# Table de matières

## Chapitre3: Conception et réalisation

# Table de figures

# Introduction générale

Les besoins d'accéder de façon uniforme à des sources de données multiples et réparties sont chaque jour de plus en plus fort dans les systèmes administratifs, éducatifs, et particulièrement, dans nos hôpitaux, là où l'accès à l'information pertinente dans un temps limité est primordiale.

Les applications réparties sont apparues pour répondre à ces besoins, qui grâce à eux les informations ne sont plus stockées sur un seul et même serveur mais peuvent être dispersées aux quatre coins de la planète et ceci sur un nombre infini de serveurs, mais malgré cela elles ont été beaucoup critiquées sur le fait qu'elles n'autorisaient guerre l'incompatibilité entre les langages de programmation, un inconvénient parmi plein d'autres qui a suscité une nouvelle technologie multi langages, multiplateformes, facile à implémenter sur différents plateformes qu'est les services web.

Par ailleurs, les services Web sont devenus le nouveau point de convergence de l'ensemble des acteurs du marché de l'informatique, grâce à la disponibilité d'outils et de standards permettant la découverte et l'invocation automatisées des fonctions métiers via un échange de messages informatisés (SOAP, UDDI, WSDL). Cependant ces mêmes standards, largement utilisés actuellement, ne permettent pas de décrire les fonctionnalités des services Web que de manière syntaxique ce qui rend leur découverte moins efficace.

L'utilisation d'une ontologie a pour ambition de lever cette difficulté, elle permet un meilleur partage de connaissances par les différents intervenants de la plateforme (aussi bien par l'homme que par la machine), ainsi que la représentation sémantique de leurs contenus.

# Chapitre 1

# Les Services Web

# 1. Introduction

L'amélioration des moyens de communication a depuis toujours été l'une des préoccupations majeures de l'humanité et ce n'est évidemment pas avec la complexité grandissante de nos systèmes d'information, toujours demandeurs de plus de ressources, que la tendance s'inversera. Actuellement, nous pourrions même dire qu'il est indispensable à faire communiquer l'ensemble des systèmes sachant les machines isolés ne disposent plus de la quantité d'informations suffisantes pour répondre à la majorité des demandes.

Avec l'accroissement de la complexité des systèmes, il devient également de plus en plus difficile de permettre la communication entre machines, c'est principalement pour répondre à cette problématique que sont nées les premières technologies d'applications réparties

A la fin des années quatre-vingt-dix, plusieurs environnements sont utilisés pour mettre en pratique les applications réparties, il s'agit principalement de RMI, RPC, CORBA et DCOM ,cependant l'interopérabilité qu'offraient ces derniers n'autorisait guerre l'incompatibilité entre les langages de programmation, un inconvénient parmi plein d'autres qui a suscité une nouvelle technologie multi langages, multiplateformes, facile à implémenter sur différents plateformes qu'est les services web, considéré aujourd'hui comme le point de convergence technologique d'acteurs de divers domaines : e-commerce, e-learning e-government.

# 2. Définition

Les services Web représentent actuellement l'ensemble de standards les plus connus pour la réalisation des applications Internet. Cette technologie repose sur des standards très populaires, tels que le XML ou le protocole HTTP.

Les Web services ont été proposés initialement par IBM et Microsoft, puis en partie standardisés par le consortium du World Wide Web : le W3C.

Selon W3C : « Un service Web est un système conçu pour permettre l'interopérabilité des applications à travers un réseau. Il est caractérisé par un format de description interprétable/compréhensible automatiquement par la machine (en particulier WSDL). D'autres systèmes peuvent interagir avec le service Web selon la manière prescrite dans sa description et en utilisant des messages SOAP, généralement transmis via le protocole HTTP et sérialisés en XML et en d'autres standards du Web ».[36]

Du point de vue d'un client, un service Web est une "boite noire" qui ne donne pas de détails sur son implémentation et qui expose, comme le montre la figure 1, un ou plusieurs ports

(points d'accès). Chaque port rassemble un ensemble d'opérations qui réalisent la fonctionnalité du service.

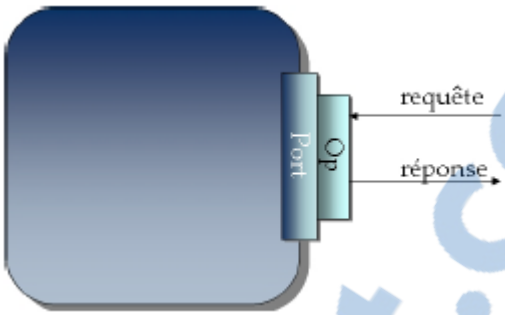



**Figure 1 Service web simple**

Un service Web (comme celui présenté précédemment) reçoit une requête de la part d'un client, traite cette requête et à la fin du traitement envoie sa réponse. Néanmoins, la réalisation interne de cette fonctionnalité peut être plus complexe. Pour cela, le service peut utiliser les fonctionnalités d'autres services (voir la figure 2), fait dont le client n'est pas conscient. Cela nécessite une coordination entre les différents appels des services utilisés. Cette caractéristique de taille sera détaillée dans le troisième chapitre.[22]

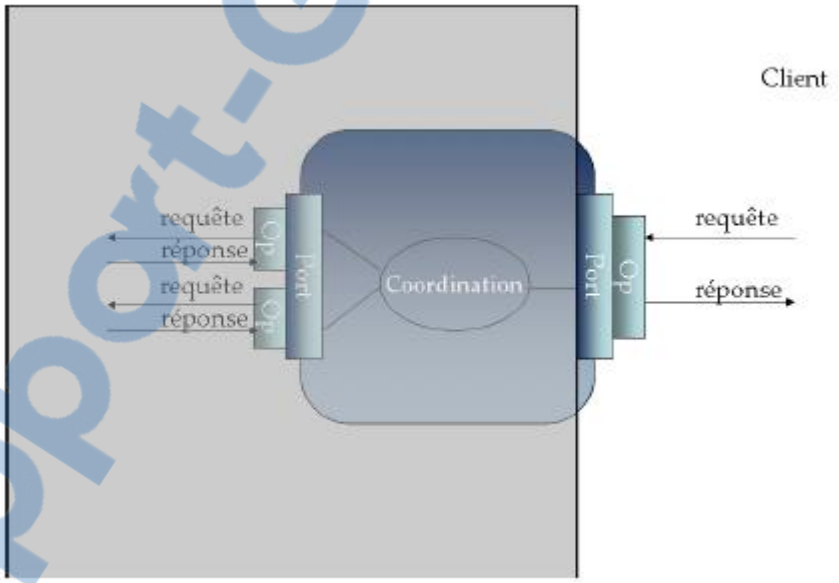



**Figure 2 Service web composite**

# 3. La base de conception SOA

L'architecture des web services est principalement orientée autour du modèle SOA (Services Oriented Architecture) et pour beaucoup d'ailleurs, les termes SOA et web services sont des synonymes. Il existe tout de même une différence fondamentale entre les deux car SOA n'est

pas en soit une technologie, mais plutôt un principe de conception alors que les web services, sont eux, des implémentations technologiques.

SOA est une architecture logicielle mettant en œuvre des connexions de couplage lâche(faible) entre divers composants logiciels appelés services. Par définition un service a pour but de proposer, en faisant éventuellement appel à d'autres services complémentaires ou concurrents, un résultat particulier en fonction des informations qui lui ont été envoyées par les tiers. Chacun de ces services, dont les interfaces et les contrats sont obligatoirement connus, répond à des spécifications précises et est considéré en tant que processus métier.

SOA est né en réponse aux problèmes des systèmes existants qui étaient beaucoup trop liés à un langage ou une plateforme particulière et qui ne permettaient aucune évolution sans causer le disfonctionnement de certaines fonctionnalités et une augmentation drastique des coûts de maintenance.[23]

Une architecture à services est, ainsi, l'ensemble des politiques, patrons de conception et environnements qui permettent l'intégration des différentes fonctionnalités exposées comme des services.

La figure 3 présente les principaux acteurs qui interviennent dans une architecture à services.

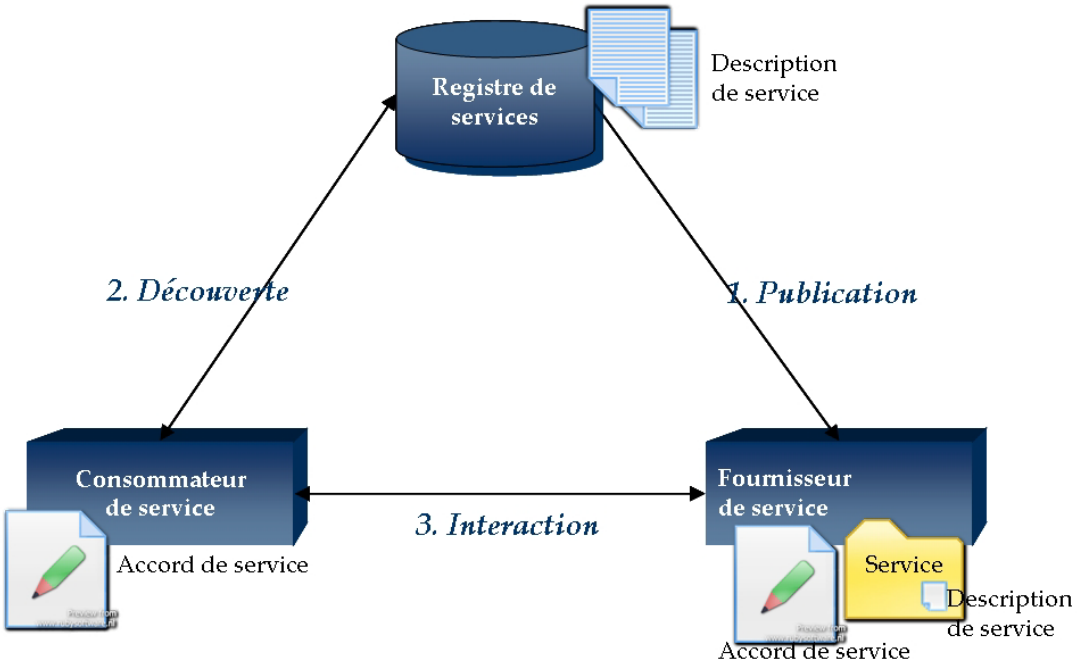



**Figure 3 Les acteurs d'une architecture à services AOS**

Le premier acteur est le ***fournisseur du service*.** Il représente une personne/une organisation qui fournit des fonctionnalités sous forme de service. Après le développement de ce dernier, un fournisseur doit mettre à disposition des éventuels utilisateurs les informations nécessaires pour pouvoir l'utiliser, c'est-à-dire la *description du service*, en la publiant dans un *registre de*

*services* (appelé également annuaire de services ou *courtier de services*) qui sert d'acteur intermédiaire entre fournisseurs et consommateurs.

Un utilisateur de service, nommé ***client*** interroge le registre de services pour obtenir ceux correspondant à ses besoins. La découverte d'un service est réalisée grâce à sa description disponible dans l'annuaire.

Après avoir sélectionné le service demandé, le client peut, dans certains cas, négocier auprès du fournisseur les termes suivant lesquels il peut l'utiliser.

A la fin de la négociation, un *accord de service* est réalisé entre le consommateur et le fournisseur. La plupart du temps, cet accord contractualise les termes de l'utilisation par le consommateur sans garantie totale du résultat (technique appelée le Best-effort).

Grâce aux informations disponibles dans la description du service, le consommateur peut, dès lors, réaliser la liaison et appeler les fonctionnalités du service.

Les éléments d'un tel environnement peuvent être divisés en deux catégories (voir figure 4) :

- Les mécanismes de base (fonctionnels) qui permettent la publication, découverte, composition, négociation, contractualisation, invocation des différents services ;
- Les mécanismes additionnels qui assurent la prise en charge de la qualité de service (les besoins non-fonctionnels) tels que la sécurité, les transactions [22]

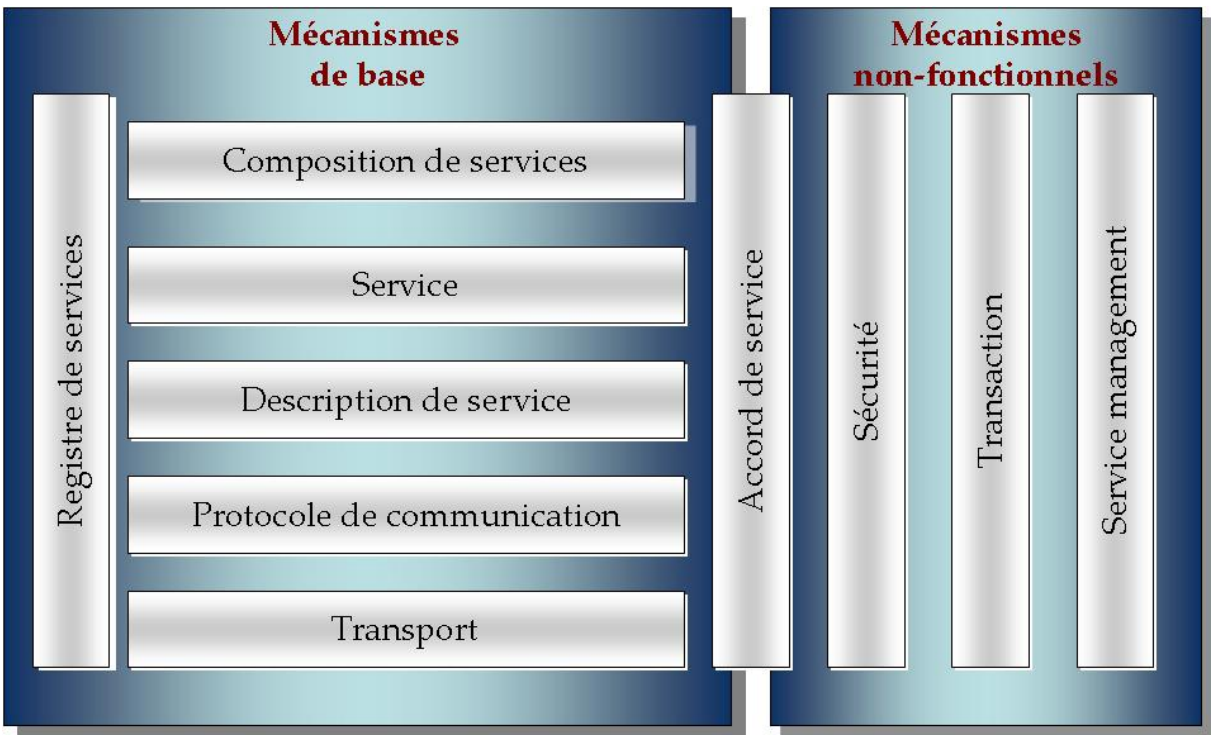



**Figure 4 Environnement d'intégration de service**

Les aspects fonctionnels incluent :

- **La couche Transport** : elle représente le mécanisme employé pour déplacer les demandes de services du consommateur au fournisseur, et les réponses du fournisseur de services au consommateur de services.
- **Le Protocole de communication de services** : il s'identifie comme un mécanisme convenu par le fournisseur de services et le consommateur de services pour communiquer ce qui est demandé et ce qui sera retourné.
- **La couche de Description de Services** est un schéma convenu pour décrire ce qu'est le service, comment il devrait être appelé, et quels sont les besoins requis permettant d'appeler le service avec succès.
- **La couche Service** décrit un service réel qui est rendu disponible pour l'usage.
- **La couche composition de service** fournit les mécanismes nécessaires pour assembler des services au sein d'une application.
- **L'Annuaire de services** est un entrepôt de descriptions de services et de données qui peuvent être employées par des fournisseurs de services permettant ainsi d'éditer leurs services, et par des consommateurs de services afin de découvrir ou trouver des services disponibles.

