

UNIVERSITE DE BLIDA 1

Faculté des Sciences

Département d'Informatique

MEMOIRE DE MAGISTER

Spécialité : Informatique Répartie et Mobile (IRM)

**LA CONFIANCE DANS
L'INTERNET DES OBJECTS**

Par

SELT Rachid

Devant le jury composé de

Mme. S. OUKID	Maître de Conférences A, U. de Blida 1	Président
Mme. N. BOUSTIA	Maître de Conférences A, U. de Blida 1	Examinatrice
M. D. DJENOURI	Maître de Recherche A, CERIST	Examineur
Mme. N. BENBLIDIA	Professeur, U. de Blida 1	Co-promotrice
M. Y. CHALLAL	Maître de Conférences A, ESI	Promoteur

Blida, Octobre 2015

RESUME

L'Internet des Objets est une technologie émergente qui a connu une croissance rapide dans les dernières années. Elle représente une extension de l'internet vers le monde physique qui se base sur les objets ubiquitaires. De ce fait, les problèmes liés à la confidentialité des données et la sécurité des échanges présentent un challenge pour les concepteurs. Afin de faire face à ces problèmes, des approches suffisamment sophistiquées sont nécessaires pour assurer les services demandés, et, par conséquent, garantir une adoption à grande échelle de cette technologie. Dans notre travail, nous présentons une approche pour modéliser et décrire l'architecture de l'internet des objets. Dans cette dernière, les contraintes liées à la vie privée et la sécurité des utilisateurs sont assurées, tout en prenant en considération le contexte dynamique dans lequel évoluent les utilisateurs, leur niveau d'expertise et leurs préférences en matière de sécurité et de confidentialité.

Mots-clés: internet des objets, vie privée, sécurité, sensibilité au contexte et ontologie.

ملخص

انترنت الأشياء هي تكنولوجيا سريعة النمو في السنوات الأخيرة, إنها تمثل امتدادا للإنترنت في العالم المادي. لذلك خصوصية وامن المستخدمين أصبحت تحديا كبيرا . للتعامل مع هذه القضية، هناك حاجة إلى نهج متطورة لضمان هذه الخدمات، وبالتالي ضمان اعتماد نطاق واسع لإنترنت الأشياء. في عملنا هذا نقدم نموذج و وصف بنية إنترنت الأشياء، حيث يتم ضمان خصوصية وامن المستخدم مع الأخذ بعين الاعتبار السياق الديناميكي المتطور للمستخدمين، خبراتهم و تقضياتهم فيما يتعلق بالأمن والخصوصية .

كلمات مفتاحية : إنترنت الأشياء , الخصوصية , الحماية , سياق المستخدم و انتولوجيا

ABSTRACT

The Internet of Things (IoT) is a rapidly growing technology in recent years. It represents an extension of the Internet into the physical world embracing everyday objects. In consequence, users' privacy and security are becoming a great challenge. To cope with this issue, sophisticated approaches are needed to guarantee these services and hence ensure a large-scale adoption of IoT. In our work we present an approach to model and describe the architecture of internet of things, in which user privacy and security are ensured while taking into consideration the dynamic context in which evolve users, their experience and preferences with respect to security and privacy.

Keywords: Internet of things, privacy, security, context-aware and ontology.

REMERCIEMENT

Mes remerciements sincères et profonds s'adressent à M. CHALLAL Yacine, Maître de conférences au ESI, mon promoteur, de m'avoir proposé ce sujet de Magister. Qu'il trouve ici l'expression de ma parfaite gratitude, pour l'intérêt qu'il a porté à mon travail, pour ses conseils et orientations, et surtout pour son support.

Je tiens à exprimer ma reconnaissance à Mme. BENBLIDIA Nadja, Professeur à l'USDB, d'avoir acceptée de codiriger ce travail et pour l'aide indéniable qu'elle m'a apporté.

Je souhaite remercier profondément Mme. OUKID Saliha, Maître de conférences à l'USDB qui m'a fait l'honneur de présider mon jury de soutenance.

J'exprime mes vifs remerciements à Mme. BOUSTIA Narhimene, Maître de conférences à l'USDB, et à M. DJENOURI Djamel, Maître de recherche à CERIST, d'avoir acceptés de juger mon travail.

Je remercie également tous mes amis et mes collègues pour leurs encouragements, leurs aides et leur support.

DEDICACES

A mes parents...

TABLE DES MATIERES

RESUME.....	
REMERCIEMENTS.....	
TABLE DES MATIERES.....	
LISTE DES ILLUSTRATIONS, GRAPHIQUES ET TABLEAUX.....	
INTRODUCTION GENERALE.....	12
1. CONCEPTS DE BASE	
1.1 Introduction.....	15
1.2 Contexte.....	15
1.3 Notion d'ontologie.....	20
1.4 Cloud Computing	32
1.5 Conclusion.....	35
2. LA SECURITE DANS L'INTERNET DES OBJETS	
2.1 Introduction.....	37
2.2 Internet du futur.....	37
2.3 Informatique ubiquitaire.....	38
2.4 Intelligence Ambiante.....	39
2.5 Internet des objets.....	39
2.6 Historique.....	40
2.7 Technologies de base.....	41
2.8 Applications.....	43
2.9 Protection de la vie privée.....	44
2.10 La sécurité et la sensibilité au contexte.....	44
2.11 Sécurité dans l'internet des objets.....	45
2.12 Conclusion.....	55
3. CONCEPTION DE L'ARCHITECTURE SENSIBLE AUX CONTEXTE POUR IoT	
3.1 Introduction.....	56

3.2 Architecture proposée.....	56
3.3 Représentation du contexte.....	62
3.4 Conclusion.....	67
4. EXPERIMENTATIONS	
4.1 Introduction.....	69
4.2 Environnement de développement.....	69
4.3 Expérimentation.....	72
4.4 Conclusion.....	83
CONCLUSION GENERALE	84
REFERENCES	86

LISTE DES ILLUSTRATIONS, GRAPHIQUES ET TABLEAUX

Figure 1.1	Les trois niveaux composant une ontologie	23
Figure 1.2	Le Web sémantique (Layer cake)	25
Figure 1.3	structure d'un fichier OWL	28
Figure 1.4	les type du cloud computing	33
Figure 1.5	Origine des Web services sémantiques	35
Figure 2.1	internet du futur	38
Figure 2.2	Évolution du marché mondial des ordinateurs	38
Figure 2.3	objet dans l'internet des objets	40
Figure 2.4	historique d'IdO	41
Figure 2.5	Exemple de représentation d'un réseau de capteurs	42
Figure 2.6	RFID architecture	42
Figure 2.7	Un exemple de QR Code (site web université de BLIDA)	43
Figure 2.8	représentation graphique des défis de sécurité dans l'IdO	45
Figure 2.9	3C modèle pour IdO	47
Figure 2.10	Sécurité modèle pour l'IDO	48
Figure 2.11	sécurité dans IoT	48
Figure 2.12	Architecture de Hydrogen	50
Figure 2.13	Environnement Domotique Intelligent	52
Figure 2.14	Architecture d'un « sentient object »	53
Figure 2.15	Architecture global de FI-Ware	54
Figure 2.16	Architecture de Cognitive Management Framework	54

Figure 3.1	les trois couches de l'IOT	57
Figure 3.2	présentation graphique de l'architecture avec middleware	59
Figure 3.3	diagramme de séquence de l'architecture avec middleware	59
Figure 3.4	diagramme de séquence de l'architecture sans middleware	60
Figure 3.5	présentation graphique de l'architecture	61
Figure 3.6	l'ontologie principale SOUPA-Core et extensions	63
Figure 3.7	l'ontologie SOUPA-Policy	63
Figure 3.8	Ontologie haut niveau du CONON	64
Figure 3.9	exemple d'ontologie de CONON	65
Figure 3.10	diagramme du couplage	66
Figure 3.11	couplage des ontologies	67
Figure 4.1	Utiliser Jess avec OWL	71
Figure 4.2	Exemple de contexte (université de BLIDA)	73
Figure 4.3	Exemple de contexte avant l'inférence	74
Figure 4.4	Exemple de contexte après l'inférence	74
Figure 4.5	Temps de chargement pour l'ontologie de haut niveau et l'ontologie de domaine spécifique	76
Figure 4.6	Les résultats du Temps de transfert et d'exécution pour l'exemple de localisation	77
Figure 4.7	Les résultats du Temps de transfert et d'exécution pour l'exemple médical	77
Figure 4.8	Résultat montre la différence entre la première et la deuxième exécution pour l'exemple de localisation.	78
Figure 4.9	Résultat montre la différence entre la première et la deuxième exécution pour l'exemple médical	78
Figure 4.10	Résultat du temps de raisonnement	79
Figure 4.11	La fenêtre principale de notre application	80
Figure 4.12	La fenêtre pour sélectionner l'ontologie.	80

Figure 4.13	La fenêtre d'ouverture de l'Ontologie	81
Figure 4.14	La fenêtre de l'application pour l'exemple de localisation	81
Figure 4.15	La fenêtre de l'application pour l'exemple de localisation	82
Figure 4.16	La fenêtre de l'application pour l'exemple médical	82
Figure 4.17	La fenêtre de localisation autorisée	83
Figure 4.18	La fenêtre de localisation non autorisée	83
Table 4.1	règles SWRL pour l'exemple de localisation	75
Table 4.2	règle SWRL pour l'exemple médical	75

INTRODUCTION GENERALE

Contexte général

L'internet des objets fait partie intégrante de l'internet du futur et peut être défini comme une infrastructure de réseau globale et dynamique basée sur des protocoles de communication interopérable où les objets réels et virtuels ont une identité qui par le biais d'interfaces intelligentes sont parfaitement intégrés dans un réseau d'information.

L'internet des objets utilise souvent des informations personnelles pour personnaliser les services rendus à l'utilisateur. La plus courante étant sa localisation, des préférences ou un profil, un compte, d'autres informations de capteurs. Toutefois, même si chacune de ces informations ne semble pas représenter de menace, leur corrélation peut se révéler invasive.

En informatique ambiante, le contexte joue un rôle capital. Les applications informatiques étendent leurs interactions avec l'environnement : de nouvelles entrées et sorties sont utilisées, telles que des capteurs et d'autres dispositifs mobiles d'interaction avec l'environnement réel et physique.

Par ailleurs, la modélisation des informations de contexte est primordiale pour leur gestion à savoir leur transport, leur échange et leur stockage mais vu la diversité des informations de contexte, cette modélisation est étroitement liée au domaine de leur utilisation.

Les ontologies ont, entre autres, pour objectif de faciliter la communication entre les systèmes afin de partager leurs données hétérogènes et distribuées. D'un point de vue général, les ontologies définissent des représentations standard des phénomènes du monde réel permettant à l'ensemble des acteurs d'une communauté de communiquer sans ambiguïté.

La représentation des connaissances par les ontologies peut s'accompagner des mécanismes de raisonnement. Le raisonnement concerne la manipulation des connaissances déjà acquises pour produire de nouvelles connaissances.

Problématique et objectifs de l'étude

Le problème qui découle immédiatement de l'augmentation du volume de données personnelles en circulation sur l'Internet des Objets est celui de la sécurité des transferts. Fraude de la carte à puce, détournements de comptes utilisateurs (Hôpital, Banque, Assurance. Etc.).

Problématique : la sécurité et la confidentialité de la vie privée sont les défis majeurs dans l'internet des objets. Notre travail présentera ces défis et quelques modèles de sécurité proposés.

L'objectif de notre travail est l'extraction des politiques de confidentialité, à partir des connaissances capitalisées dans l'ontologie. Pour permettre à l'utilisateur de contrôler le partage et l'utilisation de ses renseignements dans le contexte de l'internet des objets. La définition des politiques de confidentialité spécifiques aux utilisateurs fait aussi partie de nos principaux objectifs.

Notre travail consiste à raisonner à partir d'une ontologie des politiques de confidentialités et développer des requêtes qui permettent de construire des politiques de confidentialité respectant un profil de sécurité ou pas selon la demande de l'utilisateur. Aussi de définir les règles SWRL (Semantic Web Rule Language) permettant d'une part d'enrichir la sémantique de l'ontologie et d'inférer de nouveaux politiques de sécurité.

Contribution

Notre contribution dans ce mémoire, consiste en la proposition d'une approche de sécurité basée sur le contexte et les ontologies. Elle utilise plusieurs techniques telles que l'inférence et le raisonnement d'information, les politiques de sécurité et les services web, elle utilise aussi l'informatique en nuage.

Organisation du mémoire

Ce mémoire est structuré en quatre chapitres :

Le premier chapitre est consacré à la présentation de concepts de base concernant les ontologies la sensibilité au contexte et l'informatique en nuage. Nous commençons par la sensibilité au contexte. Et nous présentons après les origines de la notion d'ontologie et ses différentes définitions en ingénierie des connaissances. Nous présentons ensuite ses diverses classifications. Nous abordons les langages utilisés pour la représentation des ontologies. La fin du chapitre est consacrée aux l'informatique en nuage et ces différents types et services.

Dans le deuxième chapitre, nous présentons la notion de l'internet des objets,

l'objectif est de répertorier les informations fondamentales permettant de se familiariser avec les différentes notions et concepts liés à l'internet des objets. Dans ce chapitre, nous présentons quelques définitions et une présentation générale des technologies liées (RFID, QR Code et NFC). Ensuite, les caractéristiques techniques ainsi que les domaines d'application les plus connus, nous nous sommes intéressés aux approches de sécurité et la vie privée de l'utilisateur dans l'internet des objets, tout en mettant l'accent sur quelques travaux de recherche récents où la sécurité et la vie privée sont basées sur le contexte de l'utilisateur. Nous avons opté pour cette dernière approche pour proposer notre propre modèle de l'architecture.

Le troisième chapitre présente notre contribution, la conception d'un modèle de confiance adaptée aux contraintes de l'Internet des Objets et ses besoins en termes de sécurité, l'identification des principales couches ainsi que l'échange de messages entre ces couches. Nous avons utilisé, aussi, les différents diagrammes UML pour présenter les interactions et l'échange de données entre les acteurs de l'architecture.

Le quatrième chapitre concerne l'application de notre approche de sécurité sur un exemple. Nous avons choisi comme exemple l'utilisation de deux ontologies développées séparément; elles concernent le domaine de localisation et médicale. Aussi Nous présentons l'environnement de développement qui va supporter notre approche de sécurité. Nous exposons le prototype d'expérimentation afin de valider notre approche.

En conclusion nous présentons les points importants de ce travail et quelques perspectives qui découlent de cette étude.

CHAPITRE 1

CONCEPTS DE BASE

1.1 Introduction

Le contexte de sécurité fait le lien entre la sensibilité à un contexte et un environnement considéré comme étant de confiance, de plus, la sécurité des informations contextuelles, plus particulièrement pour la protection de la vie privée, qui reste un problème grandissant de l'informatique.

Ce chapitre ne vise qu'à fournir une vision générale sur la sensibilité au contexte, les ontologies et l'informatique en nuage, Il a pour objectif de présenter des termes, des concepts et des notions de base nécessaires, afin de nous disposons d'un bagage théorique pour qu'il nous faciliterons l'accomplissement de notre travail.

1.2 Contexte

Le terme contexte est utilisé dans de nombreux domaines, et n'a pas de définition unique. Nous trouvons sa définition dans les dictionnaires. Pour le Petit Robert, le contexte est "l'ensemble du texte qui entoure un élément de la langue (mot, phrase, fragment d'énoncé) et dont dépend son sens, sa valeur". et Pour Larousse, "Ensemble des conditions naturelles, sociales, culturelles dans lesquelles se situe un énoncé, un discours. ". On y trouve une deuxième définition : "Ensemble des circonstances dans lesquelles se produit un événement, se situe une action".

Il existe plusieurs définitions du contexte, qui varient selon le domaine d'utilisation et de l'objectif final de son application. La définition courante la plus utilisée est celle de Dey et al. [4] qui définit le contexte comme étant :

«Toute information qui peut être utilisée pour caractériser la situation d'une entité. Une entité est une place, une personne, ou un objet qui est considéré pertinent à l'interaction entre un utilisateur et une application, incluant l'utilisateur et l'application eux-mêmes »

Une définition plus générale a été proposée par Zainolet Nakata [74] qui considère «

qu'une information est dite de contexte quand elle désigne une situation qui est pertinente pour l'interaction entre l'utilisateur et l'environnement de l'application »

Le contexte est également formalisé selon des modèles mathématiques. Par exemple, [16] propose un modèle cumulatif où le contexte(Ctx) est une agrégation temporaire de situations. Une situation est un descripteur d'état pour un utilisateur (U) exécutant une tâche (T) à un certain moment (t). Le modèle est décrit par la formule suivante:

$$Ctx(U, T, t) = \bigcup_{n=1}^m (Situation(U, T, t_n))$$

1.2.1 Sensibilité au contexte

Son utilisation en informatique est nouvelle. Il est apparu dans le milieu des années quatre-vingt dix. Schilit et Theimer dans [8] ont été les premiers à introduire le terme "context-aware" (sensible au contexte) mais leur définition du contexte ne porte que sur des variables comme la localisation, l'identité des personnes proches et des objets ainsi que les changements dans ces objets.

