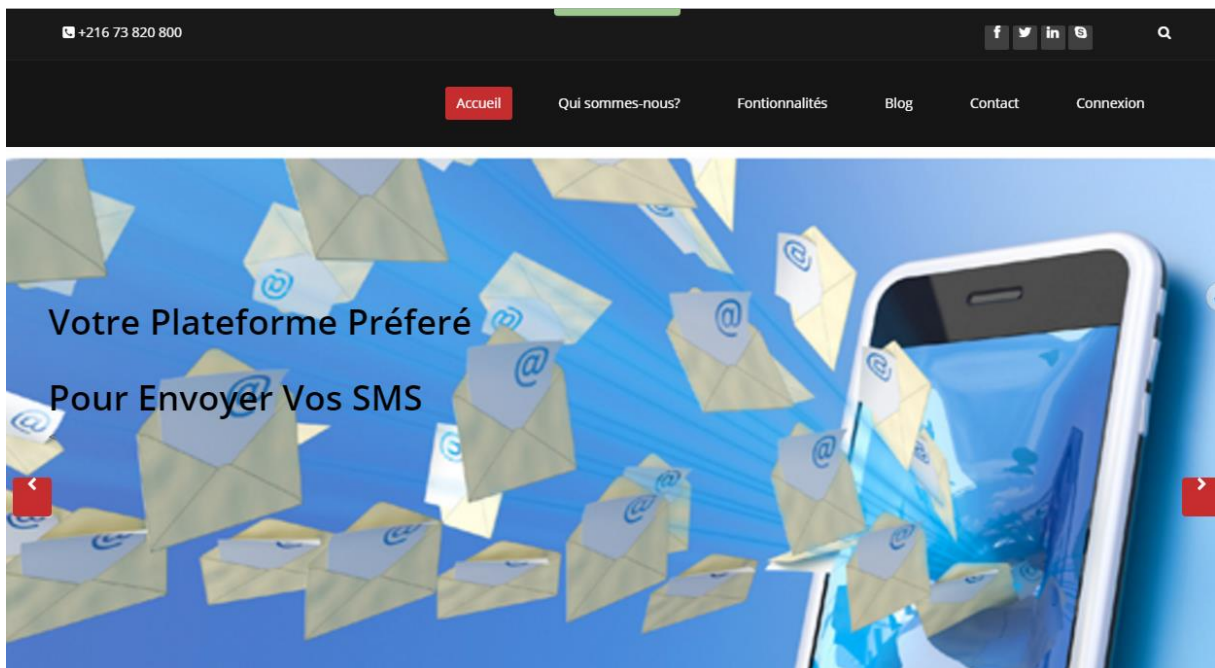


Figure 46 : Interface de présentation de service web

2.2. Interface d'authentification :

L'utilisateur doit saisir son login et son mot de passe pour accéder à l'interface d'accueil.