Les aspects de qualité de services incluent :

- **Une Politique de l'accord de service** (ou contrat de service), représente les termes de l'utilisation d'un service par un consommateur de services. Les termes spécifient la fonctionnalité fournie et attendue par un consommateur de services mais aussi les aspects non-fonctionnels de l'utilisation du service, comme, par exemple, un niveau de performances (temps de réponse, fiabilité) ;
- **La Sécurité**, qui est l'ensemble des règles pouvant être appliquées à l'identification, à l'autorisation, et au contrôle d'accès des consommateurs de services.
- **La Transaction** qui est l'ensemble d'attributs pouvant être appliqués à un groupe de services pour fournir un résultat cohérent. Par exemple, si un groupe de trois services doivent être employés pour accomplir une fonction précise, tous doivent s'exécuter.
- **La couche Gestion** est l'ensemble d'attributs qui pourraient être appliqués pour contrôler les services fournis ou consommés.[12]

# 4. Principes de base

Les Web Services reposent sur une architecture orientée services. Comme tout SOA, ils reposent donc sur les trois entités suivantes :

- les consommateurs de services ;
- les fournisseurs de services ;
- les annuaires de services.

Les Web services reposent sur l'articulation de trois standards XML que nous détaillerons davantage par la suite :

- SOAP, un protocole permettant d'invoquer à distance les opérations offertes par un Web Service en utilisant des messages XML,
- WSDL, un standard permettant de d´écrire l'interface d'un Web Service sous la forme d'un fichier de description en XML,
- UDDI, un protocole d'annuaire permettant à la fois de publier et de retrouver un Web Service.

# 5. Cycle de vie d'un web service

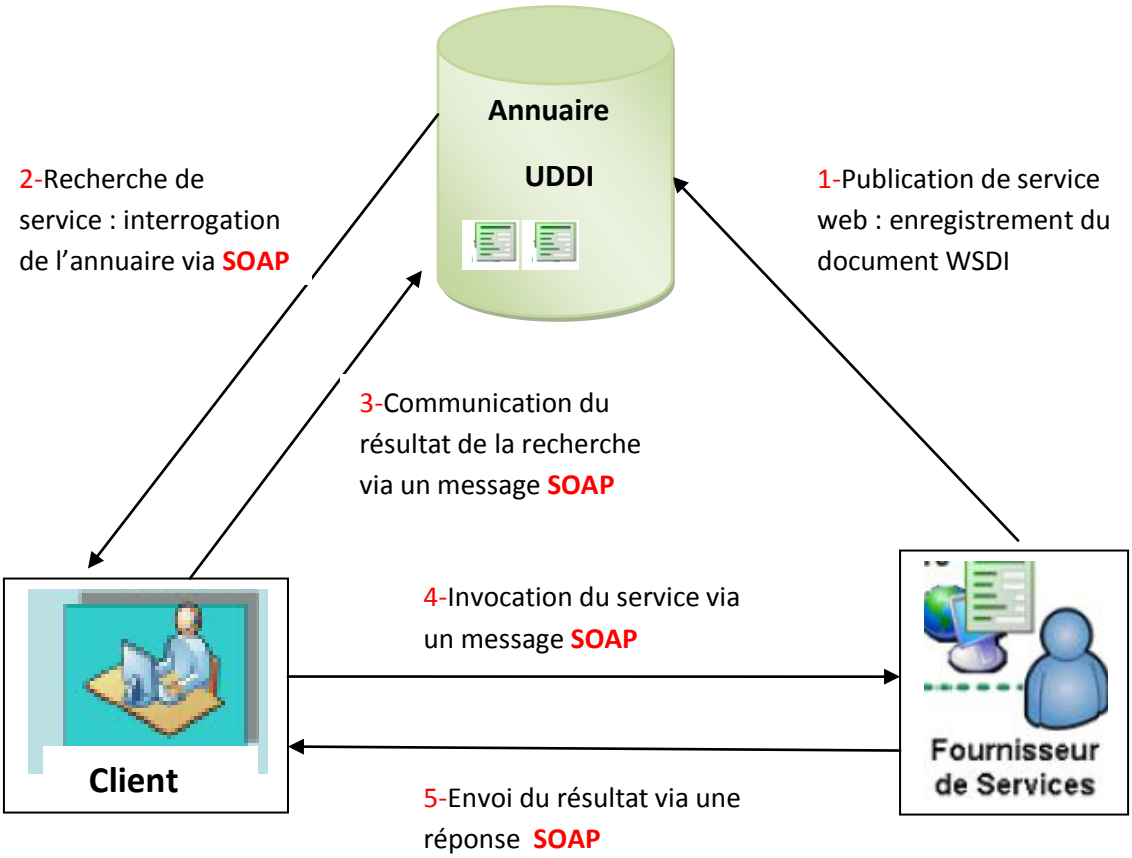



**Figure 5 Cycle de vie d'un service web**

Ce scénario se déroule en plusieurs étapes qui sont les suivantes :[9]

**1**. Le fournisseur définit la description de son service dans un document WSDL et la publie dans l'annuaire UDDI.

**2.** Le client, désirant trouver un service, interroge l'annuaire UDDI via un message SOAP.

**3.** L'annuaire retourne, via un message SOAP aussi, une liste de services qui répondent à la requête du client. Le client n'a qu'à choisir un parmi la liste.

**4.** Le client récupère le document WSDL du service choisi. Ensuite, il examine ce document afin de récupérer les informations nécessaires lui permettant de se connecter au fournisseur et d'interagir avec le service considéré. Enfin, il invoque l'opération désirée par le biais d'une requête SOAP renfermant les paramètres d'entrée de l'opération.

**5.** Le service, du coté du fournisseur, reçoit la requête, la traite, formule la réponse SOAP et l'envoie au client.

# 6. Standards utilisés pour le web service

## 6.1 SOAP (Simple Object Access Protocol)

Protocole spécifié par le W3C [97], il constitue une pièce maitresse dans l'architecture des web services car il assure la communication entre clients et services ou entre services par échange de messages au travers du Web. Il utilise principalement les protocoles HTTP(Hyper-Text Tranfer Protocol) et SMTP (Simple Mail Transfer Protocol) pour le transport de messages.

le protocole SOAP se base sur le standard XML pour encoder les données. Par conséquent, il profite des avantages de généricité, d'abstraction et de portabilité qu'offre ce standard pour la normalisation et la structuration des données.

Les messages SOAP sont englobés dans une enveloppe constituée d'un entête et d'un corps.

- L'enveloppe (obligatoire), contient le nom du message et l'espace de nom (namespace).
- L'entête (facultatif), apporte des données supplémentaires au message SOAP comme des informations concernant l'authentification, la gestion de transactions, le paiement,etc.
- Le corps (obligatoire) renferme, du côté client, l'opération du service invoquée ainsi que des valeurs des paramètres nécessaires à cette invocation, et du coté service, le résultat de l'exécution de l'opération invoquée.[9]

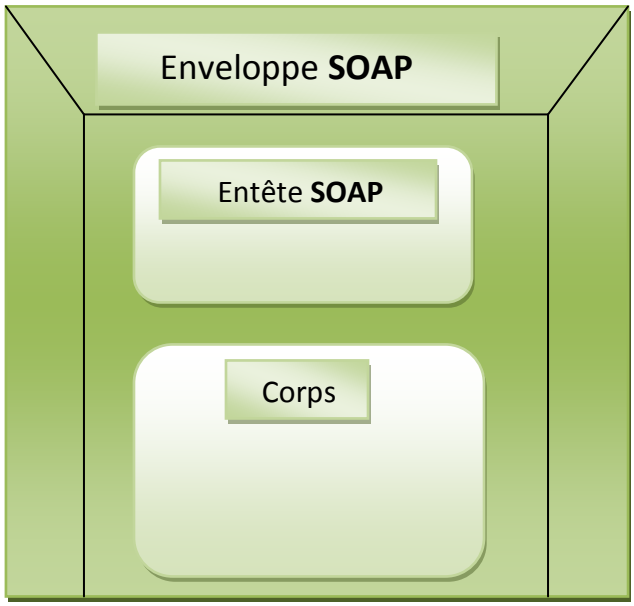



**Figure 6 structure d'un enveloppe SOAP**

## 6.2 WSDL (Web Service Description Language)

WSDL, acronyme de Web Services Description Language (langage de description des services Web) est un standard XML proposé par le W3C, utilisé pour décrire les Web Services et pour permettre aux clients de savoir comment accéder au service. Le fichier de description WSDL d'un Web Service contient un ensemble de définitions décrivant :

- l'interface du service c'est à dire les opérations que le service fournit, les formats des données et les protocoles utilisés,
- l'implémentation du service c'est `a dire l'adresse applicative (URL ou URI) pour accéder au service.[12]

Il permet de générer des documents structurés en deux parties : une partie abstraite (le Quoi) décrivant l'interface fonctionnelle du service en termes d'opérations et de messages, ainsi qu'une partie concrète (le Comment) qui contient les détails des protocoles à utiliser et de l'adresse physique des opérations. En particulier, un port type désigne une collection d'opérations, un binding consiste en une association entre un port type ainsi qu'un protocole de transport et de format de données, un port définit l'adresse physique d'un binding, et un service constitue une collection de ports.[3]

## 6.3 UDDI (Universal Description, Discovery and Integration)

UDDI est une spécification et un service de registre permettant de publier et de découvrir des services Web. Un registre UDDI est un registre basé sur XML qui contient des informations à propos d'entités d'affaire fournissant des services Web ainsi que des métadonnées concernant ces services (informations techniques ou légales). En outre, UDDI spécifie plusieurs API

(Application Programming Interface) pour interagir avec le registre, demander ou publier un service.[9]

Les données stockés dans l'UDDI sont structurées en XML et organisées en trois parties connues sous le nom de pages :

- Pages blanches : fournissent des descriptions générales sur les fournisseurs de services à savoir le nom de l'entreprise qui fournit le service, son identificateur commercial, ses adresses, etc.
- Pages jaunes : comportent des descriptions détaillées sur les fournisseurs de services catalogués dans les pages blanches de façon à classer les entreprises et les services par secteurs d'activités.
- Pages vertes : procurent des informations techniques sur les services Web catalogués.

Ces informations incluent la description du service, du processus de son utilisation et des protocoles utilisés pour son invocation.[3]

# 7. Objectifs des services web

Quatre objectifs fondamentaux sont visés par L'approche de service Web:

- **L'interopérabilité :** Elle permet à des applications écrites dans des

Langages de programmation différents et s'exécutant sur des plateformes différentes de communiquer entre elles. En manipulant différents standards que ce soit XML ou les protocoles d'Internet, les services Web garantissent un haut niveau d'interopérabilité des applications et ceci indépendamment des plateformes sur lesquelles elles sont déployées et des langages de programmation dans lesquels elles sont écrites. Ainsi, en s'appuyant sur un format d'échange de messages standard et sur l'ubiquité de l'infrastructure d'Internet, l'interopérabilité est donc une caractéristique intrinsèque aux services Web.

- **Le couplage faible :** Le couplage est une métrique indiquant le niveau d'interaction entre deux ou plusieurs composants logiciels. Nous parlons de couplage fort si les composants échangent beaucoup d'information et de couplage faible dans le cas contraire. Vu que la communication avec les services Web est réalisée via des messages décrits par le standard XML caractérisé par sa généricité et son haut niveau d'abstraction, les services Web permettent la coopération d'applications tout en garantissant un faible taux de couplage. Par conséquent, il est possible de modifier un service sans briser sa compatibilité avec les autres services composant l'application.

- **La réutilisation :** L'avantage de la réutilisation est qu'elle permet de réduire les coûts de développement en réutilisant des composants déjà existants. Dans le cas de l'approche

service Web, l'objectif de la séparation des opérations en services autonomes est en effet pour promouvoir leur réutilisation. Ainsi, lorsqu'un client définit ses exigences, il est généralement possible de réutiliser des services déjà existants pour satisfaire une partie des exigences. Ceci facilite la maintenance de l'application et permet un gain de temps considérable.

- **La découverte et la composition automatique :** La découverte et la composition sont des étapes importantes qui permettent la réutilisation des services. En effet, il faudra être en mesure de trouver et de composer un service afin de pouvoir en faire usage. En exploitant les technologies offertes par Internet et en utilisant un ensemble de standards pour la publication, la recherche et la composition, l'approche services Web tend à diminuer autant que possible l'intervention humaine en vue de permettre une découverte et une composition automatiques des services les plus complexes. En effet, pour réaliser son application, un développeur peut simplement interroger un moteur de recherche de services afin de trouver le service adéquat et à l'aide de langages de coordination appropriés il peut l'intégrer avec le reste des services de son application.[11][19]

# 8. Composition de services Web

La composition ou l'agrégation de services Web est une opération qui consiste à construire de nouvelles applications ou services appelés services composites ou agrégats par assemblage de services déjà existants nommés services basiques ou élémentaires.

La composition spécifie quels services doivent être invoqués, dans quel ordre et sous quelles pré-conditions.

Les services basiques peuvent être soient des services atomiques soient des services composites.