Dey et Abowd [19] proposent qu'un système soit sensible au contexte s'il utilise des informations du contexte pour mettre à disposition des informations ou des services utiles à l'utilisateur, l'utilité dépendant de la tâche de l'utilisateur.

Dans [50], les auteurs affirment qu'en informatique dépendante du contexte, un utilisateur *« doit être capable d'accomplir facilement une action qui peut possiblement inclure une coopération et une collaboration avec d'autres utilisateurs en utilisant de multiples périphériques et réseaux alors qu'il se déplace dans l'environnement »*. Les périphériques et applications doivent alors s'adapter automatiquement aux besoins courants de l'utilisateur.

1.2.2 Utilisation du contexte

Le but de la prise en compte du contexte est de renforcer l'adaptabilité et le support à la décision du système. Chalmers [13] identifie cinq principales utilisations du contexte dans les systèmes dépendants du contexte comme suit :

1. Senseur du contexte : où le contexte est capté et les informations décrivant le contexte courant (température, localisation, ...etc.) sont présentés à l'utilisateur.
2. Associer le contexte aux données, encore appelé augmentation contextuelle. Par exemple : les données sur les utilisateurs peuvent être associées à leur localisation ; les notes d'une conférence peuvent être associées aux utilisateurs assistant à la conférence.

3. Permettre la découverte de ressources contextuelles.
4. Dans le cas des événements déclenchés par le contexte : pour déclencher les actions telles que l'inférence des politiques de sécurités à l'entrée dans une nouvelle région.
5. Utiliser le contexte pour modifier un service ou pour proposer des services spécialisés. Par exemple pour décrire les politiques de sécurités et les préférences dans une large variété de données.

1.2.3 Types d'informations contextuelles capturées

Les sources d'informations contextuelles sont pourtant nombreuses, ce sont souvent la localisation des utilisateurs d'une application et des utilisateurs environnants qui sont prise en compte. Chen et al dans [27] a identifié quatre types :

- **le contexte physique** : toutes les données pouvant être mesurées dans le monde physique, comme la luminosité, le bruit, la température, la distance, la vitesse.
- **le contexte utilisateur** : tout ce qui concerne l'utilisateur et ce qui l'entoure : son profil, sa localisation, son humeur, ses intentions, etc.
- **le contexte d'exécution** : toutes les données virtuelles ou logicielles : connectivité réseau, bande passante, ressources informatiques disponibles ou à proximité, activité des périphériques, variables d'environnement, etc.
- **le contexte temporel** : tout ce qui a une relation avec le temps : dates, heure, durée, agenda, saison, événements périodiques, etc.

1.2.4 Acquisition des informations de contexte

Diverses méthodes sont utilisées pour collecter les informations de contexte. Mostefaoui et al [28] distingue trois types de contextes du point de vue de leur acquisition :

- **le contexte capturé** : ce type de contexte est acquis par le moyen de capteurs matériels ou logiciels appelés aussi " sondes " telles que les capteurs de température, de pression, de niveau d'éclairage et de bruit. Les informations de contextes capturées les plus fréquemment citées dans les travaux de recherche sont la localisation des utilisateurs ou des objets et les informations sur le réseau. La localisation à l'extérieur des bâtiments est généralement capturée à l'aide d'un système GPS alors que la capture de la bande passante du réseau nécessite généralement l'utilisation de systèmes spécifiques.
- **le contexte explicitement fourni** : par exemple, les préférences de

l'utilisateur sont explicitement communiquées par l'utilisateur à l'application.

- **Le contexte dérivé ou interprété** : contrairement au contexte capturé et au contexte explicitement fourni qui représentent des informations de contexte de bas niveau (qui proviennent directement de leur source), le contexte dérivé représente un contexte de haut niveau qui peut être calculé à la volée à partir d'autres informations de contexte. Par exemple, le pays où se trouve l'utilisateur est un contexte de haut niveau déduit à partir du contexte de bas niveau qui représente ses coordonnées collectées à partir d'un GPS.

Afin de pouvoir utiliser les informations de contexte, l'application doit avoir un mécanisme qui lui délivre ces informations. Il existe deux méthodes permettant de délivrer les informations de contexte à l'application, la première est basée sur une approche où l'application est conduite par les capteurs et la deuxième est basée sur la séparation entre l'acquisition du contexte et l'application [4].

A) Approche conduite par les capteurs

L'approche conduite par les capteurs relie les pilotes des capteurs directement à l'application, ce qui oblige les développeurs de l'application à écrire le code se rapportant aux capteurs en utilisant les protocoles dictés par ces derniers. Cette approche a l'inconvénient de mélanger les détails de bas niveau de l'acquisition du contexte avec la sémantique de l'application. De plus, elle rend l'application très dépendante des capteurs et incapable d'interagir avec d'autres capteurs pour une même information de contexte.

B) Approche de séparation entre le contexte et l'application

L'approche de séparation entre le contexte et l'application permet de cacher à l'application les détails se rapportant aux capteurs. Un intergiciel générique se charge d'acquérir et stocker les informations de contexte. L'application interagit avec cet intergiciel selon deux modes :

1. le mode par interrogation qui consiste à envoyer une requête vers l'intergiciel pour récupérer la valeur du contexte.
2. le mode par notification qui consiste à s'inscrire auprès de l'intergiciel pour être notifiée par ce dernier de la valeur du contexte.

Le mode requête est plutôt utile dans le cas où l'application utilise l'information de contexte une seule fois et suppose que l'application est proactive, c'est à dire qu'elle sait quand demander l'information de contexte. Le mécanisme de notification est approprié pour

un besoin répétitif du contexte. Dans ce cas, l'application peut établir des conditions qui précisent quand elle doit être notifiée.

1.2.5 Modélisation des informations de contexte :

Les caractéristiques variées des informations de contexte comme leur hétérogénéité, leur dynamique et leur imperfection nécessitent des modèles abstraits de description de contexte. Plusieurs travaux ont défini des modèles de contexte. Nous en citons quelques uns dans ce qui suit [70].

- **païres de clé-valeur** : un des premiers travaux sur la modélisation du contexte est celui de Schilit [7]. L'information de contexte est modélisée par une paire clé valeur.
- **Modèle en couches** : le model en couche [3] fournissent un modèle de traitement du contexte basé sur des couches dans lesquels la sortie des capteurs est transformée en un ou plusieurs signaux. Ces signaux subissent un traitement pour former une description abstraite du contexte comportant un ensemble de valeurs, chacune liée à une mesure de certitude qui estime sa précision.
- **Modèle orientée objet** : Dans cette approche [37], les informations de contexte sont regroupées en un ensemble d'entités. Chaque entité représente un objet conceptuel ou physique tel qu'une personne, un dispositif ou un réseau. Les propriétés des entités tel que le nom d'une personne sont représentées par des attributs. Les entités sont liées à leurs attributs à travers des associations.
- **Modèle d'objets sensible** : ce modèle [2] a proposé une manière plus formelle pour la présentation du contexte. Le contexte est décrit en utilisant un langage basé sur le modèle entité-association et l'information de contexte est stockée au moment de l'exécution dans une base de données relationnelle.
- **Modèle d'ontologie** : La modélisation du contexte par une ontologie a un réel intérêt ; elle permet le partage des informations de contexte dans un système distribué et avec une sémantique bien définie. Une ontologie nécessite un langage de représentation basé sur un modèle sémantique. Dans les dernières années, plusieurs langages d'ontologies basés sur les technologies web ont été proposés. Nous pouvons citer parmi eux OWL (Ontology Web Language). Ces langages sont utilisés pour la modélisation du contexte.

1.3 Notion d'ontologie

Il est alors très important de définir et modéliser le contexte de l'utilisateur afin d'offrir une information de contexte aux systèmes dépendants du contexte. Cette démarche nous pousse à concevoir des modèles susceptibles de représenter les différentes composantes du contexte et les relations qui existent entre elles. La plus prometteuse de ces méthodes repose sur les ontologies.

1.3.1 Origine du terme

Ontologie est un terme qui 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** qui veut dire ce qui existe, l'Être, l'existant, et **logos** qui veut dire l'étude, le discours, d'où sa traduction par l'étude de l'Être et par extension de l'existence [24].

1.3.2 Définition

Une ontologie est un modèle conceptuel spécifique élaboré dans le domaine de la gestion du savoir. Une ontologie peut représenter des relations complexes entre des objets et inclure les règles et axiomes manquants dans un réseau sémantique. Une ontologie qui décrit le savoir dans un domaine précis est souvent reliée à des systèmes de prospection de données et de gestion des connaissances [42].

Une ontologie peut être construite en partant d'une base de connaissances qui sera raffinée et enrichie de nouvelles définitions si de nouvelles applications sont créées. Une définition a été proposée dans ce sens : « Une ontologie fournit le moyen de décrire de façon explicite la conceptualisation des connaissances représentées dans une base de connaissances [51]

1.3.3 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 basé sur les: concepts (ou classes), relations (ou propriétés), fonctions, axiomes (ou règles) et instances (ou individus) [6].

A) Les concepts

Les concepts sont utilisés dans leur sens large. Ils peuvent être abstraits ou concrets, élémentaires (*électrons*) ou composés (atomes), réel ou fictifs. En résumé, un concept peut être tout ce qui peut être évoqué : description d'une tâche, d'une fonction, d'une action, d'une stratégie ou d'un processus de raisonnement, etc. [51].

Un concept peut représenter un objet, une notion ou une idée. Il peut se définir comme une entité composée de trois éléments [26] :

- Un terme (ou plusieurs) exprimant le concept en langue.
- Une notion également appelée intension (avec 's'), contenant la sémantique du concept, exprimée en terme de propriétés et d'attributs, de règles et de contraintes. Les règles qui décrivent des inférences possibles sont des affirmations sous la forme : antécédent -> conséquent.
- Un ensemble d'objets appelé aussi extension du concept, regroupant les objets manipulés à travers le concept : ces objet sont appelés instances du concept.

Les propriétés portant sur un concept

- **La généricité** : un concept est générique s'il n'admet pas d'extension.
- **L'identité** : un concept porte une propriété d'identité si cette propriété permet de conclure quant à l'identité de deux instances de ce concept. Cette propriété peut porter sur des attributs du concept ou sur d'autres concepts.
- **La rigidité** : un concept est dit rigide si toute instance de ce concept en reste instance dans tous les mondes possibles.
- **L'anti-rigidité** : un concept est anti-rigide si toute instance de ce concept est essentiellement définie par son appartenance à l'extension d'un autre concept.
- **L'unité** : un concept est un *concept unité*, si pour chacune de ses instances, les différentes parties de l'instance sont liées par une relation qui ne lie pas d'autres instances du concept.

Les propriétés portant sur deux concepts

- **L'équivalence** : deux concepts sont équivalents s'ils ont la même extension.
- **La disjonction** : deux concepts sont disjoints si leurs extensions sont disjointes.
- **La dépendance** : Un concept C1 est dépendant d'un concept C2 si pour toute instance de C1 il existe une instance de C2 qui ne soit ni partie ni constituant de l'instance de C1.

B) Les relations

Les relations représentent un type d'interaction entre les notions d'un domaine. Elles sont formellement définies comme tout sous-ensemble d'un produit de n ensembles, c'est à dire $R : C_1 \times C_2 \times \dots \times C_n$. Comme exemple de relation binaire, on peut citer *sous-classe-de* ou encore *connecté-à* [51].

Les propriétés intrinsèques à une relation

- **Les propriétés algébriques** : symétrie, réflexivité, transitivité
- **La cardinalité** : nombre possible de relations de ce type entre les mêmes concepts (ou instances de concept). Les relations portant une cardinalité représentent souvent des attributs.

Les propriétés liant deux relations

- **L'incompatibilité** : on dit que deux relations R1 et R2 sont incompatibles si elles ne peuvent lier les mêmes instances de concepts.
- **L'inverse** : on dit que deux relations binaires R1 et R2 sont inverses l'une de l'autre si, quand R1 lie deux instances I1 et I2, alors R2 lie I2 et I1.
- **L'exclusivité** : deux relations R1 et R2 sont exclusives si, quand R1 lie des instances de concepts, R2 ne lient pas ces instances, et vice-versa. L'exclusivité entraîne l'incompatibilité.
- **Les propriétés liant une relation et des concepts** :
- **Le lien relationnel** : Il existe un lien relationnel entre une relation R et deux concepts C1 et C2 si, pour tout couple d'instances des concepts C1 et C2, il existe une relation de type R qui lie les deux instances de C1 et C2. Un lien relationnel peut être contraint par une propriété de cardinalité, ou porter directement sur une instance de concept.
- **La restriction de relation** : Pour tout concept de type C1, et toute relation de type R liant C1, les autres concepts liés par la relation sont d'un type imposé.

C) Les fonctions

Sont des cas particuliers de relations dans lesquelles le nième élément de la relation est défini de manière unique à partir des n-1 premiers. Formellement, les fonctions sont définies ainsi : $F : C1 \times C2 \times \dots \times Cn-1 \rightarrow Cn$.

D) Les axiomes

Les axiomes sont des expressions qui sont toujours vraies. Ils ont pour but de définir dans un langage logique la description des concepts et des relations permettant de représenter leur sémantique. Ils représentent les intentions des concepts et des relations du domaine et, de manière générale, les connaissances n'ayant pas un caractère strictement terminologique. Leur inclusion dans une ontologie peut avoir plusieurs objectifs [62] :

- Définir la signification des composants.
- Définir des restrictions sur la valeur des attributs.
- Définir les arguments d'une relation.

- Vérifier la validité des informations spécifiées ou en déduire de nouvelles.

E) Les instances

Les instances sont utilisées pour représenter des éléments dans un domaine. [42]

1.3.4 Les trois niveaux d'une ontologie :

Une ontologie est construite en général autour de trois niveaux (Figure 1.1) d'abstractions [51].

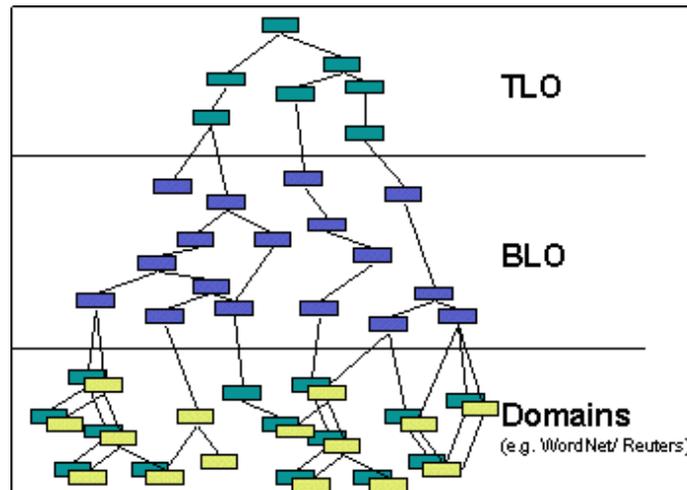


Figure 1.1 : Les trois niveaux composant une ontologie [51].

1. *Top Level Ontology*(TLO): qui regroupe des catégories de concepts de haut niveau
2. *Basic Level Ontology*(BLO): qui est un ensemble de concepts de la vie de tous les jours, et qui sont partagés par tous le monde.
3. Des Ontologies spécifiques de divers domaines qui peuvent être liées au niveau BLO.

1.3.5 Les principaux types d'ontologies

De manière générale, on identifie les catégories suivantes [51]

- *Les ontologies de représentation de connaissance* : regroupent les primitives de représentation utilisées afin de formaliser les connaissances.
- *Les ontologies générales/communes* : incluent le vocabulaire lié aux objets, aux événements, au temps, à l'espace, à la causalité, au comportement et à la fonction.
- *Les méta-ontologies* : également appelées ontologies génériques ou noyaux d'ontologies, sont réutilisables dans différents domaines.