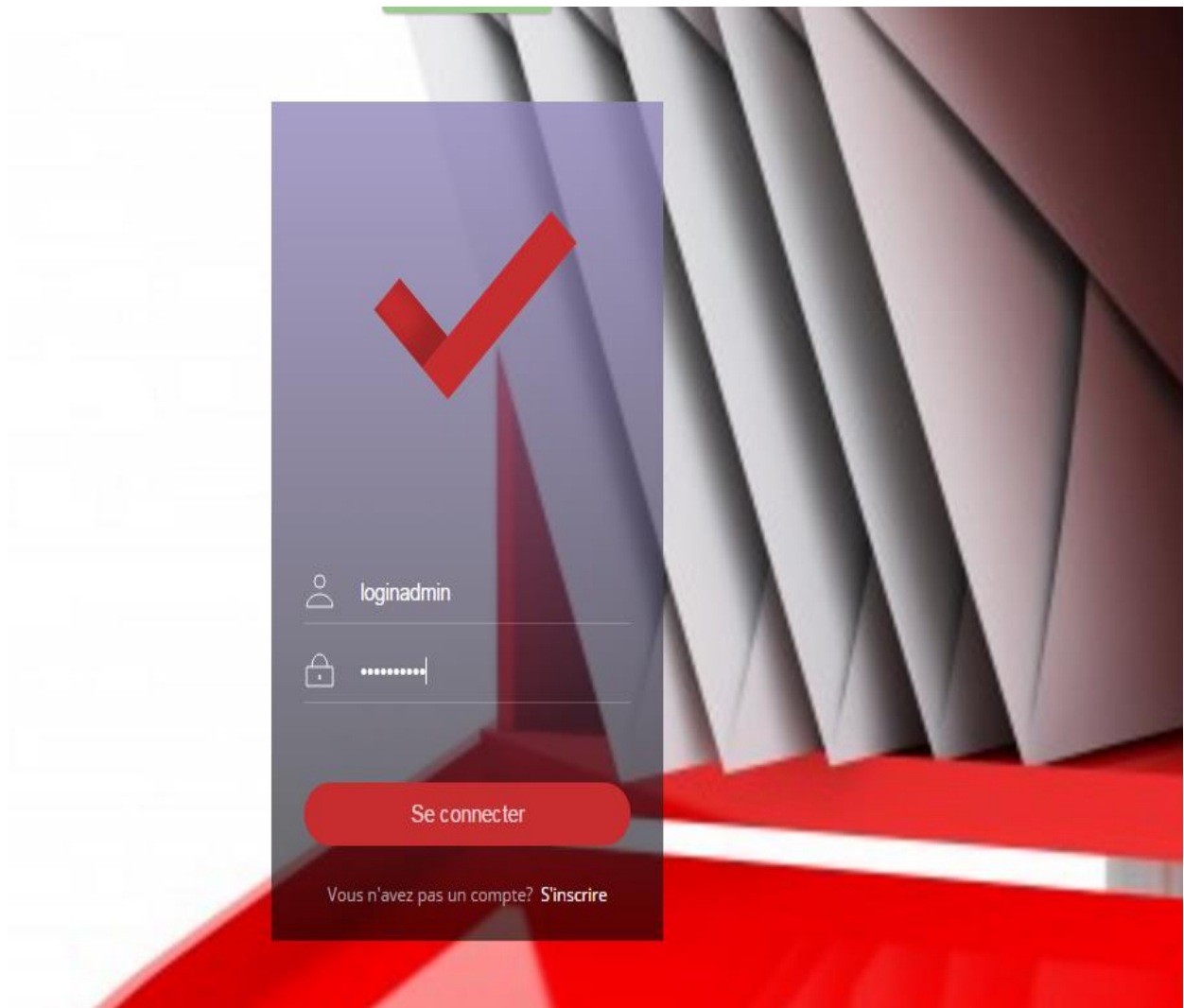


Figure 47: Interface d'authentification

2.3. Interface de gestion des tâches :

Cette interface permet la gestion des tâches planifiées (ajout, modification et suppression et consultation)