La composition de services Web vise essentiellement quatre objectifs :

- Créer de nouvelles fonctionnalités en combinant des services déjà existants.
- Résoudre des problèmes complexes auxquels aucune solution n'a été trouvée.
- Faire collaborer plusieurs entreprises ensemble.
- Optimiser et améliorer une fonctionnalité existante.[9]

### Types de composition

La composition des services Web peut être soit une composition statique soit une composition dynamique :

- **La composition statique** : est appelée aussi composition off-line, précompilée ou encore proactive. C'est une composition qui utilise des services basiques qui sont au

préalablement définis d'une façon figée et qui ne peuvent pas changer en fonction du contexte du client.[10]

Ce type d'application est celui qui est aujourd'hui le plus utilisé, en particulier pour les industriels. Il éxiste deux visions de la composition statique qui sont l'orchestration et la chorégraphie.

- ✓ **L'orchestration :** aborde le problème de façon centralisée, ou les collaborations de Service Web statiques sont contrôlées par le service composé, tel un chef d'orchestre qui se charge d'ordonner les appels aux services Web et de rattraper les erreurs.
- ✓ **La chorégraphie :** aborde le problème de façon distribuée, chaque partenaire d'une composition, i.e. chaque fournisseur de service Web, peut réaliser une ou plusieurs tâches, chacun d'eux communicants à l'aide de service Web.

Ce type de composition statique s'appuie sur des langages de composition de services Web tels que :

- XLANG (XML Busines Process Langage) de Microsoft ;
- BPML (Business Process Modeling Langage) de BPMI ;
- WSFL (Web Service Flow Langage) de IBM;
- WSCL( Web Service Conversation Langage) de Hewlett-Packard;
- WSCI (Web Service Choregraphy Interface) de SUN;
- BPEL4WS( Business Process Execution Langage for Web Services) de l'association de IBM, Microsoft et BEA, aussi appelé BPEL ou XSBPEL.

Ces langages décrivent les interactions entre différents fournisseurs de services Web et leurs clients. XLANG, BPML et BPEL sont associés à l'orchestration alors que WSCL et WSCI sont associés à la chorégraphie.[25]

Ce type de composition engendre des applications peu flexibles, parfois inappropriées avec les exigences des clients.

- **La composition dynamique** : appelée aussi composition on-line, postcompilée ou encore réactive. Elle se réfère à la sélection des services basiques « à la volée ».Autrement dit, la sélection des services basiques ne peut pas être prédéfinie à l'avance mais elle sera faite au moment de l'exécution en fonction des contraintes imposées par le client. Ceci permet d'élaborer différents scénarii de composition qui offrent les mêmes fonctionnalités et qui tiennent compte de la dynamique de la situation du client.[10]

Ce type de composition est encore très peu utilisé car il n'est pas encore assez sur, c'est-à-dire que l'obtention et la qualité du résultat ne sont pas garanties.

W3C propose une méthode de description des services Web visant à faciliter la composition des services web automatique. Cette description se base un langage appelé OWL-S

# 9. Service web sémantique

L'objectif premier du Web sémantique est de définir et lier les ressources du Web afin de simplifier leur utilisation, leur découverte, leur intégration et leur réutilisation dans le plus grand nombre d'applications. [32]

Le Web sémantique doit fournir l'accès à ces ressources par l'intermédiaire de descriptions sémantiques exploitables et compréhensibles par des machines. Cette description repose sur des ontologies.

À ce jour, le standard de description WSDL ne supporte pas la description de services Web comme une ressource utilisable dans le contexte du Web sémantique. Or, l'automatisation des processus d'enregistrement, de recherche et d'acquisition peut faciliter, à terme, la tâche des concepteurs de systèmes à base de services Web qui doivent faire face à l'augmentation du nombre de services Web disponibles.

Afin d'automatiser les processus d'enregistrement (action d'identification du service Web), derecherche (action issue de la requête du client) et de sélection (action de choix) des services Web, des travaux académiques ont été initiés, principalement dans le domaine du Web sémantique [36]. Les services Web issus des travaux de ce domaine sont appelés des **services Web sémantiques**.

les services Web sémantiques sont la combinaison de deux technologies celle des services Web et celle du Web sémantique.

Les services Web sémantiques sont des services Web dont la description est améliorée par des langages empruntés au Web sémantique , tel que RDF et OWL.

Cet emprunt au Web sémantique permet à ces services Web d'être découverts et sélectionnés automatiquement par des machines ou d'autres services Web distants. Ceci permet aux services Web sélectionnés de répondre au mieux à la requête du client.

## ➢ Langages de description de services Web sémantiques

Ils existent des travaux qui proposent une manière de représenter de manière sémantique des services Web (autrement dit proposent de décrire des services Web sémantiques). Malgré l'abondance de ce type de travaux, aucun ne s'est imposé comme une solution de description de services Web sémantiques. OWL-S est l'un des travaux issus du W3C qui tentent d'apporter une solution standard en termes de description de services Web sémantiques.[35]

- **OWL-S** [31]

OWL-S (Ontology Web Language for Services) est un langage issu des travaux de la DARPA et de son programme Agent Markup Language (DAML) et prend la suite de DAML-S (DARPA Agent Markup Language Service). Il a été intégré au consortium W3C en 2004, au sein du groupe d'intérêt sur les services Web sémantiques, lors de la recommandation du langage OWL.

Le but initial du langage OWL-S est de mettre en œuvre des services Web sémantiques. Cette mise en œuvre inclut un grand nombre d'objectifs, rendus possibles par le biais de l'expressivité héritée de OWL et de l'utilisation de la logique de description. Ces objectifs sont :

- **la description de services Web sémantiques** ;
- **l'invocation automatique de ces services**, par le biais de la détection et de l'interprétation automatique de la localisation et des paramètres d'entrée/sortie.
- **la composition automatique de** services (description et invocation) et la surveillance de l'exécution de la composition.

# 10. Conclusion

Les services Web sont à l'heure actuelle de plus en plus incontournables, Bien qu'ils soient maintenant capables d'échanger des données de façon quasi autonome à travers internet, ils sont pour l'instant encore incapables d'en comprendre le sens et ne peuvent par conséquent effectuer aucun raisonnement intelligent à partir de celles-ci.

Pour pallier à cette problématique, plusieurs travaux de recherche ont été menés autour de la description des services Web, qui utilisaient de plus en plus les ontologies pour fournir une représentation de l'information sémantique, à la fois, détaillée, riche et facile à manipuler par les machines. Les ontologies permettent d'améliorer la description et la découverte de services Web.

# Chapitre 2

# Les ontologies

# 1. Introduction

Face à l'émergence des sources d'informations disponibles de plus en plus nombreuses et complexes, il est nécessaire de permettre une description de ces informations non seulement en termes de structure (aspect syntaxique) mais également en termes de signification (aspect sémantique). La description de métadonnées sur les sources d'informations prenant en compte la description structurelle mais aussi sémantique des informations est un problème important. Cette sémantique peut être exprimée à l'aide d'ontologies.

Une ontologie permet la modélisation d'un domaine de connaissances et peut être vue comme un modèle conceptuel d'un domaine particulier, qui décrit les concepts de ce domaine et les relations entre ces concepts. Elle est généralement considérée comme une base de connaissances et est au centre des développements émergents du Web sémantique. Les ontologies permettent la modélisation d'informations agrées par une communauté de personnes et accessibles par une machine pour développer des services automatisés et par conséquent, jouent un rôle de référence pour décrire la sémantique des informations à partager.

L'utilisation d'ontologies permet la représentation formelle des connaissances à l'aide de modèles basés sur des logiques (logiques de représentation, logique de description) et le raisonnement sur ces connaissances à l'aide d'outils d'inférences (vérification de la cohérence des informations, classification des informations, etc.). Les connaissances modélisées dans une ontologie peuvent être partagées et/ou réutilisées dans différents environnements (applications) afférant à un même domaine d'intérêt.[27]

# 2. Définition d'ontologie

La définition des ontologies est héritée d'une tradition philosophique qui s'intéresse à la science de l'Etre. Aujourd'hui, elle signifie la « science des étants » c'est-à-dire l'ensemble des objets reconnus comme existants dans un domaine.

L'ontologie est utilisée, depuis plusieurs années, dans l'Ingénierie des Connaissances (IC) et l'Intelligence Artificielle (IA) pour structurer les concepts d'un domaine. Les concepts sont

rassemblés et ces derniers sont considérés comme des briques élémentaires permettant d'exprimer les connaissances du domaine qu'il recouvre.

Les ontologies sont utiles pour partager des connaissances, créer un consensus, construire des systèmes à base de connaissances. De nombreux projets d'ontologies sont en œuvre comme celui du Web sémantique. Le problème fondamental est de respecter la diversité des langages et des représentations du monde, tout en permettant les échanges d'informations.

- **définition issue de la philosophie**

Le terme **Ontologie** (avec un O majuscule) a tout d'abord été défini en Philosophie comme une branche de la Métaphysique qui s'intéresse à l'existence, à l'être en tant qu'être et aux catégories fondamentales de l'existant.
En effet, ce terme est construit à partir des racines grecques ontos (i.e. ce qui existe, l'Etre, l'existant), et logos (i.e. l'étude, le discours) d'où sa traduction par« l'étude de l'Etre » et par extension « de l'existence » [1]

le terme ontologie a été utilisé par les premiers étudiants d'Aristote pour désigner ce que Aristote appelait lui même : « philosophie première». Selon OED (oxford english dictionnary), la première apparition du terme « ontologie »en anglais était dans le dictionnaire de Bailey en 1721.[26]

- **définition issue de l'intelligence artificielle**

Au début des années 90, des chercheurs en Intelligence Artificielle se sont intéressés à cette notion pour la formalisation des connaissances. Dans cette discipline, ce qui « existe » peut être « représenté ». Dans ce contexte, ils ont défini une **ontologie** (avec un o minuscule) comme un artefact permettant de représenter l'existant par l'utilisation d'un vocabulaire formel et consensuel. Une des premières définitions de l'ontologie communément admise en Intelligence Artificielle a été énoncée par Gruber comme la « *spécification explicite d'une conceptualisation* ». Cette définition de l'ontologie a ensuite été affinée par R. Studer et al comme « *spécification formelle et explicite d'une conceptualisation partagée* » :

- ✓ *Formelle* : l'ontologie doit être lisible par une machine, ce qui exclut le langage naturel.
- ✓ *Explicite* : la définition explicite des concepts utilisés et des contraintes de leur utilisation.

- ✓ *Conceptualisation* : le modèle abstrait d'un phénomène du monde réel par identification des concepts clefs de ce phénomène.
- ✓ *Partagée* : l'ontologie n'est pas la propriété d'un individu, mais elle représente un consensus accepté par une communauté d'utilisateurs.[1]

# 3. Composantes d'une ontologie

Comme tout formalisme de représentation, les ontologies sont basées sur l'utilisation d'un certain nombre de composantes (dites aussi briques ou constituants) de base, véhiculant avec eux les connaissances traduites par ces dernières et qui sont principalement : Concept, Relation, Fonction, Axiomes, Instance.

- **Les concepts :** aussi appelés termes ou classes de l'ontologie, constituent les

objets de base manipulés par les les ontologies. Ils correspondent aux abstractions pertinentes du domaine du problème, retenues en fonction des objectifs qu'on se donne et de l'application envisagée pour l'ontologie, par exemple, la description d'un ensemble d'objets, d'une tâche, d'une fonction, d'une stratégie, d'un processus de raisonnement, etc.

- **Les relations :** traduisent les interactions existant entre les concepts présents

dans le domaine ciblé. Ces relations sont formellement définies comme tout sous ensemble d'un produit cartésien de n ensembles, c'est à dire R : C1 x C2x … x Cn et incluent 1) la relation de spécialisation (subsomption), 2) la relation de composition (méronymie), 3) la relation d'instanciation, etc. Ces relations nous permettent de capturer, la structuration ainsi que l'interaction entre les concepts, ce qui permet de représenter une grande partie de la sémantique de l'ontologie**.**

- **Les fonctions** : sont des cas particuliers de relations dans lesquelles le nième

élément (extrant) de la relation est défini de manière unique à partir des n-1 éléments précédents (intrants). Formellement, les fonctions sont définies ainsi : F : C1 x C2 … x Cn-1 → Cn. Comme exemple de fonctions binaires, nous pouvons citer la fonction *mère-de.*

- **Les axiomes :** permettent de modéliser des assertions toujours vraies, à propos

des abstractions du domaine traduites par l'ontologie. Ils permettent de combiner des concepts, des relations et des fonctions pour définir des règles d'inférences et qui peuvent

intervenir, par exemple, dans la déduction, la définition des concepts et des relations, ou alors pour restreindre les valeurs des propriétés ou les arguments d'une relation.

- **Les entitées :** ou *individus* constituent la définition extensionnelle de

l'ontologie. Ils représentent des éléments singuliers véhiculant les connaissances (statiques, factuelles) à propos du domaine du problème.[20]

# 4. types d'ontologie

Plusieurs classifications des ontologies ont été proposées dans la littérature, Nous présentons dans la suite une classification basée sur le niveau de granularité que GUARINO et autres [16], définissent en quatre classes.

- **Les ontologies supérieures** (Upper or Top-level Ontologies) [17]: ont pour objet l'étude des catégories de choses qui existent dans le monde, comme les concepts de haute abstraction tels que: les entités, les événements, les états, les processus, les actions, le temps, l'espace, les relations, les propriétés, etc. et qui sont indépendants d'un domaine particulier.

Les exemples d'ontologies de haut niveau sont :[28]
SOWA , CYC , et SUO10 .

- **Les ontologies de domaine (Domain ontologies**) [24][18] : ce sont des ontologies qui sont construites sur un domaine particulier de la connaissance. Elles fournissent le vocabulaire des concepts du domaine de connaissance, ainsi que les théories et les principes de base de ce domaine. Les ontologies de domaine constituent donc des méta-descriptions d'une représentation de connaissances du domaine.

Il y a une frontière claire entre les ontologies de domaine et les ontologies de haut niveau. Les concepts dans les ontologies de domaine sont habituellement des spécialisations des concepts déjà définis dans les ontologies de haut niveau, et le même principe pourrait se produire avec les relations. De nombreuses ontologies de domaine existent déjà, telles que MENELAS dans le domaine médical, ENGMATH pour les mathématiques, TOVE dans le domaine de la gestion des entreprises , etc.