- *Les ontologies de domaine* : sont réutilisables dans un domaine donné. Elles fournissent le vocabulaire des concepts d'un domaine et les relations entre ces derniers, les activités de ce domaine ainsi que les théories et les principes de base de ce domaine.
- *Les ontologies de tâche* : fournissent un vocabulaire systématisé des termes utilisés pour résoudre les problèmes associés à des tâches qui peuvent appartenir ou non à un même domaine. Ces ontologies fournissent un ensemble de termes au moyen desquels on peut décrire au niveau générique comment résoudre un type de problème. Elles incluent des noms génériques (par exemple plan, objectif, contrainte), des verbes génériques (par exemple assigner, classer, sélectionner), des adjectifs génériques (comme assigné) et d'autres mots qui relèvent de l'établissement d'échéances.
- *Les ontologies de domaine-tâche* : Se sont des ontologies de tâches réutilisables dans un domaine donné, mais pas dans différents domaines. Par exemple une ontologie domaine-tâche dans le domaine médical, pourrait inclure les termes liés au timing d'une intervention chirurgicale : planifier-intervention chirurgicale.
- Les ontologies d'application : Contiennent suffisamment de connaissances pour structurer un domaine particulier.

1.3.6 Langages de construction d'ontologies

1.3.6.1 Le Web sémantique

Le Web Sémantique est une vision d'un Web qui serait non-ambigüe et compréhensible par une machine. Le Web sémantique reste entièrement fondé sur le Web "classique" et ne le remet pas en cause.

1.3.6.2 L'architecture du Web Sémantique

L'architecture repose sur des technologies de structuration de documents fondées sur XML. Sur cette base, des langages de métadonnées sémantiques de haut niveau ont été développés et permettent la description de ressources sur le Web. Afin de fournir un cadre interprétatif à ces métadonnées sémantiques, le Web Sémantique utilise des ontologies. Au niveau supérieur, le raisonnement sur les données est assuré par des mécanismes d'inférence, qui permettent d'une part de construire de la connaissance, mais aussi d'en maintenir la cohérence. Au niveau le plus haut, nous retrouvons la notion de confiance dont la question

principale est : « Sur quelles bases peut on considérer comme vraie de l'information disponible sur le Web ? ». Les langages et outils développés dans le cadre du Web Sémantique ne sont que des bases technologiques. Ils ne résolvent pas à eux seuls les problèmes de présentation et d'utilisation de ces données sémantiques[65].

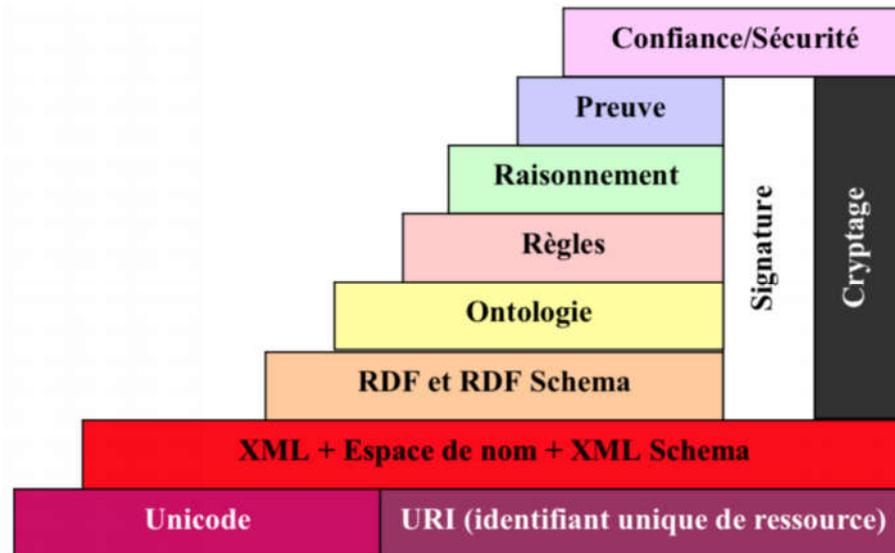


Figure 1.2: Le Web sémantique (Layer cake) [65].

1.3.6.3 XML et XMLschema

XML « *eXtensibleMarkupLanguage* » est une norme du World Wide Web Consortium (W3C) qui fournit une manière simple et standard pour échanger des informations textuelles structurées entre des applications logicielles. [66]

XML permet de décrire la structure logique de documents principalement textuels, à l'aide d'un système de balises permettant de marquer les éléments qui composent la structure et les relations entre ces éléments.

XML Schema (XML-S) est un outil de définition de grammaires caractérisant des arborescences de documents (notion de validité syntaxique). Avec les schémas XML, il est possible de contraindre la structure arborescente d'un document mais pas la sémantique des informations contenues dans ce document [1].

1.3.6.4 RDF et RDF schema

RDF « *Resource Description Framework* » est un modèle pour la représentation de métadonnées à propos de ressources. Cette représentation est faite sous la forme d'un triplet [60]:

- Sujet : la ressource que l'on définit

- Prédicat : la propriété de la ressource, qui est une liaison étiquetée et orientée du sujet vers l'objet.
- Objet : la valeur de la propriété pouvant être une autre ressource ou bien un littéral.

Si RDF fournit une capacité d'échange de connaissances, il ne permet pas à l'utilisateur de définir le vocabulaire des termes à utiliser, ni d'établir la sémantique des objets utilisés.

RDF Schema ou RDFS est un langage permettant de définir des propriétés sémantiques pour les ressources par un schéma. Dans un schéma on peut définir de nouvelles ressources comme des spécialisations d'autres ressources. Les schémas contraignent aussi le contexte d'utilisation des ressources. Avec RDFS de nouvelles notions sémantiques apparaissent. La principale est la distinction entre une classe (concept d'une ontologie) et une instance (individu d'une ontologie)

1.3.6.5 OWL

Développé par le groupe de travail sur le Web Sémantique du W3C, OWL peut être utilisé pour représenter explicitement les sens des termes des vocabulaires et les relations entre ces termes. OWL vise également à rendre les ressources sur le Web aisément accessibles aux processus automatisés, d'une part en les structurant d'une façon compréhensible et standardisée, et d'autre part en leur ajoutant des méta-informations. Pour cela, OWL a des moyens plus puissants pour exprimer la signification et la sémantique que XML, RDF, et RDF-S. De plus, OWL tient compte de l'aspect diffus des sources de connaissances et permet à l'information d'être recueillie à partir de sources distribuées, notamment en permettant la mise en relation des ontologies et l'importation des informations provenant explicitement d'autres ontologies [72].

OWL (*Ontology Web Language*) est un langage fondé sur la syntaxe RDF/XML. OWL n'est pas simplement une extension de RDF, il introduit l'aspect sémantique lui manquant, comme les outils de comparaison de propriétés et de classes (identité, équivalence, contraire, cardinalité, symétrie, etc.). Ainsi par ses primitives plus riches, OWL offre une capacité d'interprétation plus grande à la machine que RDF et RDFS.

OWL se décompose en trois sous-langages OWL Lite, OWL DL et OWL Full, offrant des capacités d'expression croissantes et donc destinés à des différentes utilisations.

OWLLite : est le sous langage de OWL le plus simple. Il est destiné aux utilisateurs qui ont besoin d'une hiérarchie de concepts simple. OWL Lite est adapté, par exemple, aux migrations rapides depuis d'anciens thésaurus.

OWL DL : est plus complexe que OWL Lite, permettant une expressivité bien plus importante. OWL DL est fondé sur la logique descriptive (*OWL Description Logics*), un domaine de recherche étudiant la logique, et conférant donc à OWL DL son adaptation au raisonnement automatisé. Malgré sa complexité relative face à OWL Lite, OWL-DL garantit la complétude des raisonnements (toutes les inférences sont calculables) et leur décidabilité (leur calcul se fait en une durée finie) c'est le sous langage OWL de notre ontologie.

OWL Full : est la version la plus complexe d'OWL, mais également celle qui permet le plus haut niveau d'expressivité. OWL Full est destiné aux situations où il est plus important d'avoir un haut niveau de capacité de description, quitte à ne pas pouvoir garantir la complétude et la décidabilité des calculs liés à l'ontologie. OWL Full offre cependant des mécanismes intéressants, comme par exemple la possibilité d'étendre le vocabulaire par défaut de OWL.

Il existe entre ces trois sous langage une dépendance de nature hiérarchique : toute ontologie OWL Lite valide est également une ontologie OWL DL valide, et toute ontologie OWL DL valide est également une ontologie OWL Full valide. [72]

1.3.6.6 Structure d'une ontologie OWL

Une ontologie OWL comporte deux sections principales : espaces de nommage (*namespace*) et entête.

Un espace de nommage

Pour pouvoir employer des termes dans une ontologie OWL, il est nécessaire d'indiquer le vocabulaire duquel ils proviennent. L'intégralité du vocabulaire d'OWL provient du *namespace* d'OWL : <http://www.w3.org/2002/07/owl#>. C'est pour cela que comme tout autre document XML, une ontologie commence par une déclaration d'espaces de noms contenue dans une balise `rdf:RDF` qui contient :

- L'espace de nommage propre à l'ontologie.
- Les espaces de nommages des ontologies importées.
- L'espace de nommage du vocabulaire OWL utilisé.
- Les objets définis dans l'espace de nommage de RDF et RDFS.
- Les types de données définis dans l'espace de nommage de XML schéma.

Un entête

Tout document OWL peut contenir un entête qui est placé juste après l'espace de nommage : il permet de décrire les informations sur le contenu de l'ontologie en cours, en utilisant la balise owl : Ontology.

Le contenu et la signification des différents éléments de cette section d'en-tête sont parfaitement décrits dans « *OWL Web Ontology Language Reference* » [49].

```

1 <?xml version="1.0"?>
2 <!DOCTYPE rdf:RDF [
3   <!ENTITY soupa    "http://pervasive.semanticweb.org/ont/2004/06/">
4   <!ENTITY rdf      "http://www.w3.org/1999/02/22-rdf-syntax-ns#">
5   <!ENTITY rdfs     "http://www.w3.org/2000/01/rdf-schema#">
6   <!ENTITY owl    "http://www.w3.org/2002/07/owl#">
7   <!ENTITY xsd      "http://www.w3.org/2001/XMLSchema#">
8   <!ENTITY act      "http://pervasive.semanticweb.org/ont/action#">
9 ]>
10
11 <owl:Ontology rdf:about="&soupa;action">
12   <owl:versionInfo>$Revision: 1.6 $, $Date: 2004/07/15 15:40:46 $</owl:versionInfo>
13   <rdfs:label>SOUPA Action Ontology</rdfs:label>
14 </owl:Ontology>
15
16 <owl:Class rdf:ID="Action">
17   <rdfs:label>Action</rdfs:label>
18 </owl:Class>
19
20 <owl:ObjectProperty rdf:ID="actor">
21   <rdfs:label>actor</rdfs:label>

```

Espaces de nommage

entête

Figure 1.3 : structure d'un fichier OWL (Soupa action ontologie).

1.3.6.7 Les éléments d'ontologie OWL

Trois types d'objets définis pour décrire une ontologie : classes, instances et propriétés.

Classes

Une classe définit un groupe d'individus qui sont réunis parce qu'ils ont des caractéristiques similaires. L'ensemble des individus d'une classe est désigné par le terme « extension de classe », chacun de ces individus étant alors une « instance » de la classe. Les trois versions d'OWL comportent les mêmes mécanismes de classe, à ceci près que OWL FULL est la seule version à permettre qu'une classe soit l'instance d'une autre classe. A l'inverse, OWL Lite et OWL DL n'autorisent pas qu'une instance de classe soit elle-même une classe [72].

Instance de classe (individu)

La définition d'un individu consiste à énoncer un « fait », encore appelé « axiome d'individu ». On peut distinguer deux types de faits :

- les faits concernant l'appartenance à une classe
- les faits concernant l'identité des individus

Un même individu peut avoir plusieurs noms, ce qui peut créer des ambiguïtés. OWL propose une solution pour lever cette ambiguïté et ceci grâce aux propriétés : owl:sameAs, owl:differentFrom et owl:allDifferent.

Les propriétés

Une propriété permet de définir des faits ou des relations entre classes. Dans OWL il existe deux types de propriétés définis par les deux classes owl:ObjectProperty et owl:DatatypeProperty qui sont des sous-classes de la classe rdf:Property. Tout comme les classes, les propriétés OWL peuvent être organisées hiérarchiquement.

- Les propriétés d'objet permettent de relier des instances à d'autres instances
- Les propriétés de type de donnée permettent de relier des individus à des valeurs de données.

Une propriété d'objet est une instance de la classe owl:ObjectProperty, une propriété de type de donnée étant une instance de la classe owl:DatatypeProperty.

Image (Range) et domaine (Domain) d'une propriété

Domain et Range signifient respectivement la partie gauche et la partie droite de la relation définie par une propriété. Le domaine d'un ObjectProperty et d'un *DatatypeProperty* est une classe OWL. Quant à leur image c'est respectivement une classe OWL et un type de données OWL.

1.3.7 Ontologies et raisonnement

La représentation des connaissances par les ontologies peut s'accompagner des mécanismes de raisonnement. Le raisonnement concerne la manipulation des connaissances déjà acquises pour produire de nouvelles connaissances. Il utilise des mécanismes d'inférence qui permettent la résolution des problèmes pour lesquels il n'existe pas de procédures explicites dans le programme. Différents mécanismes de raisonnement sont utilisés selon les objectifs du système à mettre en place : raisonnement logique, raisonnement par classification, le filtrage, l'héritage et le raisonnement à base de règles [20].

Le raisonnement logique

Le raisonnement logique se base sur un mécanisme de déduction qui utilise un ensemble de règles d'inférence pour déduire des nouveaux faits à partir des faits connus.

La logique fournit un formalisme clair et non ambigu. Cette clarté vient d'une part du fait que la signification d'une formule ne dépend que de sa structure et de la signification donnée à ses composants atomiques et d'autre part du fait que le langage d'expression logique est proche du langage naturel. De plus les connecteurs logiques (et, ou, implication et

négation) et les quantificateurs permettent une riche description du monde. Les inférences faites avec la logique du premier ordre sont correctes, complètes et fondées.

Le raisonnement par classification

Le raisonnement par classification consiste à confronter une nouvelle connaissance à un ensemble de connaissances connues pour déduire des informations liées à cette nouvelle connaissance.

Ce type de mécanisme est appliqué notamment dans les représentations terminologiques, les schémas et les représentations orientées objet. Ainsi, le mécanisme de raisonnement de base des logiques terminologiques est la classification de concepts. Cette classification est réalisée par un algorithme de classification. Pour trouver la place appropriée pour le nouveau concept, l'algorithme de classification détermine les relations de subsomption entre ce concept et les autres concepts de la hiérarchie ; ces relations peuvent être spécifiées directement, trouvées par transitivité ou bien calculées à partir de la sémantique.

Le filtrage

Le filtrage utilisé par la plupart des réseaux sémantiques. Il consiste à parcourir le graphe et à chercher tous les sous-graphes du graphe ayant des propriétés ou une structure commune avec un graphe cible.

Cette recherche correspond à un appariement de graphes. Le mécanisme de filtrage permet de récupérer des informations explicites ou implicites de la base à la manière des associations mentales de l'être humain. Le mécanisme de filtrage est également utilisé dans des représentations par schéma (frame). Dans ce cas, le filtrage recherche les schémas satisfaisant certaines caractéristiques données ; ces caractéristiques sont décrites dans un schéma appelé filtre, qui est apparié avec des schémas de la base.

L'héritage

L'héritage est un mécanisme de raisonnement qui consiste à récupérer des informations des noeuds (ou classes) représentant des concepts plus généraux, pour les utiliser dans des noeuds plus spécialisés ; cette récupération se fait en suivant les liens de spécialisation « *est-un* ». La relation de spécialisation représente l'inclusion ensembliste ; cela veut dire que toutes les instances d'une classe le sont aussi pour ses superclasses ; à ce titre, elles ont les propriétés décrites dans les superclasses.

Le raisonnement à base de règles

Le raisonnement à base de règles est également un mécanisme de raisonnement sur les connaissances. L'élément de base des systèmes à base de règles est la règle de production ; une règle a la forme suivante :

SI <condition> ALORS <action>

La partie condition est exprimée par un prédicat logique correspondant à une affirmation sur la base de connaissances qui doit être vraie au moment de valider la règle pour que l'action soit déclenchée ; la partie action, qui est la partie exécutable de la règle indique des ajouts ou modifications à faire à la base.