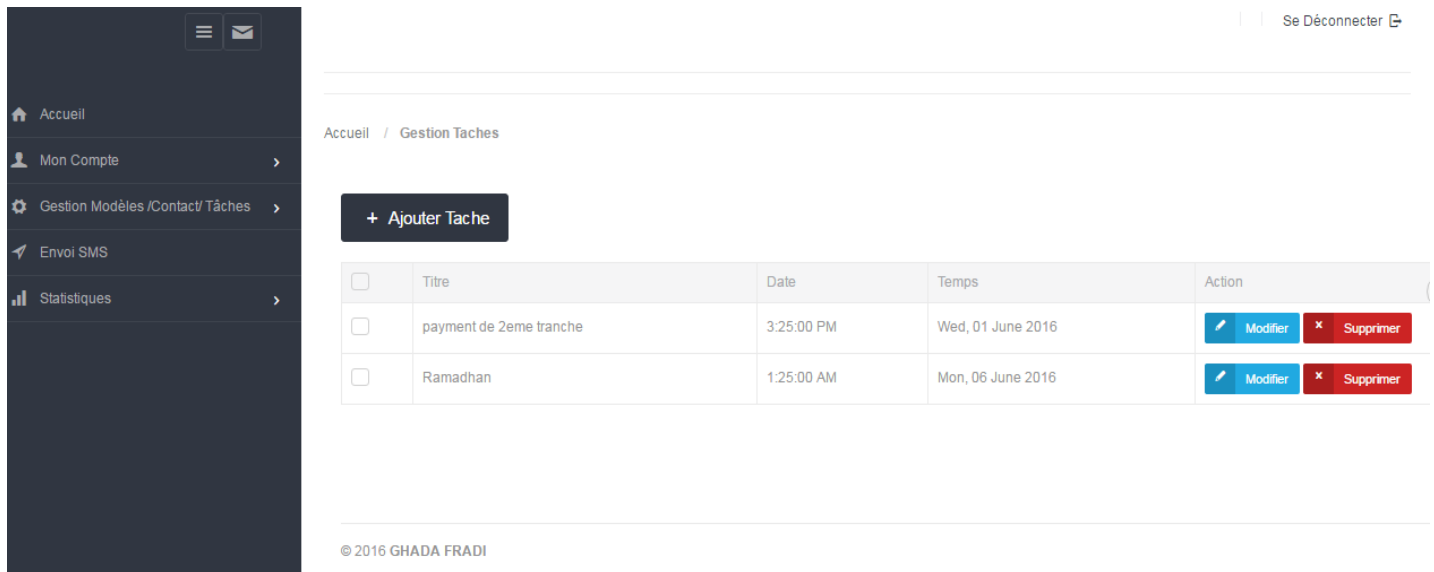


Figure 48 : Interface de gestion des tâches

3. Tableau de données de test

Login	Le nombre des caractères saisi est inférieur à 8 caractères	
Adresse email	En cas de saisi incorrect, cette l'adresse email le message suivant sera affichée	
Numéro de téléphone	Il doit être composé de 8 chiffres	

4. Conclusion

Dans ce chapitre, nous avons présenté quelques interfaces contenant des champs sur lesquels nous avons fait des contrôles de saisi pour rendre notre application porte un aspect de rigidité.

SECTION IV : RETOUR D'EXPÉRIENCE D'UNE ÉLÈVE INGÉNIEUR

1.Introduction

Dans ce chapitre, nous relatons notre expérience en tant qu'élève ingénieur intégré au sein d'une équipe qui travaille dans le cadre d'un grand projet. Cette expérience tourne autour d'un extrait de problèmes rencontrés et comment ils ont été solutionnés et autour de modalités d'échange et de collaboration entre les membres de l'équipe EL Amed.

2.Outils de travail collaboratif

2.1. Slack

Nous avons utilisés Slack qui est un environnement de travail collaboratif qui permet l'échange des fichiers, qui permet l'échange des fichiers, qui permet l'envoi de notification et l'accès à des espaces de chat. Nous l'avons utilisé par exemple pour transférer des versions de mon PFE vers mon encadreur en entreprise et pour partager quelques documents avec mes d'autre membre de notre équipe.