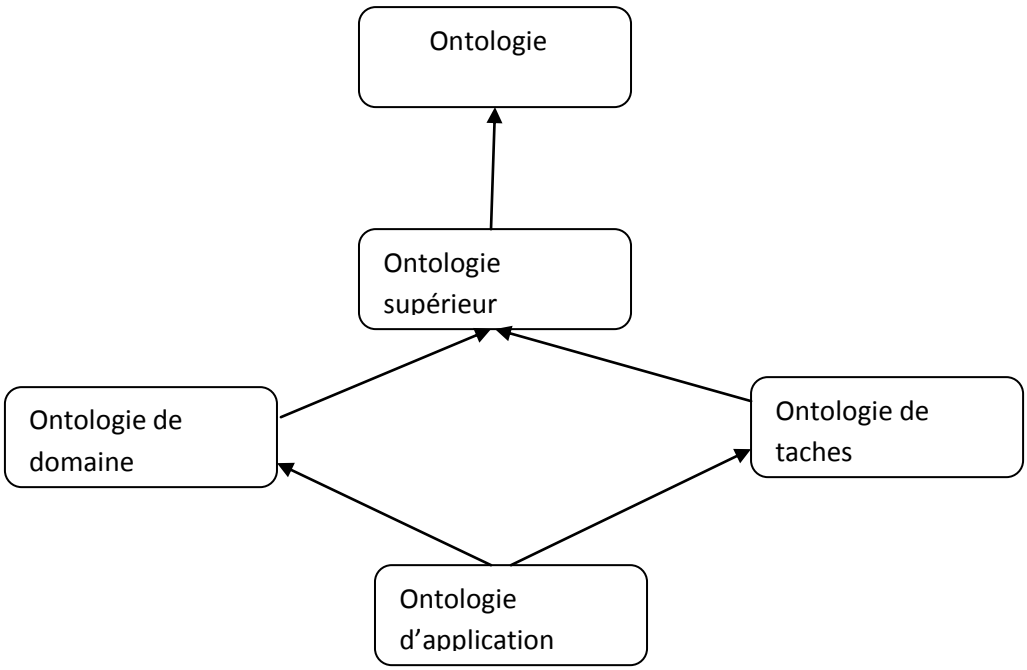



**Figure 7 Classification des ontologies selon Guarino**

- **Les ontologies de tâches (Task ontologies)** [24] : Ces ontologies sont utilisées pour gérer des taches spécifiques liées à la résolution de problèmes dans les systèmes, et par suite le vocabulaire qu'elles décrivent est relié à une activité ou à une tache générique telles que les tâches de diagnostic, de planification, de configuration, etc et cela en spécialisant les termes dans les ontologies de haut niveau. Les ontologies de tâche fournissent un vocabulaire systématique des termes utilisés pour résoudre les problèmes liés aux tâches

- **Les ontologies d'application** [18] : ce sont les ontologies les plus spécifiques. Elles permettent de décrire des concepts dépendants à la fois d'un domaine et d'une tâche. Dans cette classification, la notion d'ontologie d'application définit le *contexte* d'une application qui décrit la sémantique des informations et des services manipulés par une ou un ensemble d'applications sur un même domaine.[16]

# 5. Cycle de vie d'ontologies

Les ontologies étant destinées à être utilisées comme des composants logiciels dans des systèmes répondant à des objectifs opérationnels différents. Leurs développement doit s'appuyer sur les mêmes principes que ceux appliqués en génie logiciel [13].

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 un certain nombre d'activités transversales de support (évaluation, documentation, gestion de la configuration).[5]

Actuellement nous pouvons recenser dans la littérature une multitude de méthodologies, un peu plus d'une trentaine selon [14]

Un cycle de vie inspiré du génie logiciel est proposé dans (Dieng et al. 2001). Il comprend une étape initiale d'évaluation des besoins, une étape de construction, une étape de diffusion, et une étape d'utilisation. Après chaque utilisation significative, l'ontologie et les besoins sont réévalués et l'ontologie peut être étendue et, si nécessaire, en partie reconstruite.[7]

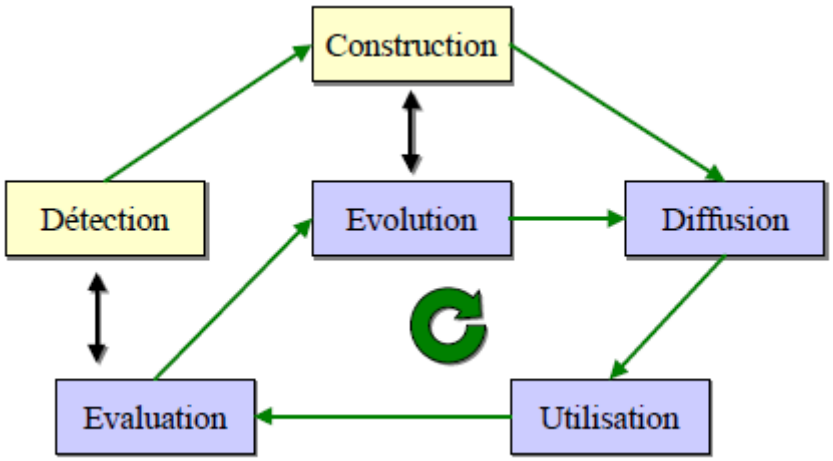



**Figure 8 Cycle de vie d'une ontologie**

# 6. Processus de construction d'une ontologie

Le processus de construction d'ontologies repose sur un enchaînement de trois étapes (conceptualisation, ontologisation, opérationnalisation) permettant de passer des données brutes à l'ontologie opérationnelle. Les données brutes, constituant un corpus (exprimé a priori en langage naturel), intègrent toutes les connaissances du domaine que l'on souhaite formaliser.

- **Conceptualisation**

Cette étape consiste à identifier des connaissances contenues dans un corpus représentatif du domaine. Ce travail doit être mené par un expert du domaine, assisté par un ingénieur de la

connaissance. Cette étape permet d'aboutir à un modèle informel, sémantiquement ambiguë et généralement exprimé en langage naturel.

- **Ontologisation**

C'est une formalisation, autant que possible, sans perte d'information, du modèle conceptuel obtenu à l'étape précédente. Ce travail doit être mené par l'ingénieur de la connaissance, assisté de l'expert du domaine. Cette étape Facilite sa représentation ultérieure dans un langage complètement formel et opérationnel . L'ontologisation peut être complétée par une étape d'intégration au cours de laquelle une ou plusieurs ontologies vont être importées dans l'ontologie à construire [16].

- **Opérationnalisation**

C'est une transcription de l'ontologie dans un langage formel (i.e. possédant une syntaxe et une sémantique) et opérationnel (i.e. doté de services inférentiels permettant de mettre en œuvre des raisonnements) de représentation de connaissances. Ce travail doit être mené par l'ingénieur de la connaissance.

# 7. Langages de description d'ontologie

- **SHOE** ("Simple HTML Ontology Extension"), basé sur du web il combine les "frames" et les règles. Il a été construit comme une extension de HTML en 1996. Il utilisait des étiquettes différentes de celles des spécifications HTML, de ce fait permettant l'insertion des ontologies dans des documents HTML. Plus tard sa syntaxe a été adaptée à XML.
- **XOL** ("XML-based Ontology exchange Language") développé comme une transformation en XML d'un petit sous ensemble des primitives du protocole OKCB, appelé OKBC-Lite.[21]
- **RDF** Le besoin d'un modèle conceptuel devient nécessaire pour la description de chaque ressource.D'où l'introduction de RDF pour *Resource Description Framework* qui est un modèle conceptuel, abstrait et formel, fondé sur un modèle de graphe de ressources, permettant de décrire les éléments simplement et sans ambiguïté selon un mécanisme basé sur des déclarations RDF. [37]

Une déclaration RDF est une phrase composée d'un triplet <Sujet, Prédicat, Objet> qui peut être traitée par la machine pour permettre à celle-ci de le faire tout en comprenant la signification de ce triplet. Chaque ressource, et plus précisément, chaque sujet du triplet est identifié par un URI (*Uniform Resource identifier*). Cette identification se fait de manière

unique à l'aide d'un nom sans avoir à localiser la ressource (un bon exemple d'URI est l'URL). Le prédicat exprime la propriété, comme par exemple « est créateur » dans le triplet « Monsieur X – est créateur – du Web Sémantique » où Monsieur X est le sujet et enfin « Web Sémantique » est l'objet.

Par exemple, le fait que le document van-gogh.xml « parle de » peintures à l'huile sur toile de la période néo-impressionniste peut être représenté par le graphe suivant fondé sur une ontologie provenant du domaine de l'art :[36]

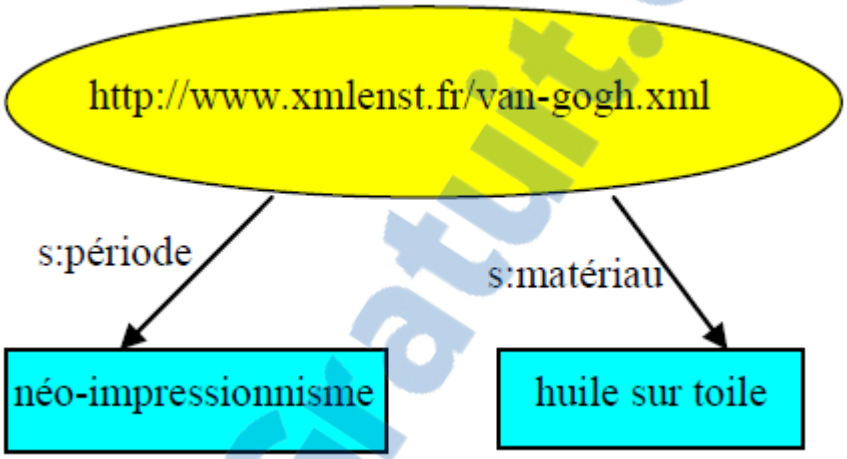



**Figure 9 Exemple d'un graphe de ressource RDF**

La recommandation RDF propose l'utilisation de XML pour la représentation des graphes de ressources. Voici une représentation XML du graphe de la Figure 9 :

```
<?xml version="1.0" encoding="iso-8859-1"?>
<rdf:RDF
        xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns:s="http://schemas.xmlenst.fr/peintres_rdf/">
<rdf:Description about="http://www.xmlenst.fr/van-gogh.xml">
        <s:période>néo-impressionnisme</s:période>
        <s:matériau>huile sur toile</s:matériau>
</rdf:Description>
</rdf:RDF>
```

**Figure 10 Exemple d'une représentation XML d'un graphe de ressource**

Pour utiliser les termes d'une ontologie, il faut pouvoir indiquer les différents vocabulaires utilisés. Ainsi, au début de chaque ontologie, on intègre un ensemble de déclarations d'espaces de noms dans une balise *<rdf:RDF>*. Ces déclarations permettent l'interprétation sans ambiguïté des URIs et la lecture facile du reste de l'ontologie. On y trouve l'identification de l'espace de noms implicite de l'ontologie, l'identification de l'espace de noms relative aux préfixes utilisés. Enfin, on y trouve les différentes descriptions RDF ou, comme on le verra dans les sections suivantes, des éléments RDFS ou OWL.

Ainsi, RDF permet de définir aisément une ontologie, son inconvénient est qu'il ne supporte pas la vérification de la cohérence des données (vérification que le champ « date de naissance» est vraiment une date par exemple).

- **RDF Schéma**

Une évolution de RDF est introduite dans RDFS (pour RDF Schema) . Ce langage est simple et permet l'implémentation du modèle RDF pour la définition des ontologies avec une approche cette fois-ci orientée objet. Les trois notions principales permettant cela sont: la ressource(*rdfs:Resource*), la classe (*rdfs:Class*) et la propriété. On a avec RDFS l'avantage de pouvoir créer une hiérarchie de classes et de propriétés, comme dans les langages orientés objet, grâce aux notions de *subClassOf* et *subPropertyOf*. On peut bien entendu instancier une classe à l'aide de *rdf:type*. L'autre avantage vient du fait que RDFS restreint le domaine d'application des propriétés (avec un domaine qui précise la classe du sujet du triplet utilisant la propriété et un intervalle précisant la classe de l'objet du triplet utilisant la propriété).

```
<?xml version="1.0" encoding="iso-8859-1"?>
<rdf:RDF
        xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
    <rdfs:Class rdf:ID="Objet_Iconographique"/>
    <rdfs:Class rdf:ID="Peinture">
        <rdfs:subclassOf rdf:resource="#Objet_Iconographique"/>
    </rdfs:Class>
    <rdfs:Class rdf:ID="Période"/>
    <rdf:Property rdf:ID="date">
        <rdfs:domain rdf:resource="#Objet_Iconographique"/>
        <rdfs:range rdf:resource="Période"/>
    </rdf:Property>
    <rdf:Property rdf:ID="matériau">
        <rdfs:domain rdf:resource="Peinture"/>
        <rdfs:range rdf:resource="rdfs:Literal"/>
    </rdf:Property>
    <Peinture about="van-gogh.xml">
        <date>
            <Période id="#néo-impressionnisme"/>
        </date>
        <matériau>huile sur toile</matériau>
    </Peinture>
</rdf:RDF>
```

**Figure 11 exemple de RDF Schéma**

Dans ce document, l'ontologie est composée de trois concepts ou classes (Literal, Peinture,Période) et de deux propriétés (matériau et date). Peinture est une description RDF, elle est appelée la ressource sujet des deux propriétés qui ont comme ressources objets, respectivement, Matériau et Période.

L'intérêt de définir un schéma RDFS n'est pas seulement de pouvoir contrôler la terminologie et la structure des descriptions RDF, mais également d'introduire la possibilité de raisonner sur les liens *isA* (est-un) qui existent entre les concepts et les propriétés. Ceci est surtout utile quand on veut interroger des métadonnées pour découvrir des ressources. Voici un exemple de requête RQL, qui cherche toutes les ressources sur des objets iconographiques de la période néo-impressionniste : [39]

```
select X from Objet_Iconographique{X}.Période.{Y} where Y = "#néo-impressionnisme"
```

Cette requête prend en compte le fait que toutes les instances de la classe Peinture sont également des instances de la classe *Objet_Iconographique* (à travers la propriété rdfs:subclassOf) et renvoie la ressource *van-gogh.xml*.
Un inconvénient majeur est cependant à noter: il est difficile de définir dynamiquement une nouvelle classe à partir d'une classe donnée avec des restrictions supplémentaires (Exemple d'une telle classe: Un enfant est une personne dont l'age est inférieur à 18 ans).[6][38][32]
Les langages ci-dessus ont établi les bases du Web Sémantique. Et c'est dans ce contexte que trois autres langages ont été développés comme des extensions de RDF(S). Il s'agit de :

- **OIL** ("Ontology Interchange Language and Ontology Inference Layer"), développé en début 2000 dans le cadre du projet européen On-To-Knowledge . Il ajoute des primitives de RC basées sur des "frames" à RDF(S) et sa sémantique formelle est basée sur les logiques de description.
- **DAML+OIL** ("DARPA Agent Markup Language") , créé plus tard entre 2000 et 2001 par un comité mixte des USA et de l'Union Européenne dans le contexte du projet DAML sur la spécification précédente de DALM-ONT, qui a été construit en fin 2000, et sur OIL. DAML+OIL ajoute des primitives de RC basées sur les logiques de description à RDF(S).[21]
- **OWL** OWL, recommandé par le W3C en février 2004, est le plus expressif des langages ontologiques pour le Web . La conception d'OWL a bénéficié de plusieurs générations de langages de représentation des connaissances, d'une base théorique solide en logique et d'une volonté de la part de ses concepteurs pour créer un langage approprié à une utilisation dans le cadre du Web Sémantique. En fait, OWL est issu des travaux autour du langage DAML+OIL, lui-même fusion de deux projets l'un européen, OIL, et l'autre américain, DAML. La plupart des chercheurs ayant participé à l'élaboration du langage DAML+OIL ont ensuite travaillé à OWL.