Il existe plusieurs systèmes de raisonnement parmi eux nous citons :

RACER (Renamed Abox and Concept Expression Reasoner)

RACER [68] est un système implémentant une logique de description. Prise en compte des représentations au format *DAML+OIL*. RACER permet le test de satisfiabilité d'un concept (vérifier qu'un concept C admet des instances) le test de subsumption de concepts (vérifier qu'un concept C est subsumé par un concept D), et le test d'instanciation (vérifier qu'un individu a est instance d'un concept C, si seulement si $a \in C$).

JENA

JENA [5] est une bibliothèque de classes Java qui facilite le développement d'applications pour le web sémantique. Elle permet de : manipulation de déclarations RDF, lecture et écriture RDF/XML, Notation, Stockage en mémoire ou sur disque de connaissances RDF, Langage d'interrogation d'une base RDF, Gestion d'ontologies : RDF-Schema.

Pellet

Le moteur Pellet [22] est un des projets du MIND-SWAP Group, un groupe de recherche sur le web sémantique de l'université du Maryland. Il est disponible en Open Source et développé en Java. Pellet offre des évolutions fréquentes, travaille sur des ontologies décrites en RDF ou OWL et permet les requêtes avec RDQL (RDF Data QueryLanguage) et SPARQL (Protocol And RDF QueryLanguage).

JESS

Jess [35] est un moteur d'inférence entièrement écrit en langage Java de Sun par Ernest Friedman-Hill au Sandia National Laboratories à Livermore, en Californie. L'utilisation de Jess permet de construire des logiciels en Java qui ont la capacité de «raisonner» en utilisant les connaissances fournies dans la forme de règles déclaratives. Jess est petit, léger, et l'un des moteurs les plus rapides disponibles.

Jess apparaît donc comme un choix de premier ordre pour ce qui est d'utiliser un

moteur de règle, son puissant langage de script donne accès à toutes les API Java. Jess inclut un environnement de développement complet basé sur la plate-forme Eclipse.

Nous avons présenté les notions de base concernant les ontologies, leurs définitions, les composants qui la définissent et terminer par le raisonnement, notre travail est d'inférer à partir d'une ontologie de contexte dans le nuage, par la suite nous définissons les concepts de base concernant l'informatique en nuage.

1.4 Cloud Computing

1.4.1 Définition

De [69] Le Cloud computing s'appuie sur une infrastructure (le nuage) composée d'un grand nombre de ressources vitalisées (par exemple : réseaux, serveurs, stockage, applications ou services), distribuées dans le monde entier. Ces ressources peuvent être allouées, puis relâchées rapidement, avec des efforts de gestion minimaux et avec peu d'interactions entre le client et le fournisseur. Aussi, cette infrastructure peut être dynamiquement reconfigurée pour s'ajuster à une charge de travail variable (passage à l'échelle). Finalement, les garanties de prestation offertes par l'informatique dans le Nuage prennent typiquement la forme de contrats de niveau de service.

1.4.2 Les types du Cloud computing

D'après [38] Nous distinguons trois types du Cloud computing. Le Cloud public, également le premier apparu, le Cloud privé et le Cloud hybride qui est en fait la combinaison des deux premiers.

- **Le Cloud public**

Le principe est d'héberger des applications, en général des applications Web, sur un environnement partagé avec un nombre illimité d'utilisateurs. La mise en place de ce type de Cloud est gérée par des entreprises tierces est accessible selon le modèle pay-as-you-go (payer selon la consommation)

- **Le Cloud privé**

C'est un environnement déployé au sein d'une entreprise. Ainsi, elle doit gérer toute seule son infrastructure. Dans ce cas, implémenter un Cloud privé signifie transformer l'infrastructure interne en utilisant les technologies telles que la virtualisation pour enfin délivrer, plus simplement et plus rapidement, des services à la demande. L'avantage de ce

type de Cloud par rapport au Cloud publique réside dans l'aspect de la sécurité et la protection des données. En effet, l'ensemble du matériel est conservé au sein de votre propre emplacement. De ce fait, les ressources sont détenues et contrôlées par votre propre département informatique.

- **Le Cloud hybride**

En général, on entend par Cloud hybride la cohabitation et la communication entre un Cloud privé et un Cloud publique dans une organisation partageant des données et des applications.

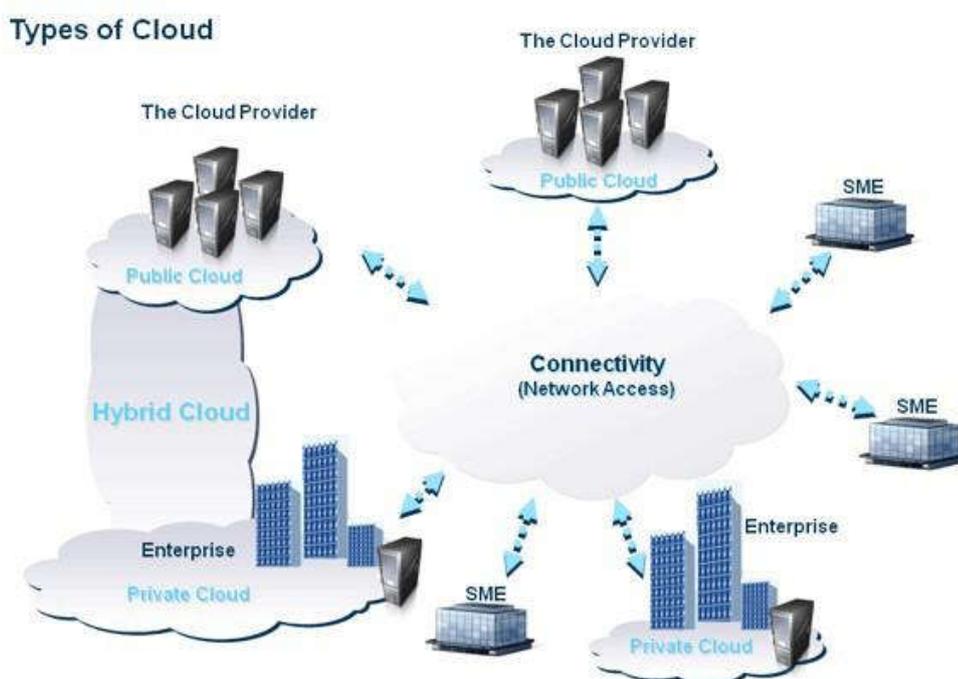


Figure 1.4 : les types du cloud computing [38].

1.4.3 Les services de Cloud Computing

[43] Le Cloud Computing permet aux entreprises de consommer des services à la demande. Les fournisseurs du Cloud distinguent trois services.

- **Infrastructure as a Service (IaaS)**

Il s'agit d'une mise à disposition, à la demande, de ressources d'infrastructures (stockage, machines virtuelles, système d'exploitation (SE), etc.) dont la plus grande partie est localisée à distance dans des Datacenter. Dans ce type de service, seul le matériel est dématérialisé

- **Plateforme as a Service (PaaS)**

Le PaaS dispose d'environnements spécialisés au développement comprenant les langages, les outils et les modules nécessaires. Dans ce type de service le matériel, l'hébergement et le Framework d'application sont dématérialisés

- **Software as a Service (SaaS)**

Sans aucune installation d'application sur l'ordinateur, le logiciel (Software) est directement utilisable à travers le navigateur Web comme service final. Ainsi, l'utilisation reste transparente pour les utilisateurs, qui ne se soucient ni de la plateforme ni du matériel. Le matériel, l'hébergement, le Framework d'application et le logiciel sont dématérialisés

En Cloud, la flexibilité est obtenue grâce à la virtualisation des systèmes d'exploitation.

La plateforme est exécutée via des machines virtuelles et les ressources peuvent être allouées et délibérées à la demande. Ainsi, l'IaaS est considéré le service le plus flexible. D'autre part, l'IaaS est le service le plus simple à mettre en place.

1.4.4 Les services Web

Définition

Un Web service est un composant logiciel identifié par une URI, qui possède une interface publiable. Cette dernière peut être découverte par d'autres systèmes, qu'ils peuvent interagir avec le Web service selon les règles prescrites par sa description, en utilisant des messages basés sur XML et portés par des protocoles standards d'internet.

Les Web Services fournissent une couche d'abstraction entre le client et le fournisseur d'un service. Cette couche est indépendante de la plateforme et du langage d'implémentation [61], grâce à un ensemble de protocoles standardisés comme **SOAP** (Simple Object Access Protocol), **WSDL** (Web Service Description Language) et **UDDI** (Universal Description, Discovery and Integration) [73].

Cette définition implique que les Web services :

- sont des composants d'application ;
- communiquent en utilisant des protocoles standard ;
- sont autonomes et auto- descriptifs ;
- peuvent être découverts ;
- peuvent être utilisés par d'autres applications ;
- s'appuient sur XML (Extensible MarkupLanguage) ; et enfin
- sont extensibles : chacun peut adjoindre ses propres données, protocoles ou

mécanismes propriétaires.

1.4.5 Web services sémantiques

Le couplage entre les Web services et le Web sémantique s'inscrit dans le cadre d'intégration des systèmes hétérogènes, plus particulièrement l'intégration sémantique. Dans ce contexte, Le concept des Web services sémantiques [17], est le fruit de la convergence du domaine des Web services avec le Web sémantique (figure 1.5). En effet, son ultime objectif est de rendre les Web services plus accessibles à la machine en automatisant les différentes tâches qui facilitent leur utilisation.

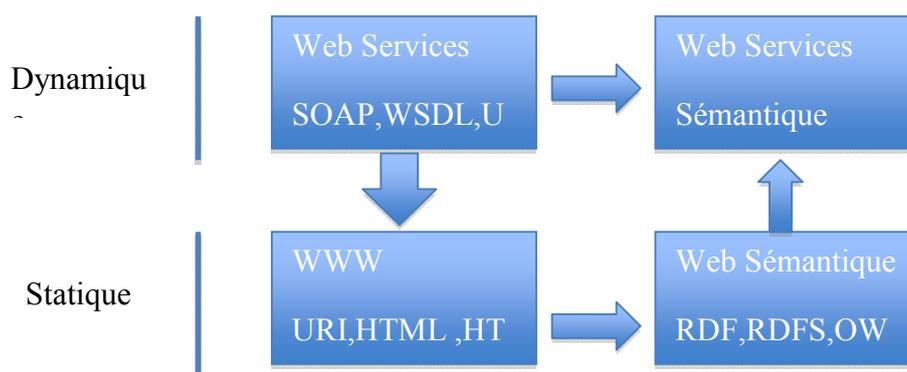


Figure 1.5 : Origine des Web services sémantiques [34].

1.4.6 OWL-S (Web OntologyLanguage-Services)

Prédécesseur de DAML-S (Darpa Agent Markuplanguage for Services), OWL-S [OWL- S] est une ontologie appelée *upperOntology* qu'on appellera ontologie supérieure OWL-S pour (Web OntologyLanguage-Services), elle est écrite en utilisant OWL et a pour objectif la description des Web services [40].

OWL-S : est une ontologie dédiée à la description des capacités et des propriétés des Web services. Le but d'OWL-S est de permettre l'automatisation de *la recherche*, de *la découverte*, de *l'invocation* et de *l'interconnexion* des Web services. Il fournit des éléments de description et spécifie les relations entre ceux-ci ainsi qu'un modèle permettant de décrire les Web services en introduisant des informations sémantique et en séparant les fonctionnalités du service de son fonctionnement interne et de la façon d'y accéder.

1.5 Conclusion

À travers ce que nous avons présenté dans ce chapitre, il ressort que la notion d'ontologie constitue l'une des approches les plus efficaces pour représenter les contextes.

Nous avons montré l'apport des ontologies pour la représentation de la sémantique et ainsi assuré l'interopérabilité des systèmes. De plus, le cloud computing peut offrir une très puissante, prévisible, fiable et évolutive infrastructure pour l'exécution des systèmes de sécurité qui est mise en œuvre complexe d'applications basées sur le contexte de l'utilisateur.

Nous avons présenté les concepts de base manipulés dans notre projet à savoir: les ontologies, la sensibilité au contexte et l'informatique en nuage, ce travail ce porte sur la confiance dans l'internet des objets, de ce fait dans le prochain chapitre, nous allons décrire en détail les notions liées à l'internet des objets.

CHAPITRE 2

LA SECURITE DANS L'INTERNET DES OBJETS

2.1 Introduction

L'Internet des objets est un sujet passionnant et de plus en plus d'actualité. Tous les objets qui nous entourent sont amenés à se connecter et à devenir ainsi des acteurs à part entière. Sans qu'on s'en rende forcément compte, ils font déjà partie intégrante de notre quotidien. L'Internet des objets est sans aucun doute le nouveau paradigme auquel tous les individus sont appelés à se confronter.

On présenté dans ce chapitre l'Internet des objets, nous commençons par une brève définition de l'Internet des objets et sa naissance, Ensuite, nous présentons sa technologie de base telle que RFID, NFC, nous concluons ce chapitre par les enjeux de sécurité.

2.2 Internet du futur

L'architecture de l'internet du futur [53] est constituée d'un noyau et de deux anneaux : le noyau sera composé de l'évolution de l'infrastructure actuelle de l'Internet. Le premier anneau sera composé d'une nouvelle génération de terminaux avec des capacités de mise en réseau et donc la possibilité de participer à des réseaux spontanés et auto-organisés. Le deuxième anneau, fondé non seulement sur ces systèmes intelligents, actifs et sensibles à leur environnement, mais également sur les technologies sous-jacentes, permettra la fusion des mondes réels et numériques.

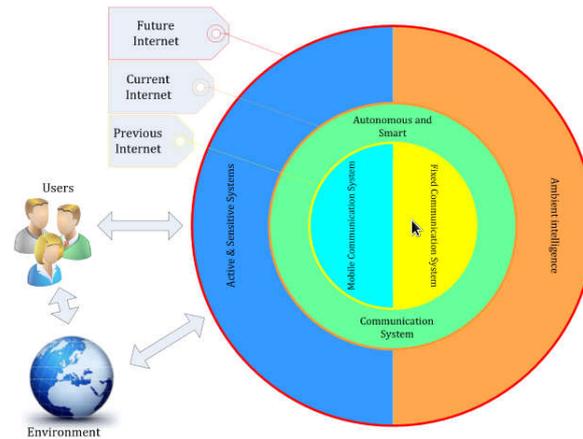


Figure 2.1 : internet du futur [53].

2.3 Informatique ubiquitaire

Le concept d'informatique ubiquitaire ou informatique ambiante a été développé par Mark Weiser [45] au cours des années 80 à Xerox Parc. Il revient sur l'évolution de l'informatique depuis les années 60, caractérisée par trois ères :

- **Les mini-ordinateurs** : une unité centrale partagée par un ensemble d'utilisateurs.
- **Les ordinateurs personnels** : une unité centrale par utilisateur.
- **La mobilité** : plusieurs unités centrales par utilisateur, ces unités centrales pouvant le suivre dans ses déplacements.

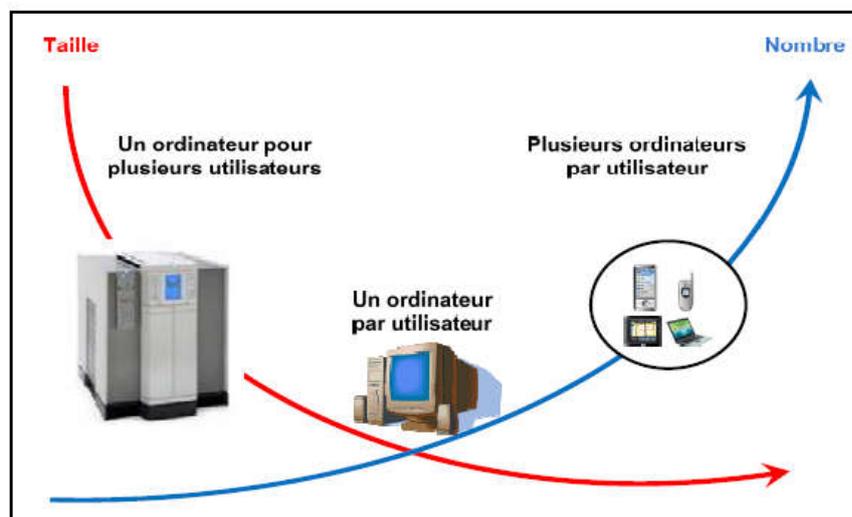


Figure 2.2: Évolution du marché mondial des ordinateurs [48].