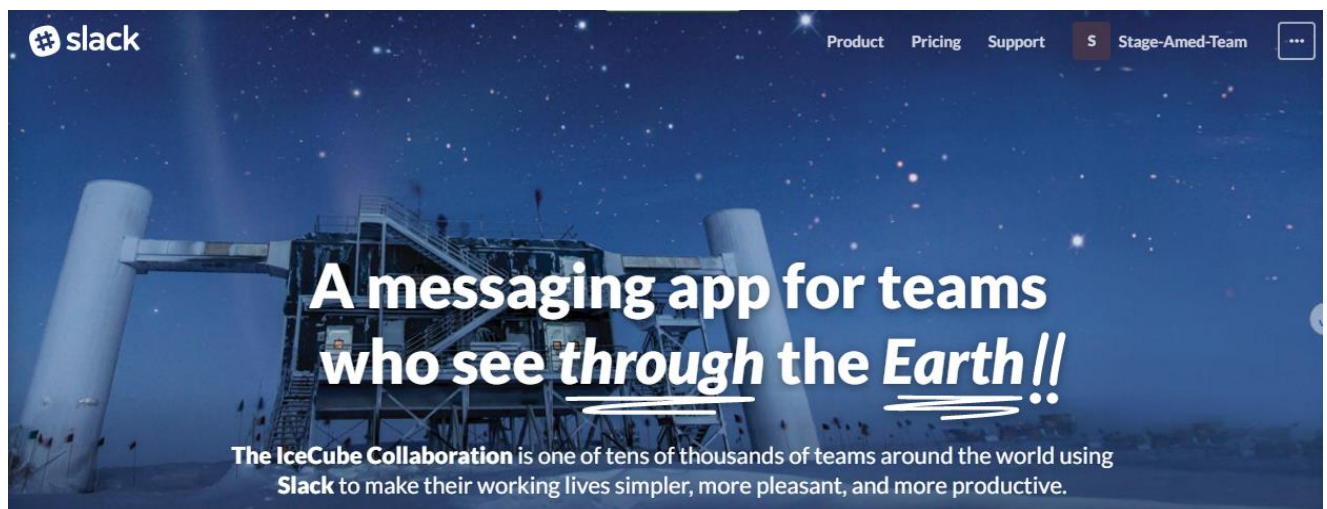


Figure 49 : Compte Slack

2.2. Gmail et google Drive



Figure 50: Compte Gmail de stagiaires

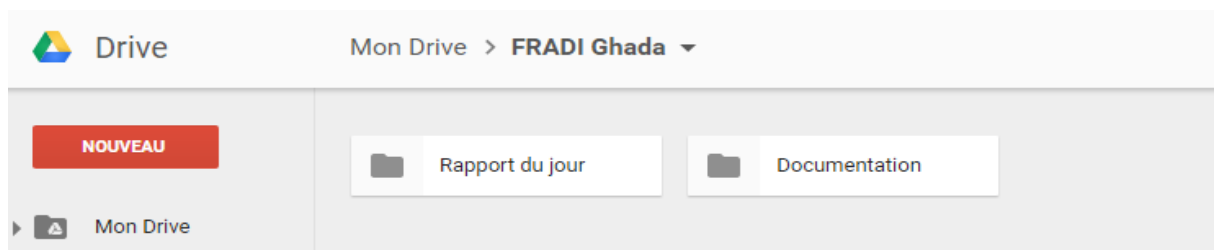


Figure 51: Rapport de jour

2.3. SVN : Subversion Number

C'est logiciel de gestion des versions sous la licence apache et BSD, il fonctionne en mode client/serveur : le serveur informatique est centralisé et unique contenant les fichiers de références, «Repository» en anglais, et un logiciel serveur subversion, le client dépose ses fichiers ou dossiers dans le serveur via un exécutable (exemple : TortoiseSVN) et à chaque modification atomique, un simple commit permet de faire la synchronisation entre le client et le serveur de référence.



Figure 52: Logo subversion

3. Les réunions

Le travail en équipe au sein d'El Amed est basé sur des réunions en cours desquelles l'un des membres de l'équipe présente son travail et les outils qu'il a utilisés. Ces réunions sont considérées comme des petites formations qui permettent aux membres d'équipe EL Amed d'entraider pour atteindre leur but final.

4. Problèmes rencontrés

Le développement de cette application a été contourné par plusieurs obstacles, ces obstacles que nous avons confrontés due à la nature du sujet qui relève du web sémantique qui est un domaine de recherche. D'une autre côté, notre sujet qui est « un système d'envoi d'sms basé sur l'ontologie » m'a mené à s'interroger des outils que nous aurons besoins pour communiquer nos SMS, à cet occasion j'ai posé cette question dans le forum de comment Ça marche et l'un des membres a répondu à ma question. La figure 52 montre la réponse que j'ai reçue par email.