Le langage OWL fournit des mécanismes pour créer tous les composants d'une ontologie : classes, instances, propriétés et axiomes. OWL repose également sur la syntaxe des triplets

RDF et réutilise certaines des constructions RDFS. Comme en RDFS, les classes peuvent avoir des sous-classes, fournissant ainsi un mécanisme pour le raisonnement et l'héritage des propriétés. Par contre, en OWL, on distingue :

- ✓ les propriétés objet (object property), i.e. les relations, qui relient des instances de classes à d'autres instances de classes. C'est l'équivalent des triplets RDF dont l'objet est une ressource.
- ✓ les propriétés type de données (datatype property), i.e. les attributs, qui relient des instances de classes à des valeurs de types de données (nombres, chaînes de caractères,…). C'est l'équivalent des triplets RDF dont l'objet est une valeur littérale.

Les axiomes fournissent de l'information au sujet des classes et des propriétés, spécifiant par exemple l'équivalence entre deux classes.OWL se compose de trois sous-langages : OWL Lite, OWL DL et OWL Full.[1] [2]

# 8. Conclusion

Les ontologies sont des descriptions formelles d'un domaine particulier, elles ont un rôle primordial dans l'interopérabilité et l'intégration des applications distribuées. Elles facilitent aussi la médiation entre les sources de données hétérogènes. Dans notre cas, nous allons les exploiter (concepts, relations, subsomption…) à fin d'attribuer à nos services web une sémantique appropriée.

# Chapitre 3

## Conception et Réalisation

# 1. Introduction

Le travail réalisé dans le cadre de notre projet vise à développer une application logicielle qui sert à renforcer la collaboration entre les différents services de notre hôpital CHU (spécialement avec le service médecine de travail) et cela en faisant communiquer leurs différents médecins appropriés, de manière dynamique et flexible, tout en se basant sur une architecture distribuée

Dans cette partie, nous allons présenter les outils utilisés ainsi que toutes les étapes de conception de notre application allant de la modélisation UML jusqu'à l'implémentation du système.

# 2. Conception

La conception de notre application s'est avéré une tâche complexe d'où la nécessité de suivre une démarche d'analyse. Le langage de modélisation que nous avons choisi est *Unified Modeling Language* (UML). La fonction d'UML consiste à spécifier, visualiser, construire et documenter un système informatique.

## 2.1 Diagramme de cas d'utilisation

Ces diagrammes regroupent les différents acteurs ainsi que les cas d'utilisation. Ils décrivent aussi, sous forme d'actions et de réactions, le comportement d'un système du point de vue d'un utilisateur qui peut être soit un médecin généraliste ou spécialiste

- Le médecin généraliste chargé de suivre le patient qui se présente au service médecine de travail ;
- Le médecin spécialiste consulte les rapports d'orientation que le médecin généraliste a déjà crée, en vue d'un suivi bien approfondi.

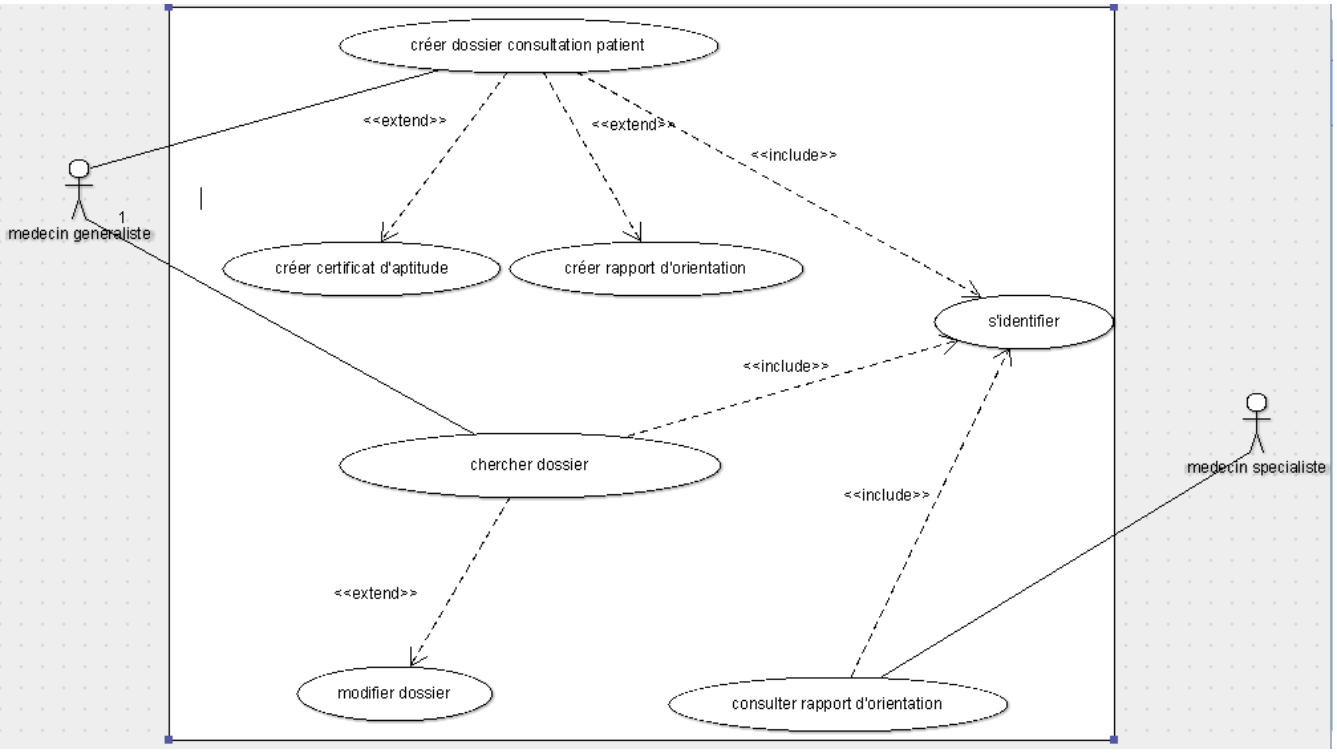



**Figure 12 diagramme de cas d'utilisation**

## 1.1. Diagramme de classes

Les diagrammes de classes expriment de manière générale la structure statique d'un système, en termes de classes et de relations entre ces classes [20].

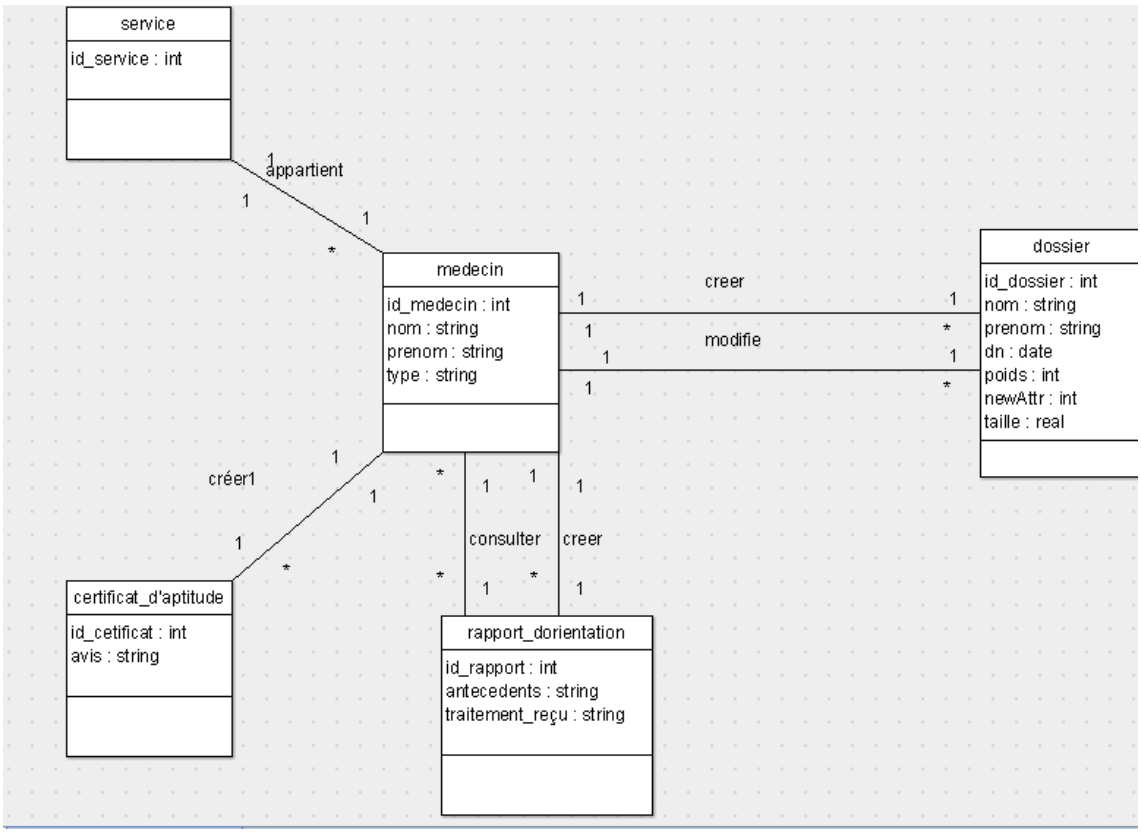
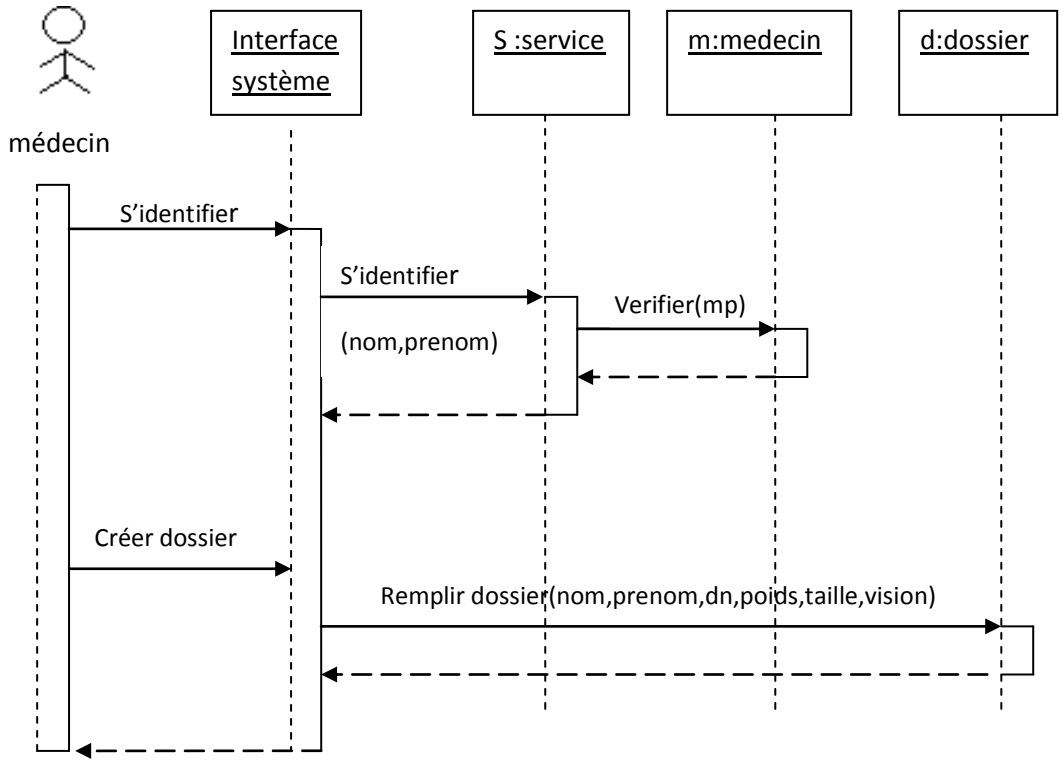



**Figure 13 diagramme de classes**

## 1.2. Diagramme de séquence

Les diagrammes de séquences montrent les interactions existantes entre les différents objets d'un cas d'utilisation, selon un point de vue temporel.

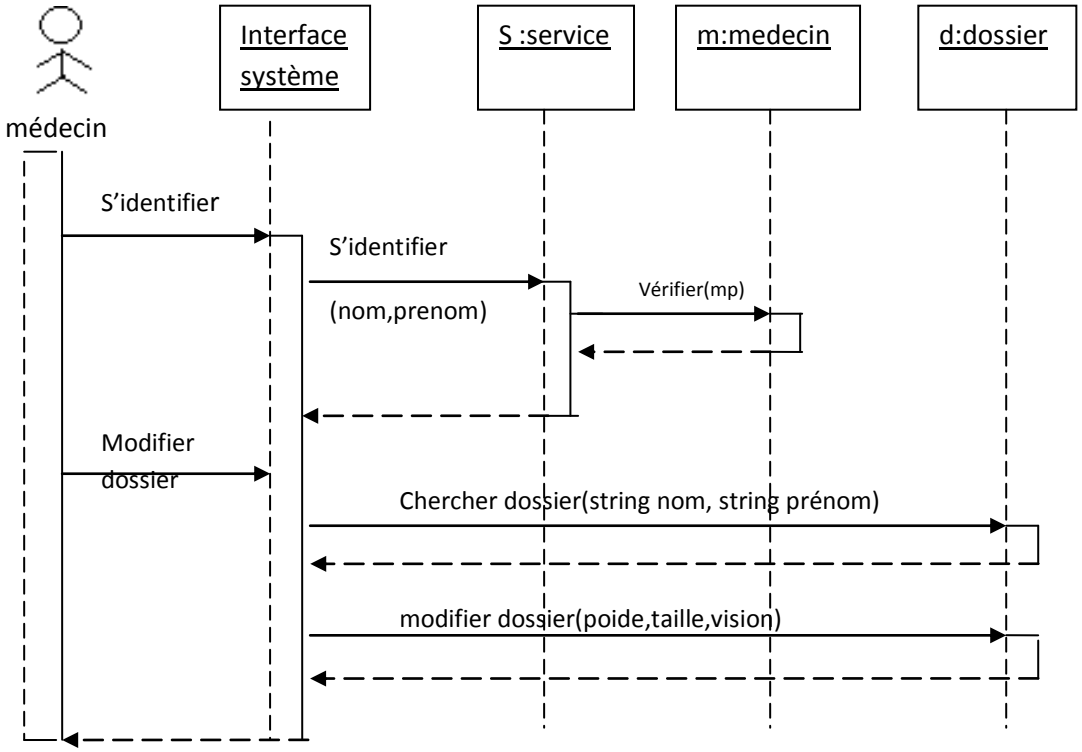
- **Cas d'utilisation : Créer dossier**




Après l'introduction du login et du mot de passe par le médecin, notre système authentifie l'utilisateur et affiche le menu principal qui lui est dédié .