L'informatique ubiquitaire peut être vue comme l'opposé de la réalité virtuelle. La réalité virtuelle met une personne à l'intérieur d'un monde créé par l'ordinateur. L'informatique ubiquitaire a pour but de permettre à l'ordinateur de "vivre" dans le monde des hommes et de s'y intégrer au point de disparaître [55].

2.4 Intelligence Ambiante

L'Intelligence Ambiante (Ambient Intelligence) se situe au carrefour de l'Informatique Ubiquitaire et de l'Interaction Homme-Machine. Avec de plus en plus d'objets communicants dans son environnement immédiat, l'Homme doit être capable d'interagir de manière instinctive avec toute l'informatique qui l'entoure. D'où l'intérêt majeur d'avoir des objets « conscients » de leur état et de ce qui les entoure : pouvoir exploiter cette information pour améliorer l'expérience de l'utilisateur [55].

2.5 Internet des objets

D'après [44] Le terme « Objet » désigne une entité (une chose) définie dans un espace à trois dimensions, soit naturelle, soit fabriquée par l'homme (un artéfact), qui a une fonction précise, désignable par une étiquette verbale (un nom). En ce sens, l'objet est sensible, c'est-à-dire qu'il est ou doit pouvoir être perceptible par au moins un des cinq sens ou par un dispositif ad hoc. Il est défini par les relations externes qu'il entretient avec son environnement, par son état et les mouvements ou modifications qu'il subit où qu'il cause. De ce fait, puisque rien n'est permanent, il évolue dans le temps.

De même, par « Internet » on fait référence au réseau mondial de réseaux d'ordinateurs interconnectés basé sur un protocole de communication standard, la suite Internet (TCP/IP), l'Internet des Objets désigne un réseau mondial d'objets interconnectés uniquement adressables, qui repose sur des protocoles standards de communication.

Donc, sémantiquement l'Internet des objets, comme son nom l'indique, a pour but de connecter des objets physiques ou bien des lieux entre eux et avec Internet. Il existe plusieurs types de définition de l'IDO, suivant que l'on s'intéresse aux aspects techniques, aux usages ou bien aux fonctionnalités. Une définition de Pierre-Jean Benghozi et Sylvain Bureau synthétise l'ensemble [56] : « un réseau de réseaux qui permet, via des systèmes d'identification électronique normalisés et unifiés, et des dispositifs mobiles sans fil, d'identifier directement et sans ambiguïté des entités numériques et des objets physiques et ainsi de pouvoir récupérer, stocker, transférer et traiter, sans discontinuité entre les mondes physiques et virtuels, les données s'y rattachant. ».

Certains définissent l'IdO comme des [32] « objets ayant des identités et des personnalités virtuelles, opérant dans des espaces intelligents et utilisant des interfaces intelligentes pour se connecter et communiquer au sein de contextes d'usages variés ».

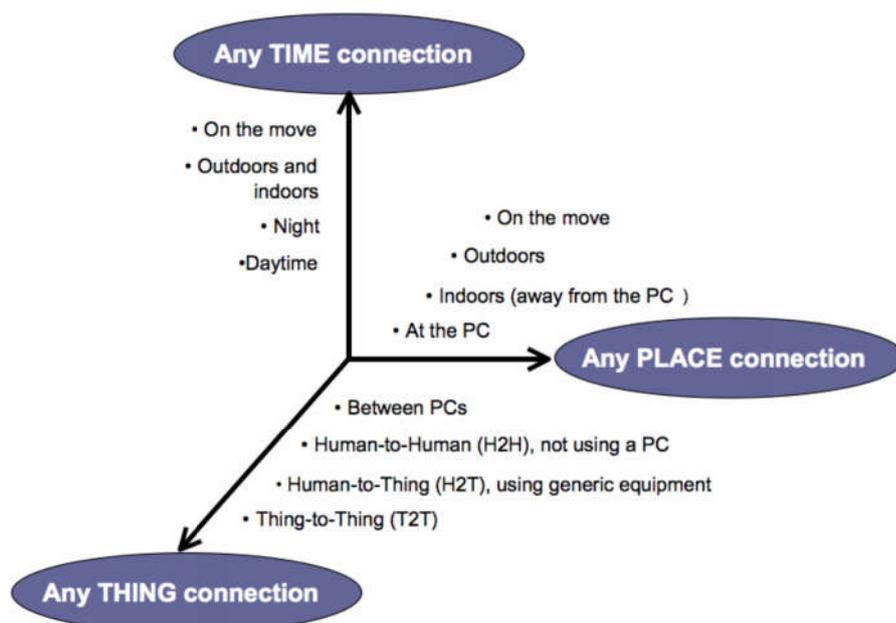


Figure 2.3 : objet dans l'internet des objets [32].

2.6 Historique

L'Internet des objets est né du concept de relier des objets entre eux via un réseau et des identifiants et d'étudier leur comportement. Des systèmes d'identifications étaient déjà bien en place avant l'IDO (le code à barres par exemple). D'ailleurs, la technologie à l'origine de l'identification par radiofréquence (RFID) existerait depuis le milieu des années 1920. Pour rendre le processus d'identification plus efficace des chercheurs ont décidé d'utiliser le RFID. Le RFID a été l'un des systèmes le plus utilisé pour connecter des objets, les premières approches de l'IDO se sont concentrées sur des besoins logistiques (gestion des stocks, traçabilité de produits...).

En 1984, Ken Sakamura a contribué à la création et au développement de l'internet des objets, par la création du premier système d'exploitation libre "Tron" qui était un grand succès vu qu'il s'enfouit dans toutes sortes d'objets physiques autres que les ordinateurs. Le professeur Sakamura lance par ce travail l'ère de l'informatique ubiquitaire que, quatre années plus tard, le professeur Mark Weiser, qui est devenu le père de l'informatique ubiquitaire, a posé ses principales bases théoriques: "L'ordinateur devrait être un prolongement de l'inconscient de l'homme", chose qui a donné naissance à l'internet des objets.

C'est en 1998 que Kevin Ashton, va utiliser la notion d'Internet des objets pour la première fois en déclarant : "si nous parvenons à ajouter l'identification par radiofréquence et d'autres capteurs aux objets de la vie quotidienne, nous pourrions alors créer un Internet des Objets et poser les fondations d'une nouvelle ère de la perception par les machines"

[14].

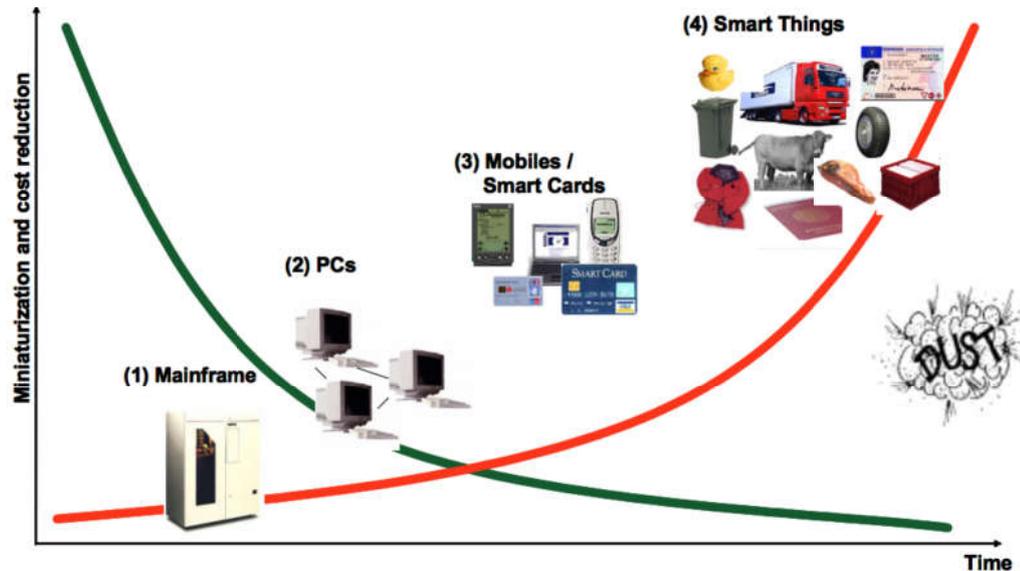


Figure 2.4 : historique d'IdO [32].

2.7 Technologies de base

2.7.1 Les Réseaux de capteurs

Les réseaux de capteurs sont des systèmes qui regroupent plusieurs capteurs afin de couvrir une zone cible. Cette zone peut être géographique ou délimitée par un système plus ou moins étendu : un ouvrage d'art, un ensemble mécanique, un outillage, un réseau télécoms, ...etc. Les réseaux de capteurs connaissent de multiples applications telles que la métrologie environnementale, urbaine ou industrielle.

Les capteurs sans fil communiquent par le biais des ondes radioélectriques. N'étant pas intégrés à un réseau préexistant; les capteurs communiquent grâce à un réseau dit « ad hoc », capable de s'organiser sans infrastructure définie préalablement. Ceci implique que chaque capteur puisse retransmettre une information indépendamment ou avec l'aide des autres capteurs et ceci afin d'envoyer l'information à une « station de base » capable de transmettre l'information à l'utilisateur final, par le biais d'Internet ou d'un réseau télécom GSM dans la majorité des cas.

Les capteurs sont capables de mesurer des grandeurs physiques, chimiques ou biologiques, de traiter ces informations et de les stocker. Ils sont alimentés électriquement via une batterie individuelle optimisée pour des tâches comme le traitement de l'information et la communication [9].

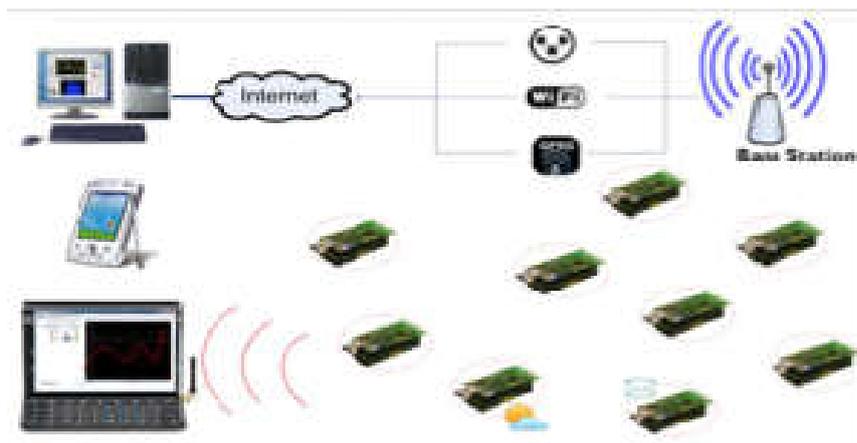


Figure 2.5 : Exemple de représentation d'un réseau de capteurs [9].

Il existe 3 principaux systèmes d'étiquetage pour l'IdO. Les étiquettes peuvent être physiques ou virtuelles et elles permettent d'associer l'objet au dispositif avec lequel il communique. Nous allons maintenant les voir.

2.7.2 RFID (Radio Frequency Identification)

Une des technologies clés de l'Internet des objets est l'identification par radiofréquence abrégée en anglais RFID «Radio Frequency Identification». La RFID est une technologie bien connue qui est employée avec succès dans de nombreuses applications. Elle a été introduite dans les années 50, et présente des avantages par rapport à son prédécesseur, le code de barres, car elle nécessite de petits dispositifs électroniques pour transmettre un code d'identification (ID) et ne nécessite pas de ligne de vue pour fonctionner. Cette technologie propose un ensemble de solutions qui permettent le suivi et la traçabilité des personnes, des animaux et pratiquement n'importe quel objet en utilisant des liaisons sans fil.

L'architecture RFID est composée d'étiquettes RFID dites Tags, qui fournissent un code d'identification, de lecteurs RFID, qui interrogent et obtiennent grâce aux ondes électromagnétiques ce code, et un logiciel médiateur («Middleware») qui fournit une plateforme pour interpréter et utiliser l'information recueillie [54].



Figure 2.6: RFID architecture [54].

2.7.3 NFC (Near Field Technologie)

La technologie NFC (Near Field Technologie) comme son nom l'indique, permet de communiquer des informations sans contact entre l'émetteur et le récepteur. À l'inverse de la RFID, c'est une technologie sans fil de courte portée (elle émet jusqu'à 10 centimètres environ). La démarche de l'utilisateur est donc nécessaire, il est difficile de connecter un objet à son insu [54].

2.7.4 Le code-barre à deux dimensions (QR Code)

Pour pallier au faible volume de données inscriptibles dans un code-barres à une dimension, des codes à deux dimensions sont inventés en 1994.

Les QR codes se présentent sous la forme de modules noirs disposés dans un carré à fond blanc et peuvent contenir jusqu'à 4296 caractères, C'est le code bidimensionnel le plus répandu dans le monde en 2010[52].



Figure 2.7 : Un exemple de QR Code (site web université de BLIDA).

2.8 Applications

Aujourd'hui de nombreux secteurs sont concernés par l'Internet des objets. Parmi les principaux on peut citer : l'énergie et le développement durable, la logistique, la domotique, les transports, la téléphonie mobile, la distribution, le service à la personne et la sécurité. Nous allons voir ici les principaux secteurs [14].

2.8.1 L'énergie et le développement durable

Aujourd'hui, l'une des préoccupations majeures de nos sociétés est l'économie d'énergie, autant pour des raisons écologiques que financières.

Les professionnels de l'aménagement urbain parlent de "villes intelligentes". C'est le fait d'améliorer l'aménagement urbain et les services d'une ville, afin d'élever sa qualité de vie, sa durabilité ou son équité sociale grâce à l'innovation technologie et le design.

2.8.2 La domotique

L'internet des objets joue un rôle important dans l'amélioration de la domotique. Elle permet d'améliorer le confort (réglage de la température, optimisation de l'éclairage...), la

communication envers les équipements (commandes à distance, signaux sonores...) ou encore la sécurité (système d'alarme).

2.8.3 La santé

Le secteur de la santé est devenu un des domaines les plus étudiés de l'Internet des objets. Aujourd'hui, grâce à l'Internet des objets il est possible de maintenir les personnes âgées à leur domicile tout en ne négligeant pas leur sécurité ou le lien social.

2.8.4 Les transports

Avec l'Internet des objets, les véhicules vont pouvoir communiquer avec de nombreux acteurs. Ils pourront tout d'abord communiquer avec des infrastructures telles que des péages, des parkings, les flux de circulation. Le conducteur saura quelle route prendre selon le trafic, où il va pouvoir se garer, payer sans contact le péage. En termes d'environnement, il sera possible de calculer précisément les émissions polluantes du véhicule. L'Internet des objets pourraient aussi favoriser le développement de l'auto-partage, grâce à une gestion simplifiée par Smartphone des étapes comme l'identification de la personne ou le paiement.

2.9 Protection de la vie privée

La protection de la vie privée (privacy) est probablement le problème de sécurité principal, mais aussi nouveau, parmi de nombreux chercheurs. Westin [9] a défini la protection de la vie privée comme "le droit pour les individus, groupes, ou institutions, de déterminer pour eux-mêmes, quand, comment, et quelle quantité d'information les concernant est communiquée aux autres".

2.10 La sécurité et la sensibilité au contexte

Il existe une relation étroite entre la sécurité informatique et la prise en compte du contexte [59]. La sécurité (confidentialité, authenticité) des informations contextuelles est en partie fournie par la prise en compte du contexte dans le contrôle d'accès. La sensibilité au contexte et le contrôle d'accès sont deux concepts similaires. Ils passent tous les deux par une prise de décision en fonction des informations collectées en entrée, pour fournir en sortie une application dont certaines fonctionnalités peuvent être activées et autorisées ou inactivées et interdites.

Thomas et Sandhu [59] identifient la prise en compte du contexte dans le processus de contrôle d'accès comme un point déterminant pour les futurs systèmes d'informatique. Ces nouvelles informations augmenteront ou remplaceront les attributs actuellement utilisés pour

rendre le contrôle d'accès moins intrusif.

2.11 Sécurité dans l'internet des objets

D'après [18], ils ont défini les défis de sécurité dans l'internet des objets par la confidentialité des données, intimité et confiance.