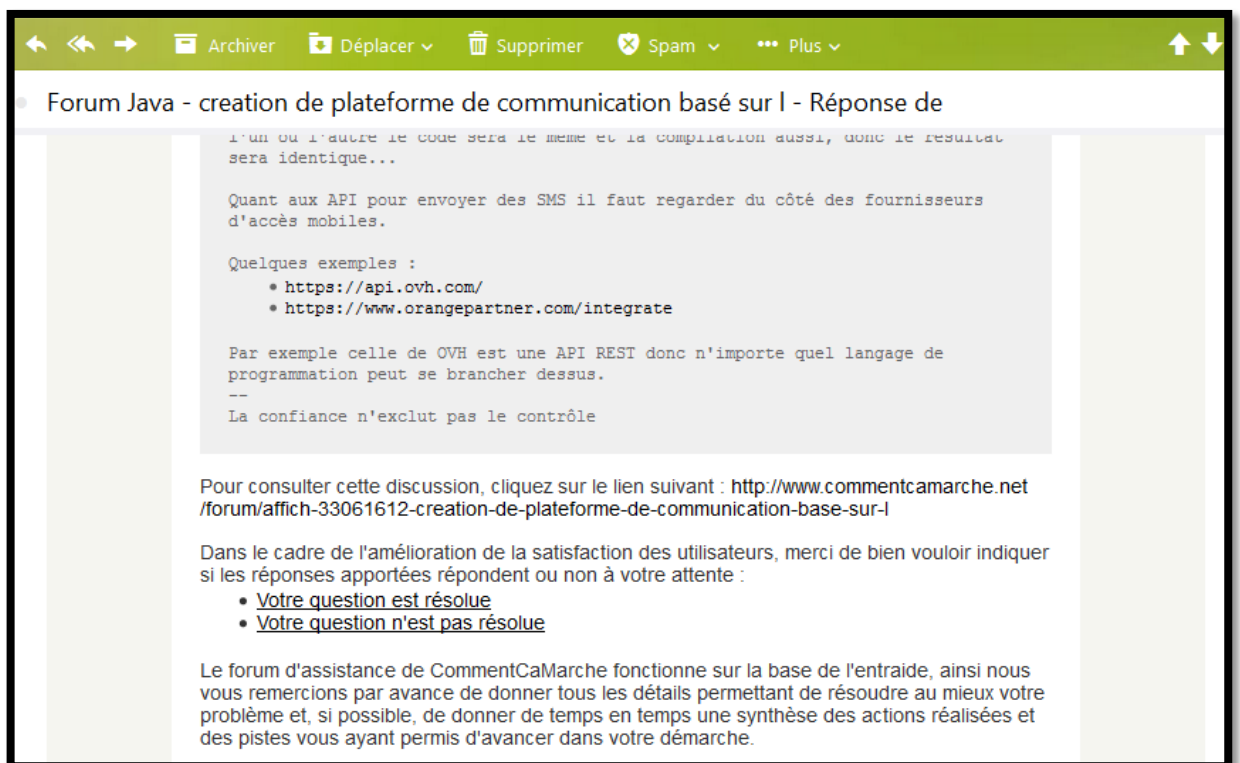


Figure 53 : Réponse de la part de Comment Ça Marche

5. Conclusion

Ce stage au sein de la société EL AMED a été une très bonne occasion pour acquérir une expérience en tant qu'ingénieur en nous permettant de s'intégrer au sein de la vie professionnelle et en apprenant des technologies évoluées.

CONCLUSION GÉNÉRALE & PERSPECTIVES

Durant ces dernières années les ontologies ont été de plus en plus utilisées dans une variété de domaine. En conséquence un fort besoin de gestion de ces ontologies en bases de données s'est fait ressentir. Aussi, des universitaires et des industriels ont proposé des solutions de persistance s'appuyant sur des SGBD existants pour permettre l'interrogation efficace de ces données tout en supportant une volumétrie importante. Contrairement aux SGBD traditionnels qui sont similaires en termes d'architecture et de modèle de stockage, les travaux menés ont conduit à la définition de base de données dites sémantiques qui présentent une grande diversité en termes de formalismes d'ontologies supportés, de modèles de stockage et d'architectures utilisés

Au terme de notre projet de Fin d'Etude où il était question de réaliser « **un système d'envoi des sms basé sur une ontologie** » en langage Java, nous pouvons évaluer le degré d'atteinte des objectifs exprimés par le groupe EL Amed. En effet, nous avons mis sur les rails une base de connaissances, nous avons lié cette base à une plateforme d'envoi de sms par le biais d'un programme java. Notre plateforme permet à l'utilisateur authentifié de gérer des contacts, des modèles, des tâches planifiées, d'envoyer un sms, de consulter ses sms envoyés ; les différentes opérations effectuées sur notre système d'envoi d'SMS peuvent être vérifiées au niveau de la base de données RDF.