Quand l'utilisateur choisit l'option « créer dossier», le système lui affiche un formulaire à remplir, Le système traite les requêtes de l'utilisateur, et créer le dossier selon les champs remplis

- **Cas d'utilisation : Modifier dossier**




# 3. Outils et environnement de développement

Avant de commencer l'implémentation de notre application, nommée e-QSR, nous allons tout d'abord spécifier les outils utilisés qui nous ont semblé être un bon choix vu les avantages qu'ils offrent.

## 3.1 Langage Java

Notre choix du langage de programmation s'est porté sur le langage JAVA pour ces nombreuses richesses. Notant que Java est un langage de programmation orienté objet, libre, simple et portable. La version du JDK que nous avons utilisé est 6.8.

## 3.2 JAVA EE6

Java Enterprise Edition est apparue à la fin des années 1990 et a apporté au langage Java une plateforme logicielle robuste pour les applications d'entreprise. Remise en cause à chaque nouvelle version, mal comprise ou mal utilisée, concurrencée par les frameworks Open Source, elle a su tirer profit de ces critiques pour s'améliorer et trouver un équilibre dans sa version Java EE 6[15]

## 3.3 JAX-WS

JAX-WS est l'acronyme *Java API for XML Web Services*, JAX-WS est à la fois un standard et une implémentation. La version courante et celle que nous avons utilisé est JAX-WS 2.0, précédemment JAX-WS s'appelait JAX-RPC [4].

Cette API permet facilement l'appel de méthodes distantes et la réception de leur réponse en utilisant SOAP et HTTP. Cette facilité permet de s'affranchir d'une mise en œuvre détaillée de SOAP pour réaliser des opérations [8]

## 3.4 PROTEGE

Nous avons préféré écrire l'ontologie de domaine avec Protégé version 4.2.0 ,qui est un système conçu pour la création d'ontologies, très populaire dans le domaine du Web sémantique .Protégé est développé en Java. Il est gratuit et à code source libre[40].

## 3.5 TOMCAT

Tomcat ne constitue qu'un conteneur web, il peut être également parfois désigné comme moteur de servlet, ou plus abusivement comme un serveur Web. Tomcat est en réalité souvent employé en combinaison avec un serveur Web Apache ou d'autres serveurs Web (JBoss, IIS, Web Sphère, etc)[41].

## 3.6 MY SQL

MySQL est un système de gestion de base de données (SGBD). Il est distribué sous une double licence GPL et propriétaire. Il fait partie des logiciels de gestion de base de données les plus utilisés au monde, autant par le grand public (applications web principalement) que par des professionnels, en concurrence avec Oracle, Informix et Microsoft SQL Server.[42]

## 3.7 Eclipse

Pour le choix de l'environnement de développement nous avons opté pour Eclipse Java EE IDE for Web Developers.Version: Helios Service Release 2.Build id: 20110218-0911 version 3.6 . Eclipse est un projet de la Fondation Eclipse visant à développer tout un environnement de développement libre, extensible, universel et polyvalent.

# 4. Développements des services web

La majorité des langages de programmation orientés Web supportent le développement de services Web dont Java, PHP, C#, C++, Python…

Il existe différents frameworks de développement de Services Web avec le langage Java:

- JAX-WS Specification Sun (jax-ws.dev.java.net)
- AXIS 1 et 2 Apache (ws.apache.org/axis et ws.apache.org/axis2)
- CXF Apache (cxf.apache.org)
- XFire Codehaus (xfire.codehaus.org)
- JBossWS JBoss (www.jboss.org/jbossws)
- …

## 4.1 Serveur d'application

Chaque service Web est déployé dans une application Web grâce à un serveur d'application. Il existe différentes catégories de serveur d'application pour gérer les Web Services dont :

➢ **Glassfish** 

La gestion du Web Service est transparente et maintenue par le serveur d'application.

➢ **Tomcat** 

Nécessite une configuration explicite du Web Service

## 4.2 Exemple de création de service web

Pour créer un service web sous eclipse on procède de la façon suivante :

- Créer un nouveau projet.
- Créer visual class java qui constituera le futur service web(ex :créer_dossier), et qui a

comme paramètre les entrées (inputs du service web, et qui constituent les champs que l'utilisateur doit remplir à fin d'invoquer le service).

```
 public void creerdossierM(String username,String mp,String nom, String
prenom, String dn, String poste_de_travail, int poids, double taille, int
vision, int audition, int respiration, int digestion, String
date_consultation )
```

- Se connecter à la base de données

```
String connectionURL = "jdbc:mysql://localhost:3306/medecine";
```

- Créer la requête chargé d'insérer les champs dans la base de données

```
String QueryString1 ="INSERT INTO dossier(id_medecin, nom, prenom,dn,
poste_de_travail,poids, taille, vision, audition, respiration, digestion,
date_consultation) VALUES(2,'"+nom+"'
,'"+prenom+"','"+dn+"','"+poste_de_travail+"', '"+poids+"','
```

```
"+taille+"','"+vision+"','"+audition+"'
,'"+respiration+"','"+digestion+"','"+date_consultation+"') ";
```

- Créer le service web et le déployer en cliquant bouton droit sur la méthode Créer_dossier

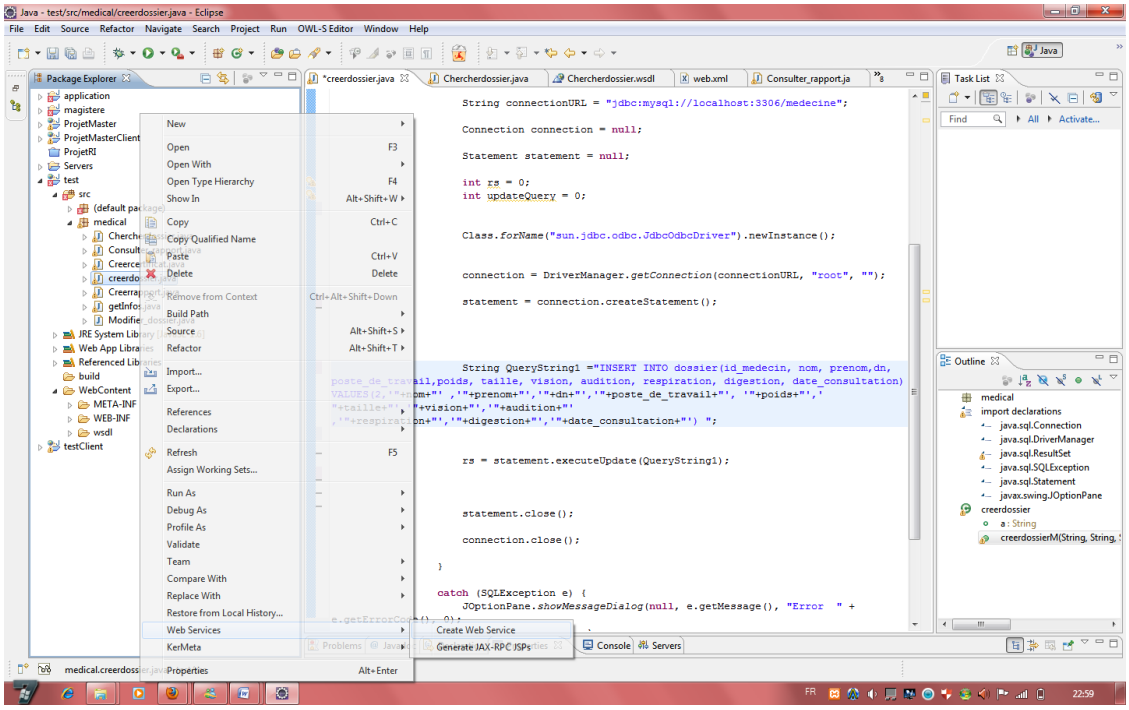



**Figure 14 créer un service web sous eclipse**

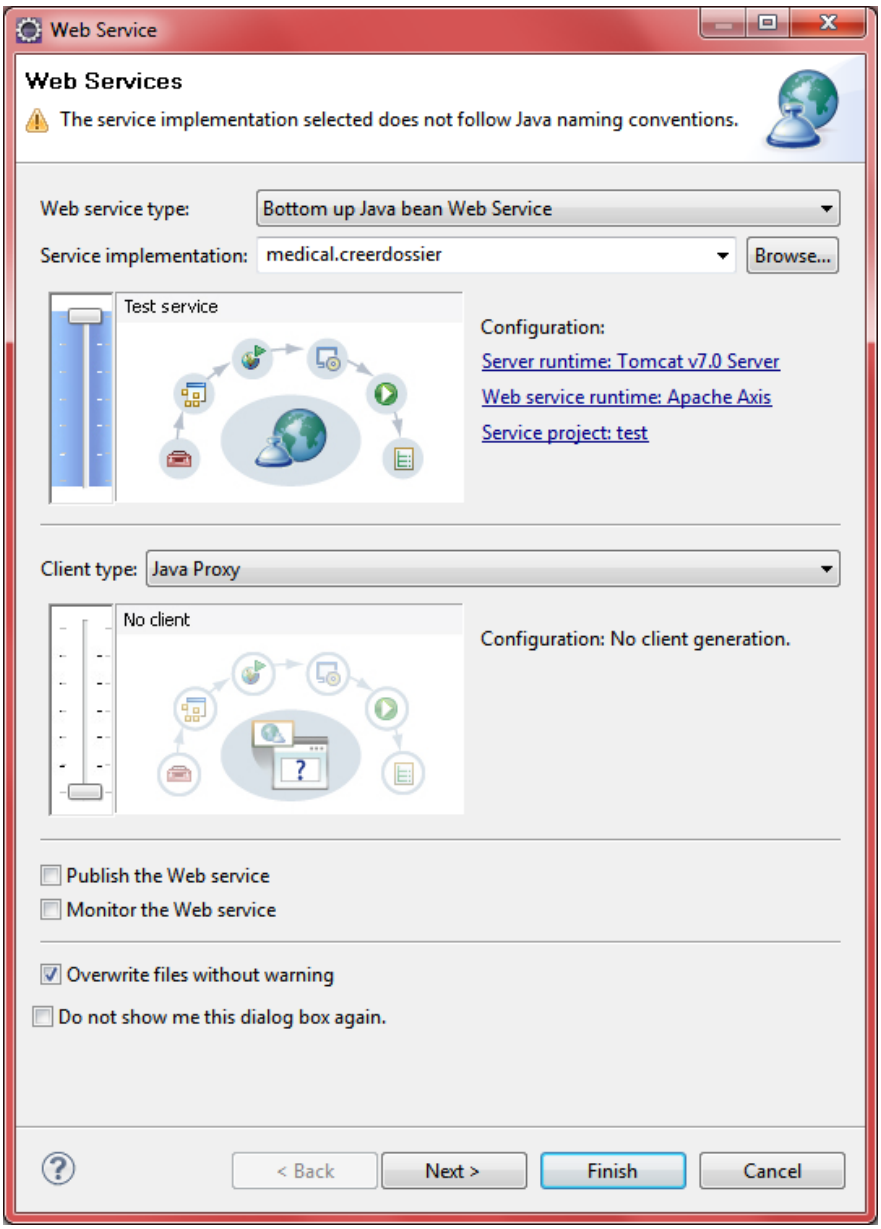



**Figure 15 déployer le service web**

- En cliquant sur next le fichier WSDL associé au service web se génère

Automatiquement

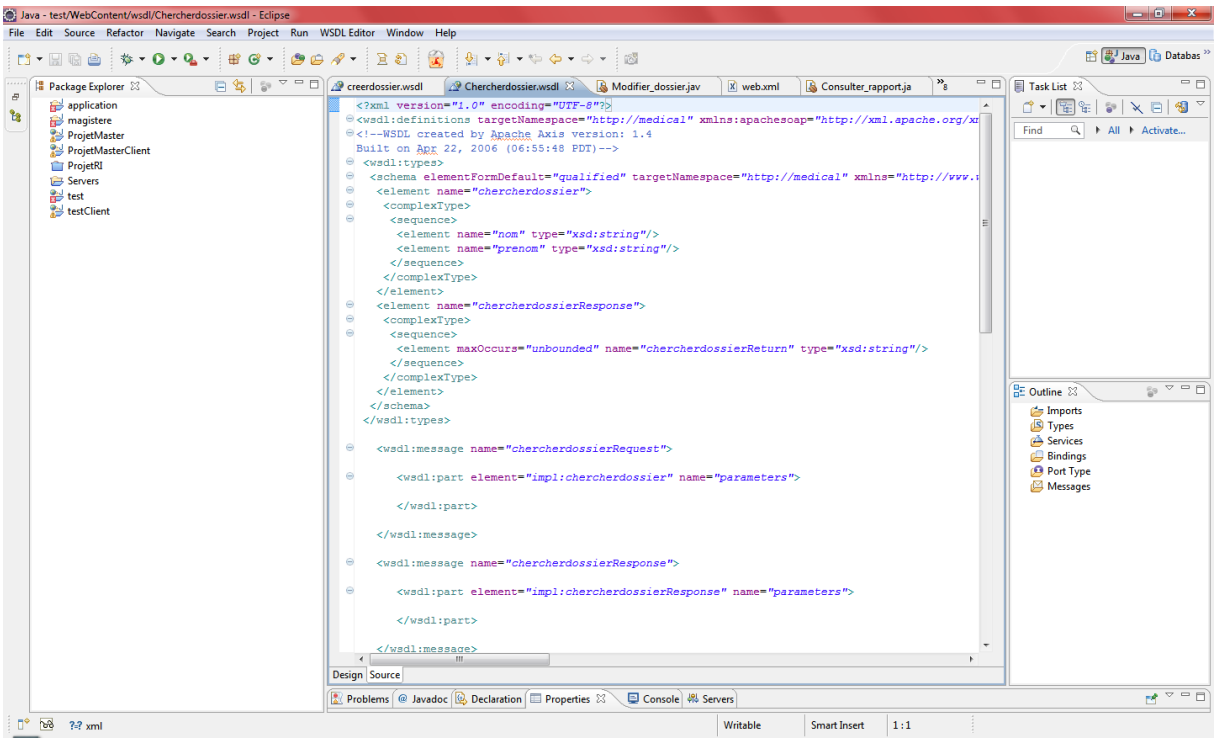



**Figure 16 le fichier wsdl associé au service web crée**

# 5. L'ontologie et l'annotation des services web

Ce graphe représente l'ontologie du domaine (crée avec protégé 4.2.0) que nous avons créé pour notre application