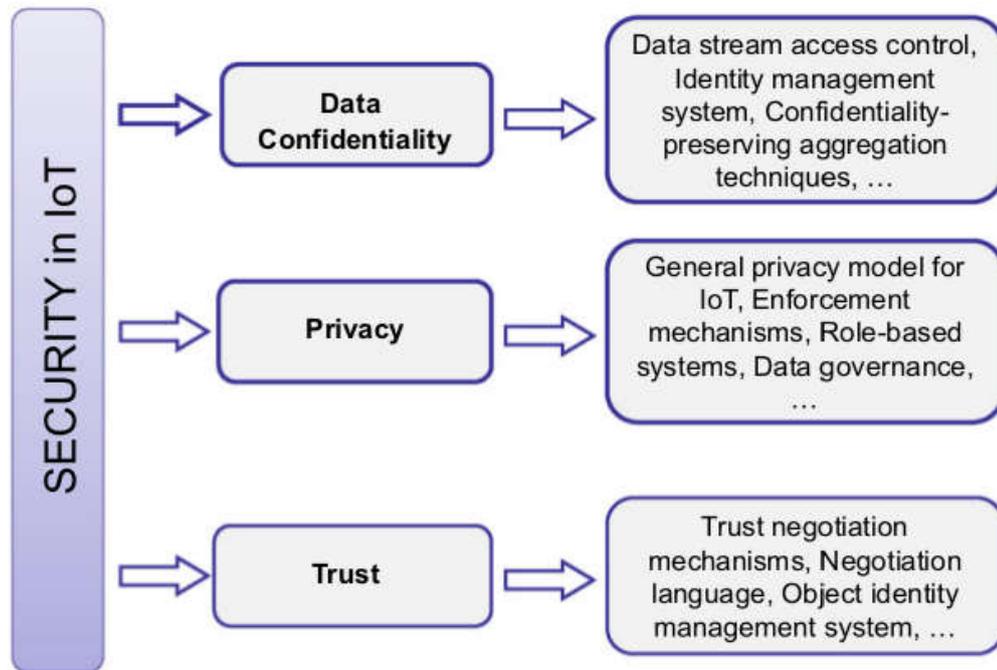


Figure 2.8 : représentation graphique des défis de sécurité dans l'IdO [18].

Confidentialité des données

La confidentialité des données est un enjeu fondamental dans les scénarios de l'internet des objets, indiquant la garantie que seules entités autorisées peuvent accéder et modifier des données. Ceci est particulièrement pertinent dans le contexte de l'entreprise, tandis que les données peuvent représenter un atout pour être protégées afin de préserver la compétitivité et la valeur de marché. Dans le contexte de l'IdO non seulement des utilisateurs, mais aussi des objets autorisés peuvent accéder aux données. Il faut pour cela examiner deux aspects importants : d'abord, la définition d'un mécanisme de contrôle d'accès et d'autre part, la définition d'un processus d'authentification objet (avec un système de gestion des identités connexes).

Les principaux défis de la recherche pour assurer la confidentialité des données dans l'IdO, se rapportent à :

- Définition des mécanismes adéquats pour contrôler l'accès aux flux de données générées par des dispositifs d'IdO.

- Définition d'un langage de requête correspondant pour les applications permettant de récupérer les informations souhaitées sur un flux de données.
- Définition d'un système intelligent de gestion de l'identité des objets

Intimité

Définit les règles selon lesquelles les données relatives aux utilisateurs individuels peuvent être accessibles. Les principales raisons qui font de la vie privée une exigence IdO fondamentale résident dans les domaines d'application envisagés par IdO et les technologies utilisées.

Les défis de la recherche en termes de mécanismes de protection des renseignements personnels pour IdO, sont donnés par :

- Définition d'un modèle général de la vie privée dans l'IdO.
- Développement des techniques d'application innovantes, capables de soutenir l'ampleur et l'hétérogénéité qui caractérise les scénarios de l'IdO.
- Développement de solutions qui concilient la nécessité de l'anonymat présenté par certaines applications avec les exigences de localisation et de suivi de certains autres. Cela implique la définition des politiques de confidentialité, qui spécifient les conditions dans lesquelles il est possible d'identifier et de localiser un objet intelligent. En outre, il convient de préciser quand il est possible d'accéder à des données sensibles.

Confiance

Confiance faite référence aux politiques de sécurité régissant les accès aux ressources et aux titres de compétences qui sont nécessaires pour satisfaire ces politiques. Négociation de confiance se réfère au processus d'échanges d'informations d'identification qui permet à une partie qui demande un service ou d'une ressource à une autre partie de fournir les informations d'identification nécessaires pour obtenir le service ou la ressource. Cette définition de la confiance est très naturelle pour la gestion des connaissances sécurisées que les systèmes peuvent avoir à échanger des informations d'identification avant le partage des connaissances. Pour cette raison, ils ont fondons leur analyse des problèmes de confiance en l'IdO sur elle.

Les défis de la recherche les plus pertinents dans la définition de mécanismes de confiance appropriée pour IdO, peuvent être résumés comme suit :

- Introduction d'une langue de négociation simple en charge les exigences

d'interopérabilité sémantique de l'IdO.

- Définition d'un mécanisme de négociation basée sur un contrôle d'accès fine des flux de données.
- Développement d'un système de gestion de l'identité d'objet.
- Conception d'un cadre de gestion de la confiance générale en mesure de tirer parti des éléments susmentionnés.

D'après [63] Les recherches censées se concentrent sur :

- sécurité des protocoles des réseaux
- Les données et la vie privée
 - intimité par la conception
 - Transparence
 - La gestion des données
- La gestion des identités
- Confiance et gouvernance
- La tolérance aux pannes
- Cryptographie et protocoles
- Identité et la propriété
- protection de la confidentialité

2.11.1 Modèles de sécurité proposée pour l'internet des objets

Dans [36] les 3 éléments essentiels dans un système d'information traditionnel, sont communication, control et calcul

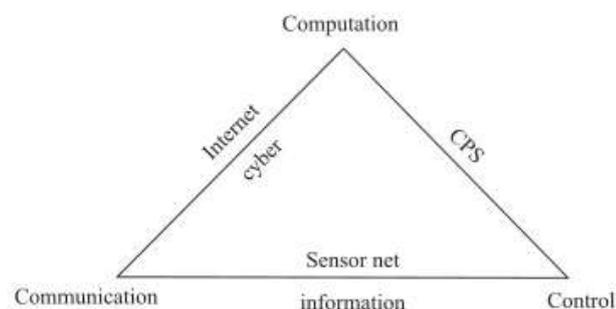


Figure 2.9 : 3C modèle pour IdO [36].

Entre la communication et le contrôle, il Ya un lien, qui est parfois appelé «l'information», est parfois appelé «réseau de capteurs», signifie que pour obtenir des informations à partir de l'extrémité de commande et puis le mettre dans la fin de la communication. Il existe un autre lien entre la communication et de calcul aussi.

Théoriquement, il n'existe aucun lien entre le contrôle et le calcul parce que le contrôle dépend de l'intervention par des personnes. Le lien entre le calcul et le contrôle est en fait le pont reliant couche de données avec la couche physique. Ce pont est appelé "CPS".

Bien que nous disions toujours « pas une invention des utilisateurs dans la sécurité », il est nécessaire de mettre certains "contrôle de sécurité" dans le milieu, il est besoin de quelque chose dans le milieu en tant que garde de sécurité. Le modèle de sécurité montré dans Figure 2.10 règle ce problème

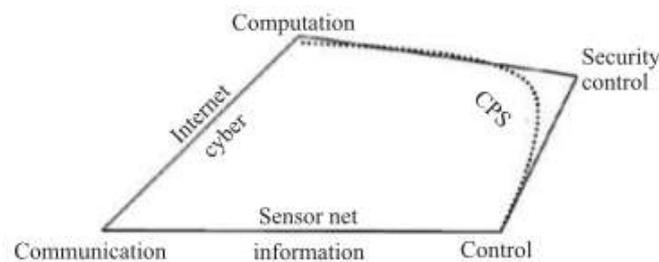


Figure 2.10 Sécurité modèle pour l'IDO [36].

L'IdO est un système complexe, impliquant capteurs, réseaux de capteurs, Internet, logiciels et d'autres aspects. Afin d'atteindre un haut niveau de sécurité du système IdO, dans [36] ils sont mentionnés que l'orientation de la recherche sur la sécurité de l'IdO doit être élaborée à partir des aspects suivants figure 2.11

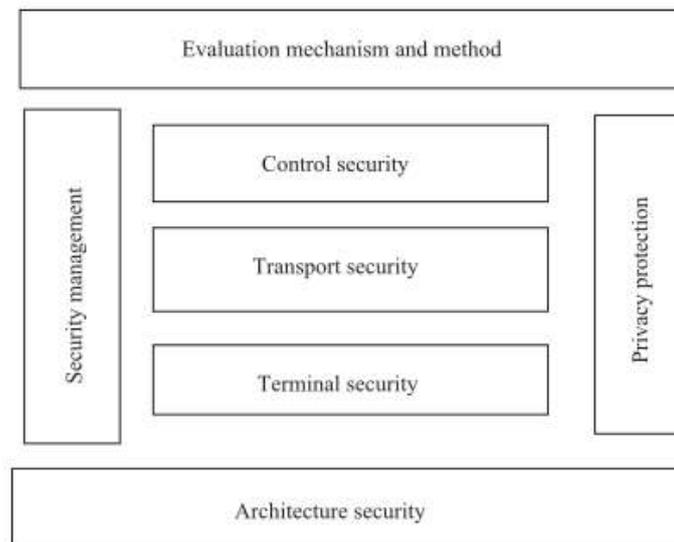


Figure 2.11 : sécurité dans IoT [36].

Dans [36], ils ont donné la direction des recherches détaillées sur les technologies clés des sept aspects suivants :

Sécurité de l'architecture

Les recherches censées se concentrent sur :

- La sécurité du système de l'identité RFID dans l'IdO;
- Le mécanisme d'assurance de sécurité de haute fiabilité dans l'IdO;
- Le modèle de confiance et la méthode de l'IdO.

Sécurité du terminal

Les recherches censées se concentrent sur :

- L'intégrité des données (fiabilité);
- La confidentialité et la disponibilité;
- certifications des tags;
- Recherche sur l'algorithme de sécurité sur les entités de capteurs sous la contrainte énergétique.

Sécurité du transport

Les recherches censées se concentrent sur :

- La sécurité des données dans le réseau de capteurs;
- sécurité de la Réseau de transport
- fiabilité de Sécurité de la transmission des données des les terminaux mobiles.

Sécurité du contrôle

Les recherches censées se concentrent sur :

- Les mécanismes de sécurité d'un comportement de contrôle de l'IdO;
- Contrôler le comportement qui peut être attendu.
- Lire l'information sur le contrôle.

Protection de la confidentialité

Les recherches censées se concentrent sur :

- sécurité de la technologie du terminal intelligent;
- amélioration du Technologie de confidentialité;
- technologie du contrôle d'accès

Gestion de la sécurité

Les recherches censées se concentrent sur :

- Recherche pour l'authentification, l'identification et le critère de fournisseur unique de services de TI
- Gestion des clés.

Méthode et Mécanisme d'évaluation

La recherche est censée se concentrer sur :

- Le niveau de protection de sécurité orienté vers l'IdO;
- L'évaluation de la sécurité orientée vers l'IdO,
- L'évaluation des risques orientée vers l'IdO.

Le modèle de la sécurité et la vie privée dans l'internet des objets proposé dans [63] est basé sur les empreintes digitales ou la mesure d'une caractéristique d'identification d'un individu ou d'un objet physique ou numérique, le modèle aussi distingue entre les appareils mobiles et statiques .ce modèle est basé sur deux principes essentiels.

- chaque élément est connecté et capable de communiquer avec un autre élément
- chaque fonctionnalité « commonplace » sera omniprésente.

2.11.2 Architectures proposés pour l'IdO

Hydrogen

Hydrogen [31] est une architecture et un Framework développés pour appuyer la sensibilité au contexte. C'est une architecture en trois couches pour convenir aux besoins particuliers des équipements mobiles composés des couches Figure 2.12 : adaptateur, gestion (serveur de contexte) et application. Le serveur de contexte contient toutes les informations captées par la couche adaptateur. Il fournit à la couche application, ou à d'autres équipements, le contexte requis en utilisant le modèle de communication peer-to-peer.

L'approche Hydrogen considère le contexte comme toute information pertinente sur l'environnement d'une application et décrit le contexte par un modèle orienté objet.

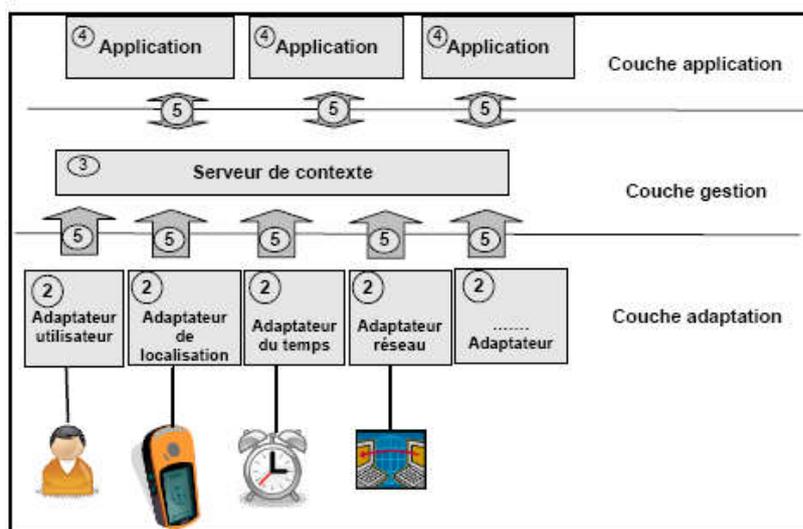


Figure 2.12 : Architecture de Hydrogen [31].

DOG

Le système DOG [11] ou Domotic Home Gateway a été conçu pour transformer les systèmes domotiques en systèmes intelligents. Il permet entre autres:

- L'interopérabilité entre les divers réseaux domotiques à travers des pilotes spécifiques.
- La modélisation ontologique de l'espace intelligent et de ses dispositifs.

DogOnt est l'ontologie sur laquelle se base le système DOG. Cette ontologie est composée des concepts suivants:

- **BuildingEnvironment**: modélise les chambres et les espaces composant un bâtiment.
- **BuildingThing**: modélise les composants du bâtiment, ceux-ci appartiennent à deux catégories:
 - **Controllable**: modélise les éléments contrôlables divisés en 2 catégories :
 - Les dispositifs: il y a les dispositifs bruns (BrownGoods) comme la télé, le HIFI, et les dispositifs blancs (WhiteGoods) comme le réfrigérateur, la lave-vaisselle.
 - Les installations: comme l'installation électrique, la climatisation, le système d'alarme.
 - **Uncontrollable**: éléments non contrôlables comme les meubles et les éléments architecturaux: les murs, la terrasse, le plafond.
- **Functionality**: modélise les fonctionnalités qui sont de 3 types:
 - Les fonctionnalités de contrôle: chaque dispositif contrôlable a une fonctionnalité de contrôle qui définit un ensemble de commandes ainsi que leurs portées.
 - Les fonctionnalités de notification: définit les notifications possibles relatives à l'état du service.
 - Les fonctionnalités de requête: chaque dispositif contrôlable peut être interrogé sur son état.

En plus, les fonctionnalités peuvent être de deux types:

Continues: les propriétés du service changent d'une manière continue (ex : lumière)

Discrètes: les propriétés changent d'une manière brusque (ex : switch on/off).

- **State**: modélise l'état d'un service fourni par un dispositif, il peut être continu ou discret, chaque état a une valeur (stateValue).
- **Command**: modélise les commandes fournies par chaque fonctionnalité.
- **DomoticNetworkComponent**: Les composants du réseau domotique (gateways).