Cependant nous avons rencontrés quelques difficultés dues majoritairement à la nature du sujet qui relève du web sémantique qui est un domaine jeune dans la recherche scientifique et 'autant plus en Tunisie. Les problèmes émanant aussi de la limitation des applications qui sont basés sur le web 3.0 qui sont très peu nombreuses. L'envoi des sms du PC vers un téléphone mobile semble généralement classique, mais dans la réalité l'initiative de l'EL AMED à intégrer de nouvelles technologies tel que les ontologies l'a rendu ardue.

Notre système de notification par SMS basé sur une ontologie nécessite encore une étape d'intégration de raisonneur pour introduire à chaque fois des nouvelles règles et en inférer de nouvelles connaissances et ainsi garantir une évolutivité continue de notre application.

Le groupe EL Amed porte l'ambition de transformer cette application d' « envoi d'sms basé sur une ontologie » en tant qu'un service web duquel d'autres acteurs pourraient tirer profit en s'inscrivant.

REFERENCES

- [1]: Agrawal, R. (2007, Février 20). Difference between OWL Lite, DL, and Full
- [2] : Baneyx, A. (2007). Construire une ontologie de la Pneumologie Aspects théoriques, modèles et expérimentations. Paris : Université Pierre et Marie Curie.
- [3] : BAO, P. A. (2015, 10 23). KANNEL : Definition et mode de fonctionnement.
- [4] : Croutte, R. B. (Juin 2014).La diffusion des technologies de l'information et de la communication dans la société française. 142 rue du Chevaleret.
- [5] : FEYLER, F. (2013). Le Web sémantique. Une approche nouvelle de l'accès à l'information pertinente.
- [6] : Galopin, A. (2015), Modélisation ontologique des recommandations de pratique clinique pour une aide à la décision à niveaux d'abstraction variables, thèse de Doctorat, Université Pierre et Marie Curie - Paris VI.
- [7] : Gandon, F. (2016). Mooc Web sémantique et Web de données.
- [8] : <http://blog.c-marketing.eu/du-web-1-0-au-web-4-0/>, (Du web 1.0 au web 4 .0) ,2012
- [9]: https://jena.apache.org/getting_started/, (Getting started with Apache Jena), 2011
- [10]: <http://slideplayer.com/slide/5265697/>, (jena inference), 2015
- [11]:<http://www.developpez.net/forums/d5932-4/general-developpement/alm/outils/outil-modelisation-uml-utilisez/>.(Quel outil de modélisation UML utilisez-vous ?)
- [12] :http://www.ibm.com/support/knowledgecenter/SS8PJ7_9.1.2/com.ibm.xtools.modeler.doc/topics/twrkucd.html?lang=fr (création des diagrammes de cas d'utilisation)
- [13]:<http://www.planetoscope.com/electronique/718-nombre-de-sms-envoyes-dans-le-monde.html>, (statistiques mondiales en temps réel),2012
- [14]:<http://www.sciencedirect.com/science/article/pii/S030643791000133X>(3XL: Supporting efficient operations on very large OWL Lite triple-stores), 2011.
- [15] : Skopal, V. D. (2011). Information Systems. Volume 36, Pages 707-824.