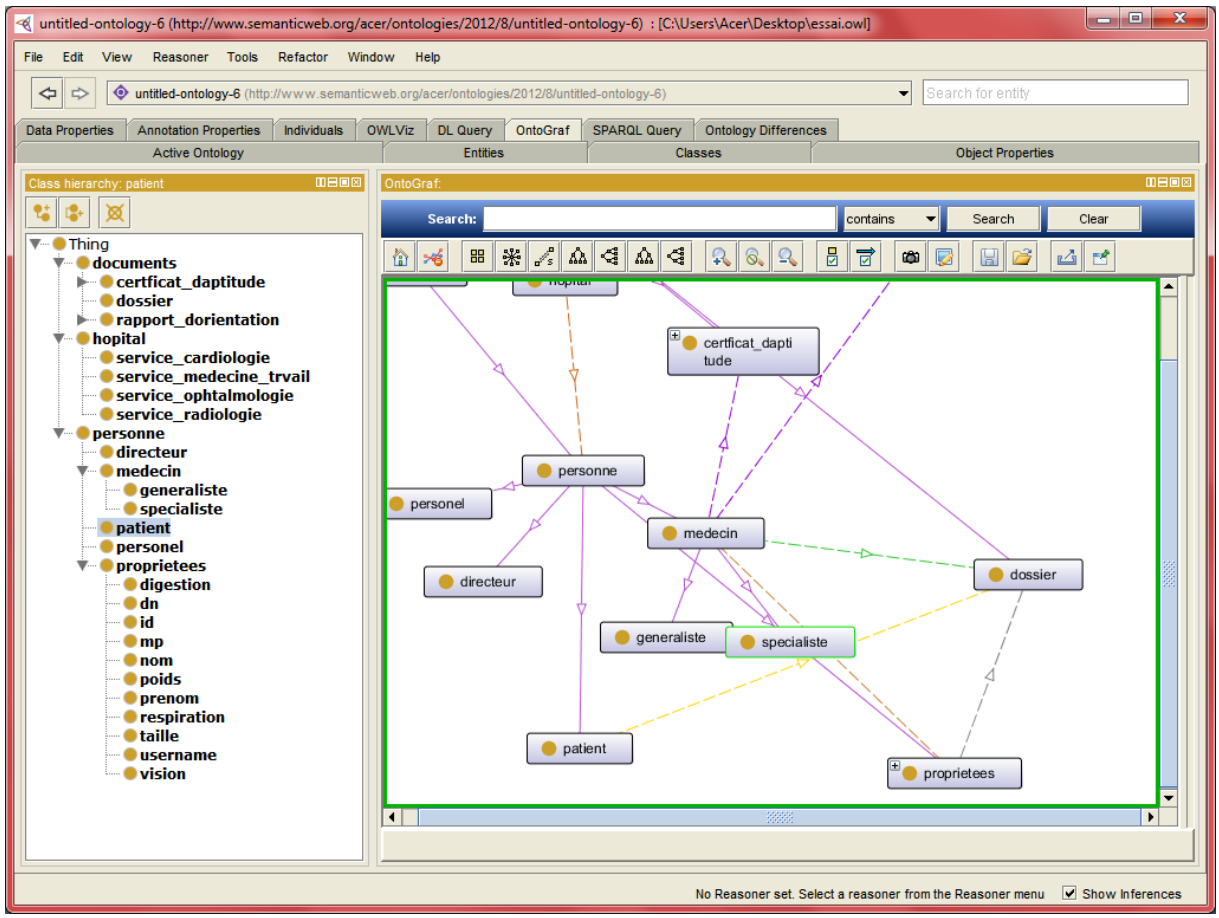



**Figure 17 création de l'ontologie avec Protégé**

Nous avons utilisé l'OWL-S pour associer la description sémantique aux services web développés utilisant les termes définis dans l'ontologie de domaine créée. Le but de l'OWL-S est d'offrir un moyen pour le développement d'un web sémantique où les services peuvent être découverts et exécutés automatiquement. Nous détaillons dans ce qui suit un exemple sur le service «ChercherDossier » de l'ajout de la sémantique à un service web.

Nous avons utilisé l'éditeur OWL-S qui est un plug-in Eclipse permettant la génération des fichiers OWL, un service décrit avec OWL-S est une instance de l'ontologie OWL-S. nous créons donc des instances des classes *Service*, P*rofile, Process* et *Grounding*.(voir Figure)

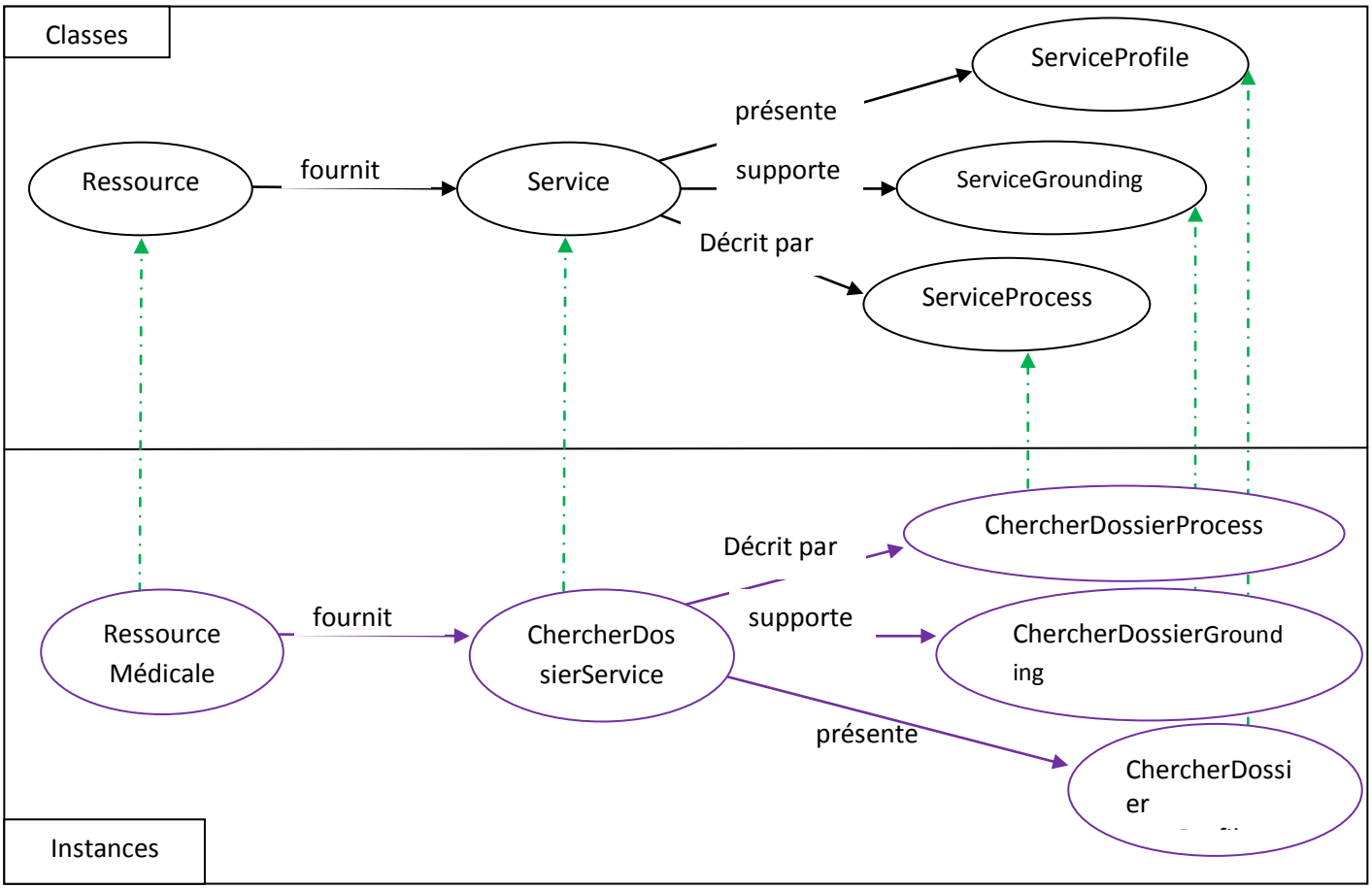



**Figure 18 Exemple de la création des instances OWL-S du service ChercherDossier.**

La figure19 montre l'ajout du Profile ou on fournit le nom du service, une description, les acteurs, les entrées, sorties, préconditions et les résultats.

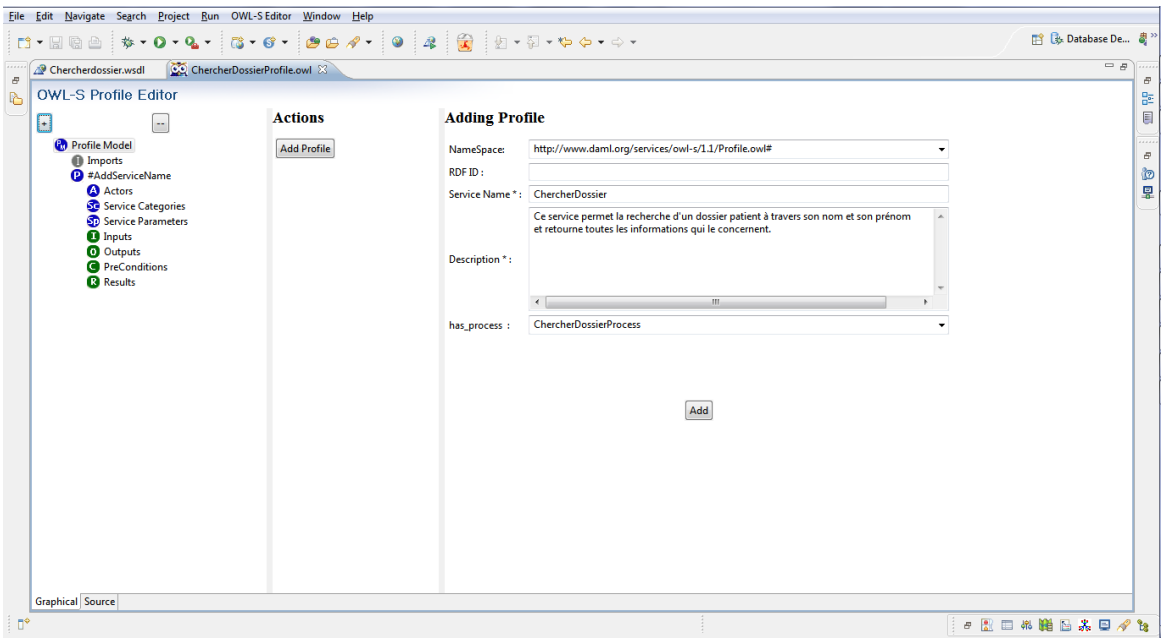



**Figure 19 Description du profile**

Ensuite nous ajoutons la description des inputs, outputs, préconditions et des effets tel que décrit dans la figure 20 où chaque paramètre contient un champ RDF ID permettant d'associer ce paramètre au concept décrit dans l'ontologie.

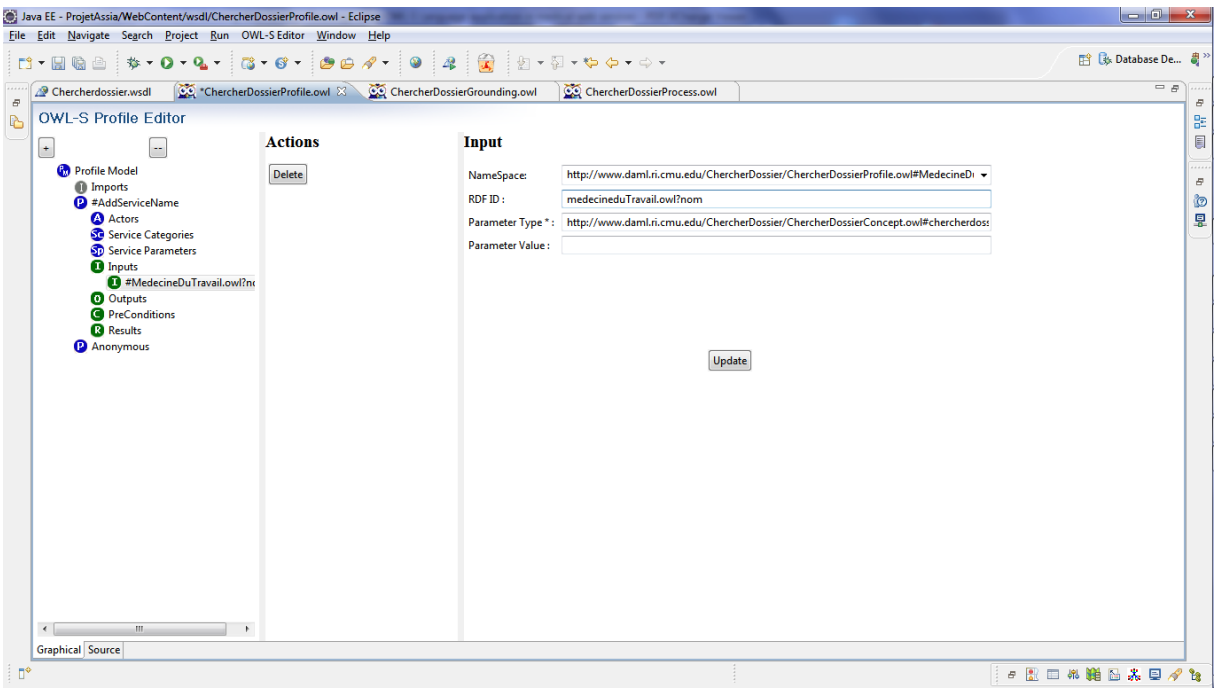



**Figure 20 Ajout de la description d'un intpout (nom).**

Après la définition de Profile, on décrit le Grounding, celui-ci permet de décrire les détails sur comment accéder au service, et le Process contient une description sur la fonctionnalité du service.