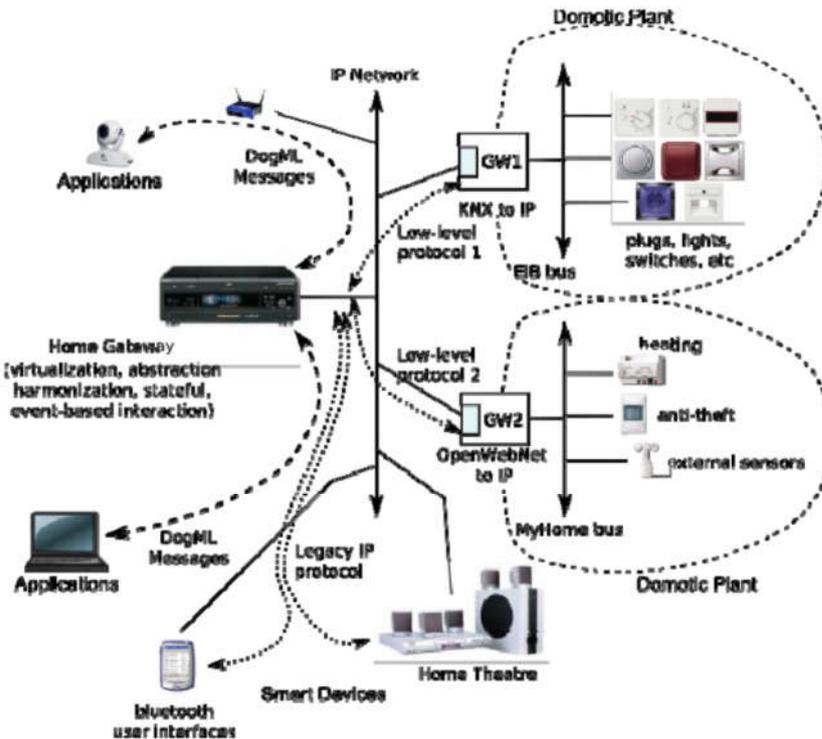


Figure 2.13 : Environnement Domotique Intelligent [11].

CORTEX

Biegel et Cahill [10] ont proposé un Framework pour faciliter le développement des applications mobiles sensibles au contexte dit CORTEX (COoperating Real-time sentTientobjects: architecture and EXperimentalevaluation). L'architecture est basée sur les «sentientobjects ». Elles ont des caractéristiques avantageuses pour un environnement informatique pervasif : (i) Elles sont douées de sensation en ayant la capacité de percevoir l'état de l'environnement par des capteurs ;(ii) Elles sont autonomes en ayant la capacité de fonctionner indépendamment du contrôle humain d'une manière décentralisé ; (iii) Elles sont proactives en prenant des initiatives pour accomplir un but proposé.

L'architecture d'un « sentientobject » Figure 2.13 est composée de deux interfaces : capture des événements perçus par les capteurs, émission des événements pour s'adapter au contexte actuel. Cette architecture contient un module pour la fusion du contexte et son interprétation dans un haut niveau d'abstraction. Elle contient également, un module pour la représentation hiérarchique du contexte dont le but est de limiter le contexte de la situation actuelle et par la suite de limiter les actions possibles. Un moteur d'inférence pour spécifier le

comportement de l'application à un contexte donné qui utilise le modèle d'exécution « événement-condition-action ».

La communication entre les différents «sentientobject », capteurs et actionneurs qui composent le système utilise le mécanisme basé sur l'événement qui s'établit d'une manière dynamique pendant le fonctionnement du système. Le système d'inférence est écrit en CLIPS. Cette architecture est une solution ad hoc pour un réseau mobile.

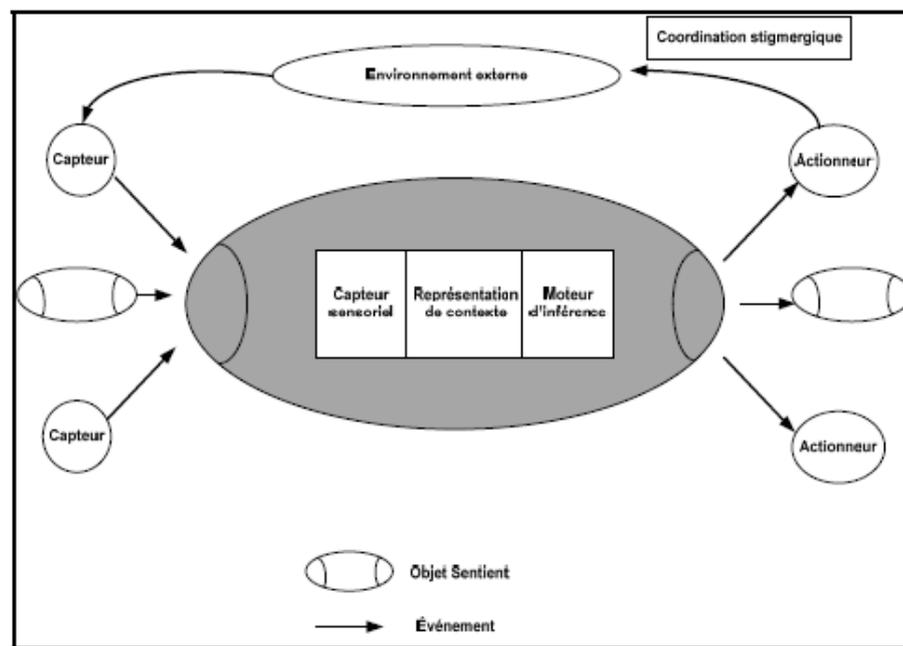


Figure 2.14 : Architecture d'un « sentient object » [10].

FI-Ware

FI (Future Internet)-Ware [23] est un projet européen dont le but est de fournir le framework au coeur d'une plate-forme commune pour l'Internet des objets. Ce Framework est constitué d'un entrepôt de service, d'un analyseur sémantique, et permet à des fournisseurs d'offrir des services, et aux utilisateurs de les consommer et de les déployer sur leurs objets.

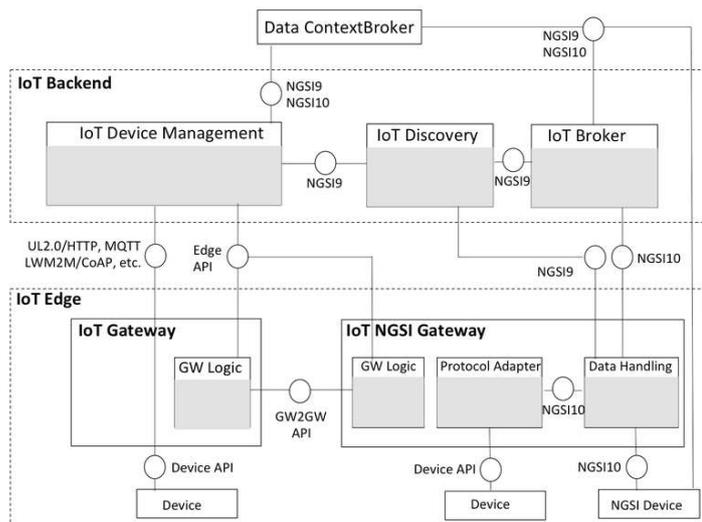


Figure 2.15 : Architecture globale de FI-Ware [23].

Cognitive Management Framework

V. Foteinos et al. [67] Proposent un Framework spécialisé dans l'IdO auquel ils ajoutent une dimension sémantique plus poussée, notamment en employant un moteur d'inférence chargé d'adapter les ressources selon le contexte, la fiabilité et la disponibilité des éléments, et d'en masquer l'hétérogénéité. Leur solution manipule des objets virtuels correspondant aux besoins de l'utilisateur, et tente dans les couches plus basses de résoudre cette virtualisation en recherchant les objets réels capables d'y répondre. L'ensemble fournit une approche de haut niveau, mais principalement en direction des utilisateurs, sans offrir une expressivité pour les interactions d'objets entre eux.

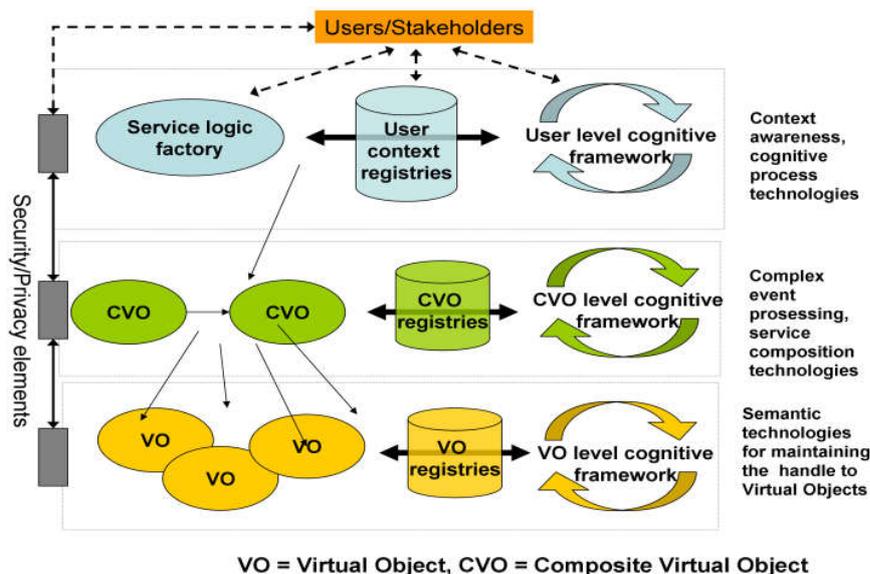


Figure 2.16 : Architecture de Cognitive Management Framework [67].

Les travaux existants dans le domaine de l'internet des objets se sont focalisés sur tous les aspects de capture, interprétation, stockage et dissémination du contexte de l'utilisateur. Cependant, il y a beaucoup moins de travaux qui s'intéressent à la sécurité et le respect de la vie privée, les utilisateurs doivent avoir le contrôle sur les informations qui les concernent. La manière dont les informations sont utilisées doit être clairement identifiée et les utilisateurs peuvent seulement récupérer les informations. Ce travail se focalise spécialement sur cet aspect.

2.12 Conclusion

Dans ce chapitre nous avons présenté les concepts de base concernant l'internet des objets, en commençant par une brève description, avec un brèf historique et aussi les technologies de base de l'internet des objets tels que RFID,NFC,QR code, et les enjeux de sécurité associé.

A la fin de ce chapitre, nous avons présenté quelques travaux de recherche récents qui ont proposés des modèles intéressants pour la sécurité des utilisateurs dans l'internet des objets.

Le chapitre suivant est dédié à la conception de notre propre approche de sécurité dans l'internet des objets. L'approche proposée a comme but de préserverla vie privée des utilisateurs.

CHAPITRE 3

CONCEPTION DE L'ARCHITECTURE SENSIBLE AUX CONTEXTE POUR IoT

3.1 Introduction

Dans ce chapitre nous allons voir la conception de notre système ; nous commençons par la présentation de l'architecture en trois couches pour l'internet des objets ensuite, nous présentons la description de l'architecture globale proposée et son fonctionnement ainsi que le rôle de chacun de ces composants pour garder la vie privée dans l'internet des objets. Nous terminons ce chapitre par une conclusion.

3.2 Architecture proposée

Nous présentons une infrastructure inspirée des concepts présentés dans l'état de l'art et composée de trois principaux niveaux. Basé sur le modèle architectural pour IOT présenté dans [41], 1) couche de capteur 2) couche réseau 3) couche d'application, notre ontologie des contextes et des politiques de confidentialité est sauvegardée dans le Cloud et accessible par la technologie (software As A Service), l'utilisateur enregistre ses profils de sécurité dans l'ontologie et les objets connectés avec l'utilisateur en prenant ces profils en considération par la connexion avec l'ontologie comme un service.

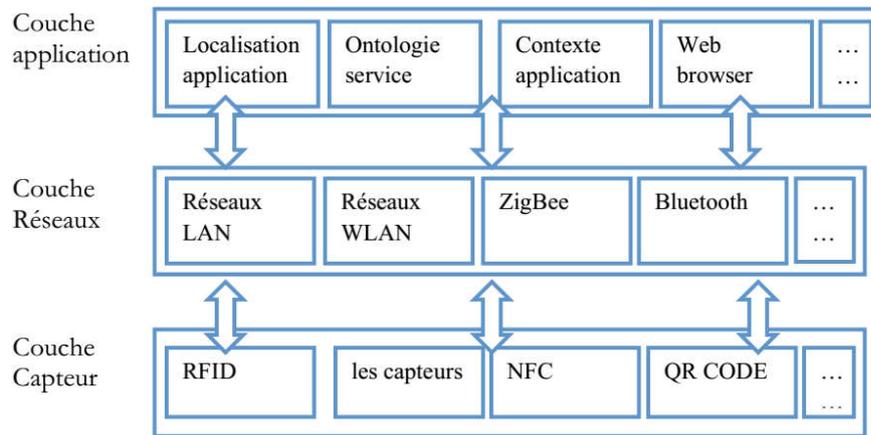


Figure 3.1 : les trois couches de l'IoT [41].

3.2.1 Couche de capteur

C'est la couche qui contient toutes les sources de données qui peuvent fournir des informations utiles pour l'ontologie du contexte. Un capteur est une source matérielle ou logicielle qui peut générer une information contextuelle.

On peut distinguer trois types des capteurs [46] :

- 1) capteurs physiques : Les capteurs physiques sont des dispositifs matériels qui sont capables de fournir des données de contexte. Sont plus fréquents et disponibles et fournissent généralement des données brutes, le plus important capteur est celui de la localisation (GPS) mais on peut inclure aussi d'autres capteurs physiques.
- 2) capteurs virtuels : Les capteurs virtuels fournissent des informations contextuelles à partir d'applications ou services logiciels. Par exemple, nous pouvons détecter l'activité de l'utilisateur sur un ordinateur en analysant les événements de la souris ou les saisies à partir du clavier. Les capteurs virtuels sont beaucoup moins coûteux que les capteurs physiques puisqu'ils sont basés sur des composants logiciels qui sont généralement moins chers que des appareils électroniques.
- 3) Les capteurs logiques : Ce type de capteurs utilise généralement plusieurs sources d'information contextuelles pour fournir une autre information de synthèse plus précise. Ces capteurs peuvent réutiliser des capteurs physiques et virtuels pour fournir un contexte de plus haut niveau.

Chaque type de capteur doit être attaché à un composant logiciel permettant l'accès aux informations capturées. Ces composants sont généralement fournis avec des pilotes logiciel set une API de communication avec les capteurs. Par exemple, une interface `getPositionOfDevice()`, qui retourne la position d'un capteur en mouvement peut être implémentée en utilisant un système GPS.

Le captage des données peut se faire d'une façon manuelle, c'est-à-dire au moment des prises de mesures faites par l'utilisateur lui-même, ou d'une façon automatique par le réseau de capteurs (localisation GPS, température corporelle, rythme cardiaque). La fréquence du captage automatique est définie et programmée selon les besoins de chaque utilisateur.

3.2.2 Couche réseaux

Les capteurs peuvent être interconnectés par des technologies comme le Zigbee, le Bluetooth et le Wifi. Ces mêmes technologies peuvent être utilisées pour la transmission des données vers l'appareil mobile de l'utilisateur.

Objects as a service

D'après [39] [29] L'intégration de La technologie « RESTFUL » dans IOT nous permet d'accéder aux ressources des objets depuis l'internet, pour intégrer le RESTful dans les objets nous avons deux possibilités :

Intégrer un serveur web dans chaque objet, ce serveur web permet de nous fournir des interfaces et établir une communication depuis internet, ou bien ajouter un middleware (serveur), qui possède une adresse IP et fonctionne sur un serveur Web, et qui comprend les protocoles propriétaires de différents appareils qui y sont connectés grâce à l'utilisation de pilotes dédiés, les demandes de services des appareils sont formulés à partir d'une URL standard.

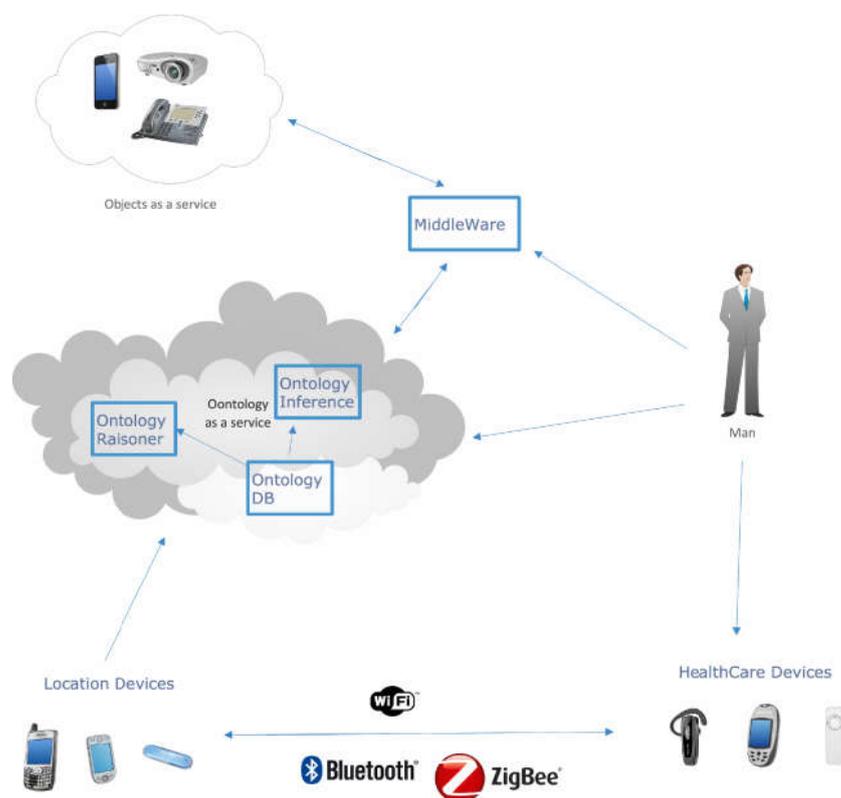


Figure 3.2 : présentation graphique de l'architecture avec middleware.