ANNEXES

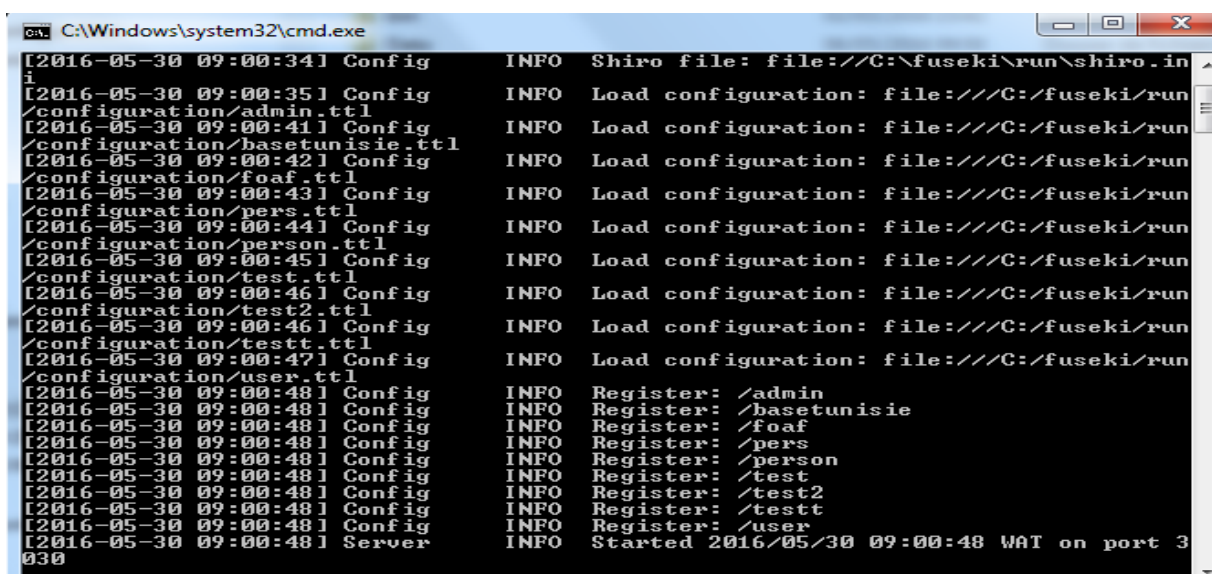
Notre base nommé « test » :

```
@prefix : <http://base/#>.
@prefix tdb: <http://jena.hpl.hp.com/2008/tdb#>.
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix ja: <http://jena.hpl.hp.com/2005/11/Assembler#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix fuseki: <http://jena.apache.org/fuseki#>.

:service_tdb_all a          fuseki:Service ;
    rdfs:label              "TDB test" ;
    fuseki:dataset          :tdb_dataset_readwrite ;
    fuseki:name              "test" ;
    fuseki:serviceQuery     "query" , "sparql" ;
    fuseki:serviceReadGraphStore "get" ;
    fuseki:serviceReadWriteGraphStore
        "data" ;
    fuseki:serviceUpdate     "update" ;
    fuseki:serviceUpload     "upload" .

:tdb_dataset_readwrite
    a          tdb:DatasetTDB ;
    tdb:location "C:\\fuseki\\run\\databases\\test" .
```

Figure 54 : Fichier de configuration de serveur Fuseki



```
C:\Windows\system32\cmd.exe
[2016-05-30 09:00:34] Config INFO Shiro file: file:///C:/fuseki/run/shiro.in
[2016-05-30 09:00:35] Config INFO Load configuration: file:///C:/fuseki/run/configuration/admin.ttl
[2016-05-30 09:00:41] Config INFO Load configuration: file:///C:/fuseki/run/configuration/basetunisie.ttl
[2016-05-30 09:00:42] Config INFO Load configuration: file:///C:/fuseki/run/configuration/foaf.ttl
[2016-05-30 09:00:43] Config INFO Load configuration: file:///C:/fuseki/run/configuration/pers.ttl
[2016-05-30 09:00:44] Config INFO Load configuration: file:///C:/fuseki/run/configuration/person.ttl
[2016-05-30 09:00:45] Config INFO Load configuration: file:///C:/fuseki/run/configuration/test.ttl
[2016-05-30 09:00:46] Config INFO Load configuration: file:///C:/fuseki/run/configuration/test2.ttl
[2016-05-30 09:00:46] Config INFO Load configuration: file:///C:/fuseki/run/configuration/testt.ttl
[2016-05-30 09:00:47] Config INFO Load configuration: file:///C:/fuseki/run/configuration/user.ttl
[2016-05-30 09:00:48] Config INFO Register: /admin
[2016-05-30 09:00:48] Config INFO Register: /basetunisie
[2016-05-30 09:00:48] Config INFO Register: /foaf
[2016-05-30 09:00:48] Config INFO Register: /pers
[2016-05-30 09:00:48] Config INFO Register: /person
[2016-05-30 09:00:48] Config INFO Register: /test
[2016-05-30 09:00:48] Config INFO Register: /test2
[2016-05-30 09:00:48] Config INFO Register: /testt
[2016-05-30 09:00:48] Config INFO Register: /user
[2016-05-30 09:00:48] Server INFO Started 2016/05/30 09:00:48 WAT on port 3030
```

Figure 55: Démarrage du serveur Fuseki