# 6. Les interfaces de l'application

L'interface homme/machine représente l'élément clé dans l'utilisation de tout système informatique. Les interfaces de notre système de recherche sont conçues de manière à être simples, naturelles, compréhensible et d'utilisation faciles.

Pour accéder à notre application, l'utilisateur doit d'abord lancer le serveur web TOMCAT pour se connecter avec l'application du services Web.

Au lancement de l'application, la fenêtre principale s'affiche.


**Figure 21 fenetre principale**

Après la fenêtre principale, c'est la fenêtre d'authentification qui s'affiche, via cette fenêtre le médecin doit s'authentifier pour bénéficier des fonctionnalités de notre système.


**Figure 22 fenêtre d'authentification**

La fenêtre du menu principale c'est la fenêtre qui contient toutes les fonctionnalités de notre application, elle fait appel à tous les services web crées au paravent.

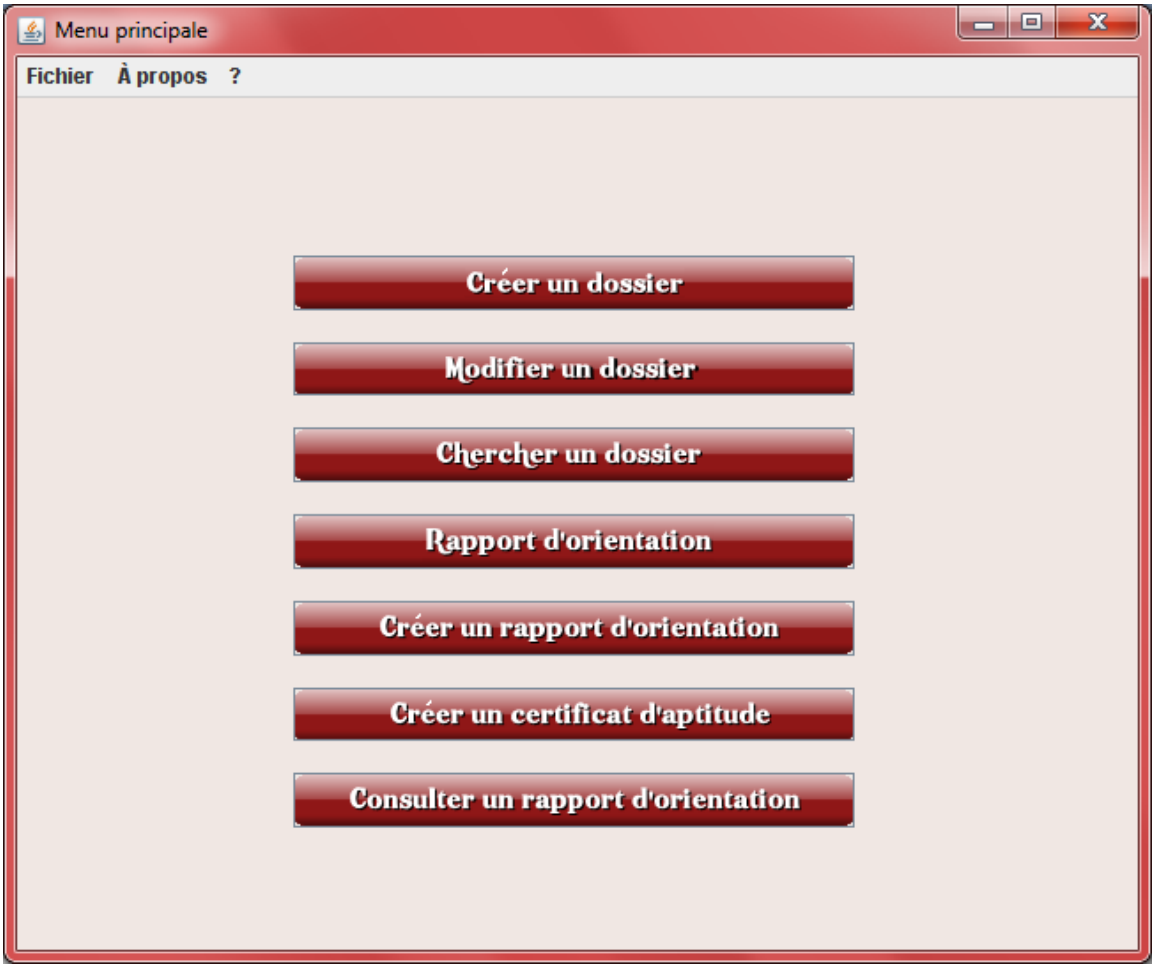



**Figure 23 menu principal**

Le bouton chercher dossier, retourne le dossier adéquat aux champs(nom, prénom) déjà rempli par le médecin grâce à l'invocation du service web chercher_dossier


## Conclusion

Dans cette partie nous avons présenté toutes les étapes qui nous ont conduit à finaliser notre application « MserS », allant de la modélisation avec UML jusqu'à l'implémentation du code source java

# Conclusion générale

# Conclusion générale

Afin de pouvoir avancer et atteindre progressivement un niveau de développement plus élevé dans nos structures hospitalière, nos hôpitaux devront prendre en considération les nouvelles évolutions apportées au niveau de la gestion et la compréhension de l'information électronique distribué.

Parmi les nouvelles techniques dans ce domaine, nous trouvons que les services web sont les plus adaptés à notre application, or les nouveaux travaux menés autour de la description des services Web utilisent de plus en plus les ontologies pour fournir une représentation de l'information sémantique, à la fois, détaillée, riche et facile à manipuler par les machines.

Dans le présent travail nous avons réalisé une application qui renforce la coopération entre les différents services de nos hôpitaux, et cela en faisant collaborer les médecins de la manière la plus simple possible, notre système est composé d'un ensemble de services Web élémentaires qui invoquent d'autre service web lors de la réalisation d'une tache complexe, ces services peuvent être générés sémantiquement en utilisant le vocabulaire de l'ontologie de domaine.

Tout travail est amené à être amélioré, en ce sens, notre système peut encore évoluer et se voir améliorer. Parmi les perspectives à prendre en compte nous citons notamment :

- Enrichissement de notre ontologie pour couvrir plusieurs domaines de recherches ;
- Extension de notre application pour supporter les processus métiers et la composition dynamique des services Web ;
- Etendre notre système, en perspective de réponde aux besoins d'autres services hospitaliers(cardiologie,radiologie,…) et non seulement au service médecine travail.

# Références bibliographique

# Bibliographie :

[1] Florence Amardeilh, Web Sémantique et Informatique Linguistique :propositions méthodologiques et réalisation d'une plateforme logicielle, thèse de doctorat, Université Paris X – Nanterre,2007.

[2] Baget J.-F., Canaud E., Euzenat J. & Hacid M.-S., Les langages du Web Sémantique, in Le Web sémantique, , Hors série de la Revue Information -Interaction - Intelligence (I3), 4(1), Cépaduès, Toulouse, pp. 21-43,2004.

[3] Fabien Baligand, Une Approche Déclarative pour la Gestion de la Qualité de Service dans les Compositions de Services ,thèse de doctorat, l'Ecole des Mines ,Paris,2008 P16 P20

[4] Mickaël Baron, developper avec java : jax-ws, Ecole Nationale Superieure de Mecanique et d'Aerotechnique –ENSMA-, 2010.

[5] M.Blaquez,M.Fernandez, J.Garcia-Pinar, et a.Gomez-Perez « building ontologies at the knowledge level using the Ontology Design Environment", In proceeding of the Banff Workshop on Knowledge Acquisition for Knowledge-based Systems,1998.

[6] Maroun Chamoun,« Intégration de l'Internet 3G au sein d'une plateforme active, École Doctorale d'Informatique, Télécommunications et Électronique de Paris,2006.

[7] Dieng R., Corby O., Gandon F., Giboin A., Golebiowska J., Matta N. & Ribiere M] .Méthodes et outils pour la gestion des connaissances : une approche pluridisciplinaire du knowledge management. Dunod, 2 edition,2001.

[8] Jean Michel Doudoux, Développons en Java, didacticiel Version 1.20 du 29/10/2009.

[9] Maha Driss, Approche multi-perspective centrée exigences de composition de services Web, thèse de doctorat, Université de Rennes,2011.

[10] S. Dustdar and W. Schreiner, A survey on web services composition, International Journal of Web and Grid Services 1, no. 1, 1-30,2005.

[11] T. Erl, SOA principles of service design, Prentice Hall, 2007.

[12] Lécué Freddy, Composition de web services sémantiques, rapport de stage Master-R en Informatique, l'Ecole Nationale Supérieure des Mines de St-Etienne,2005 .

[13].F.Furst, « l'ingénierie ontologique »,Rapport de recherche N°02-07,2002.

[14] Gomez-Perez A., "Ontology Engineering". Springer Verlag, 2003.

[15] Antonio Goncalves, Les Cahiers du programmeur Java EE 5, Editions Eyrolles, 2008.P46

[16] Guarino N., "Undestanding, building, and using ontologies". International journal of Human and Computer Studies, Vol. 45 (No. 2/3), pp 293-310, 1997.
[17] Guarino N., "Some Ontological Principles for Designing Upper Level Lexical Resources". Proceedings of First International Conference on Language Resources and Evaluation ELRA - European Language Resources Association, Granada, Spain, pp. 527-534, 1999.
[18] Van Heijst G., Schreiber A. Th., and Wielinga B. J., "Using explicit ontologies in KBS development". International Journal of Human-Computer Studies, Vol. 46 (No. 2/3) pp.183-292, 1997.
[19] M. N. Huhns and M. P. Singhl, Service-oriented computing : Key concepts and principles, IEEE Internet Computing 9 no. 1, 75-.81,2005.
[20] Said Izza, Intégration des Systèmes d'Information Industriels Une approche flexible basée sur les services sémantiques, thèse de doctorat, Ecole Nationale Supérieure des Mines de Saint-Etienne, 2006.
[21] Abdel kader Keita, Conception Coopérative d'Ontologies Pré-Consensuelles :Application au domaine de l'Urbanisme, thèse de doctorat, Institut national des sciences appliquées Lyon,2007.

[22] Cristina Marin, Une approche orientée domaine pour la composition de services, these de doctrat, Universite Joseph Fourier(Grenoble I) , 2008

[23] Didier Miclo, Web service et web sémantique, en route vers l'interopérabilité intelligente, mémoire de master Informatique, Ecole d'ingénierie informatique CS2I Nevers,2009.P13
[24] Mizoguchi R., "Ikeda: Task Ontologies for reuse of Problem Solving Knowledge". In N. J. I. Mars (editors), Towards Very Large Knowledge Bases, IOS Press, 1995. P31
[25] Benoit Schwind, Composition automatique et adaptative des services web de météorologie, thèse de doctorat, école nationale supérieure des mines de Paris, 2009.

[26] Barry Smith , Preprint version of chapter "Ontology", in L. Floridi (ed.),Blackwell Guide to the Philosophy of Computing and Information, Oxford: Blackwell, 2003
[27] Guillermo Valente Gomez Carpio, Enrichissement de requêtes et visualisation sémantique dans une coopération de systèmes d'information : méthodes et outils d'aide à

la recherche d'information, thèse de doctorat, Université Bourgogne, 2010.

[32] Berners-Lee, T., Hendler, J., Lassila, O. The Semantic Web. In: ScientificAmericain, 2001.

[29] Payne T.R., Lassila O. Semantic Web Services. IEEE Intelligent Systems. 2004, vol.19, n°4, pp.14-15.

[30] McIlraith, S., Martin D. Bringing Semantics to Web Services. IEEE Intelligent Systems. 2003, vol.18, n°1, pp.90-93.,

[31] Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P.F.,editors. The Description Logic Handbook: Theory, Implementation and Applications.Cambridge University Press, Cambridge, UK, 2003, 555p.

[28] Fensel D., Bussler C., Ding Y., Omelayenko B. The Web Service Modeling Framework WSMF. Electron Commerce Res, vol.1, n°2, 2002, pp.113–137.

[33] Roman, D., Keller, U., Lausen, H., de Bruijn, J., Lara, R., Stollberg, M., Polleres, A., Feier, C., Bussler, C., Fensel, D. Web Service Modeling Ontology. Appliedontology, 2005, vol.1, pp.77-106

[34] Denivaldo Cicero Pavão Lopes. *Étude et applications de l'approche MDA pour des plates-formes de Services Web*. l'UFR Sciences et Techniques, Université de Nantes, 2005

# Webographie:

[35] http://liris.cnrs.fr/Documents/Liris-4347.pdf

[36]http://www.xmlenst.fr/van-gogh.xml

[37] http://www.w3.org/RDF/

[38] http://www.w3.org/TR/2004/REC-rdf-schema-20040210/

[39] RQL: The RDF query language. Site Web: http://139.91.183.30:9090/RDF/RQL/

[40]http://fr.wikipedia.org/wiki/Prot%C3%A9g%C3%A9_%28logiciel%29

[41]http://fr.wikipedia.org/wiki/Apache_Tomcat

[42]http://fr.wikipedia.org/wiki/MySQL

[43 ]Web services glossary, 2004. http://www.w3.org/TR/2004/NOTE-ws-gloss-20040211/.

## Résumé

L'intégration des ontologies et de la technologie des services Web dans le domaine médicale, peut apporter une contribution primordiale dans la réutilisation des ressources médicales ou dans l'échange d'informations. Dans ce mémoire nous nous sommes intéressées à créer une application qui offre aux médecins la possibilité de communiquer entre eux à l'aide des services web élémentaires ou composés, tout en se basant sur une ontologie du domaine.

**Mots clés :** Ontologie, service Web , Service web sémantique, composition de service web

## Abstract

The use of web service technology in the medical field, can make a major contribution, especially when we add the semantic dimension (the ontology). These two factors allow the reuse of medical resources and permit a high level of interoperability. In this work we are interested in creating an application which offers to the doctors the possibility of communication; these processes are based on domain ontology and web services.

**Key words :** Ontology, Web service, Semantic web service, composition of web service.

## ملخص

الدمج بين الأنتولوجيا وخدمات الويب في مجال الطب ، يسمح بتحقيق مساهمات فعّالة ، سواء من خلال إعادة استعمال الموارد الطبية أو من خلال تبادل المعلومات. لقد تطرقنا من خلال هذا العمل إلى انجاز برنامج يمنح الاطباء امكانية التواصل بواسطة خدمات الويب وذلك بالاعتماد على الأنتولوجيا.

**الكلمات الرئيسية** :الأنتولوجيا , خدمات الويب , خدمات الويب دات المعنى , تركيب خدمات الويب .