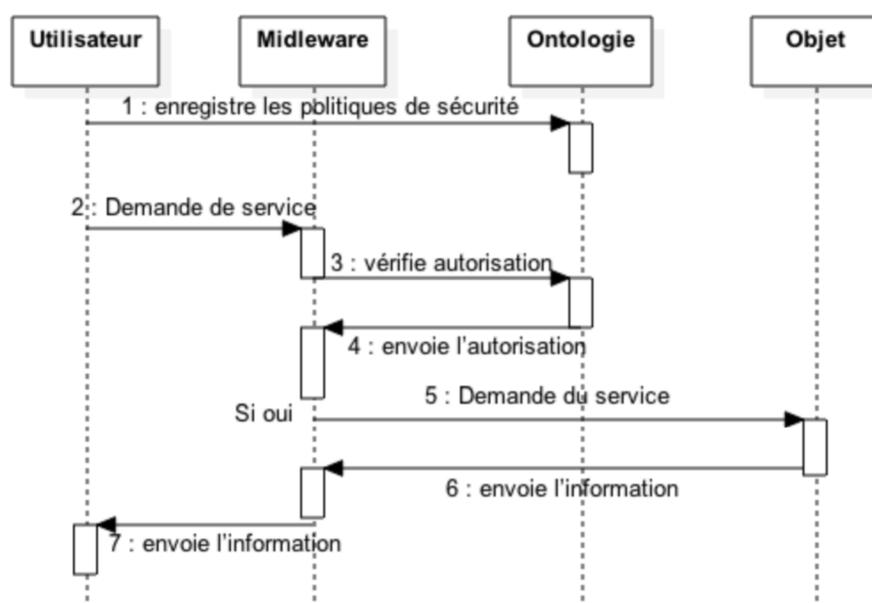


Figure 3.3 : diagramme de séquence de l'architecture avec middleware.

Dans notre architecture

- 1-L'utilisateur enregistre les politiques de Sécurité dans l'ontologie
- 2-L'utilisateur demande un service par l'URL associé a ce service par exemple `https://domicile/cuisine/refrigerateur/température`
- 3-Le middleware vérifie si cet utilisateur est autorisé ou non dans l'ontologie
- 4-Réponse de l'ontologie par autorisé ou non
- 5-Le middleware communique l'appareil associé au service
- 6-L'appareil envoie l'information demandée
- 7-Le middleware donne l'information demandée au l'utilisateur

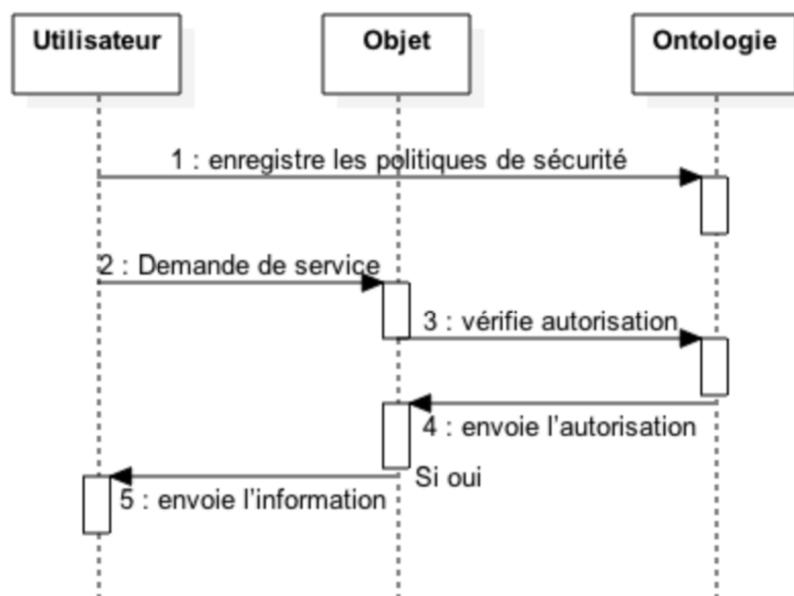


Figure 3.4 : diagramme de séquence de l'architecture sans middleware.

Dans notre architecture sans middleware

- 1-L'utilisateur enregistre les politiques de Sécurité
- 2-L'utilisateur demande un service
- 3-L'objet vérifie si cet utilisateur est autorisé ou non dans l'ontologie
- 4-Réponse de l'ontologie par autorisé ou non
- 5- L'objet envoie l'information demandée au l'utilisateur

3.2.3 Couche application

La couche application est représentée par l'application qui offre ses services aux

différents objets concernés. Cette couche est responsable de l'extraction des informations des différentes sources de données attachées à l'application. Elle doit aussi implémenter les réactions nécessaires aux changements du contexte.

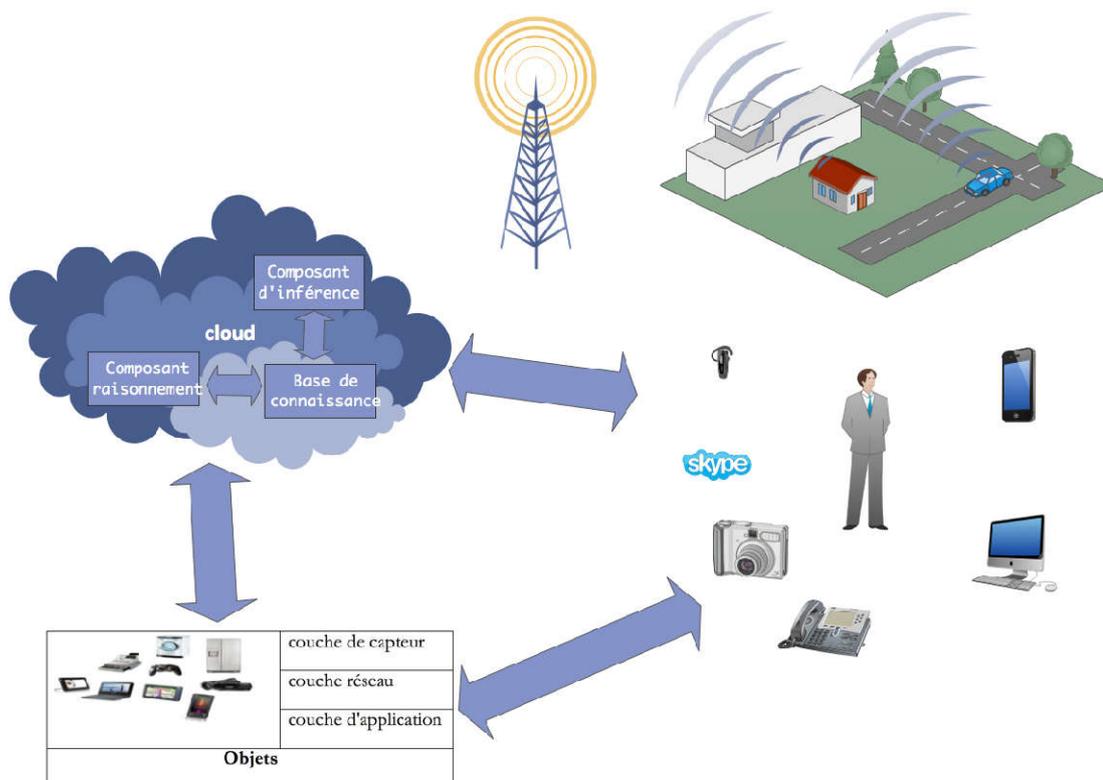


Figure 3.5 : présentation graphique de l'architecture.

A) Les composants du service de l'ontologie

Le premier est le composant de l'inférence, ce composant est responsable d'inférer le contexte récupéré à partir des capteurs en utilisant le raisonnement logique. Elle permet d'élever l'information à un niveau supérieur d'abstraction. Le deuxième composant est la base de connaissance, ce composant assure l'interrogation et la modification des connaissances contextuelles. Elle contient les ontologies du contexte d'un domaine donné et ses instances. Ces ontologies représentent les objets, les relations qui existent entre eux et leurs instances. Le troisième composant est le raisonneur, c'est le composant responsable de l'inférence des informations de contexte de bas niveau (contexte direct) à des informations de contexte de haut niveau de complexité.

Dans ce qui suit, nous illustrons notre contexte par deux ontologies. La première ontologie est « SOUPA » [30] et la deuxième ontologie est appelée « CONON » [71].

3.3 Représentation du contexte

Pour l'approche ontologique, nous pouvons utiliser des ontologies, entières ou partielles, développées précédemment. Les ontologies de haut niveau sont celles qui sont le plus souvent utilisées puisqu'elles des concepts universels, indépendants du domaine et qui sont flexibles à une adaptation pour différentes applications qui ne réalisent pas forcément les mêmes traitements et n'appartiennent pas au même domaine.

3.3.1 Ontologie SOUPA

[30] SOUPA (Standard Ontology for Ubiquitous And Pervasive Applications) a été connue pour modéliser et supporter les applications de l'informatique ubiquitaire. SOUPA est divisée en 2 couches : l'ontologie principale SOUPA-Core qui essaie de définir des vocabulaires génériques qui sont communs à différentes applications ubiquitaires, et les extensions de SOUPA qui étendent l'ontologie principale en définissant des vocabulaires additionnels qui supportent des applications spécifiques.

Parmi les concepts définis par l'ontologie SOUPA on peut citer :

Person : définit un ensemble de vocabulaires pour décrire le profil et les informations personnelles d'une personne.

Policy & Action : définit les vocabulaires pour décrire les stratégies de sécurité et de protection de vie privée.

Agent & BDI : définit les vocabulaires pour représenter les informations sur les agents et leurs états mentaux.

Space : définit les vocabulaires qui représentent les informations spatiales comme les régions géographiques.

Time : définit les vocabulaires pour exprimer le temps et les relations temporelles.

Event : définit les vocabulaires pour décrire les événements.

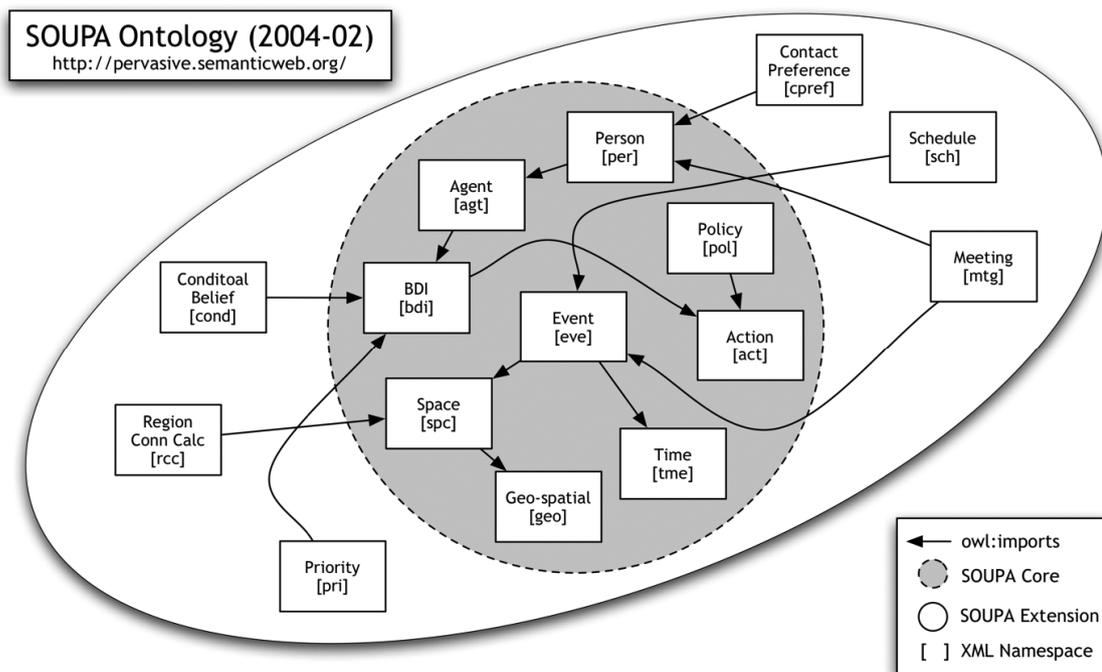


Figure 3.6 : l'ontologie principale SOUPA-Core et extensions [30].

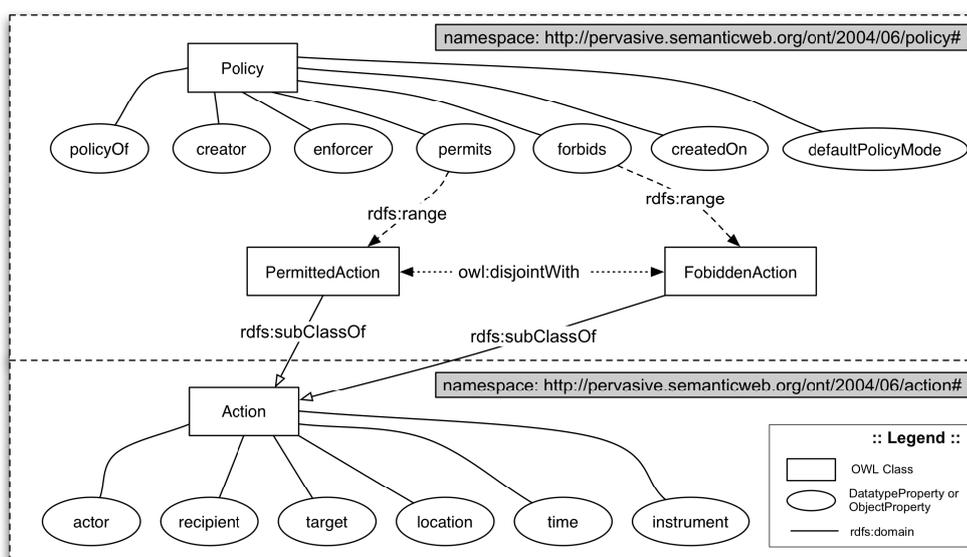


Figure 3.7 : l'ontologie SOUPA-Policy [15].

La classe OWL `pol:Policy` représente un ensemble de tous les politiques de sécurité. Une instance de cette classe représente un politique qui définit les permissions pour une entité pour exécuter des actions différentes, Pour décrire des règles de `pol:Policy`, l'ontologie définit la propriété `pol:permits` pour exprimer les autorisations et la propriété `pol:forbids` pour exprimer les interdictions et le créateur qui est lié avec la class `Person` dans l'ontologie CONON. l'ontologie définit la propriété `pol:defaultPolicyMode`, cette propriété a le domaine `pol:Policy` et image `pol:Mode`, la classe `pol:Mode` est une énumération, qui

compose de deux classes prédéfinies pol:Conservative et pol:Liberal .

La class act:Action représente un ensemble de toutes les actions, dans l'ontologie action de SOUPA, six propriétés d'action sont définies, Ils sont act:actor, act:recipient , act:target qui est lié avec la class Device de l'ontologie CONON , act:location qui est lié avec la class Location (IndoorSpace ou OutdoorSpace), et act:time qui est lié avec la class Time.

3.3.2 Ontologie CONON

CONON : dans [71] les auteurs ont proposé une ontologie du contexte CONON (CONtextONtology) codée en OWL pour la modélisation du contexte dans un environnement pervasif et pour le support de raisonnement logique sur le contexte. CONON fournit une première ontologie du contexte supérieur qui capte les concepts généraux du contexte de base, et fournit aussi l'extensibilité pour ajouter des ontologies spécifique à un domaine particulier d'une manière hiérarchique.

L'ontologie de haut niveau CONON est un modèle générique qui présente un ensemble d'entités de haut niveau et qui offre une forte extensibilité et capacité d'adaptation à différents domaines spécifiques. Les auteurs ont jugé que la localisation, l'utilisateur, l'activité et les entités informatiques forment le contexte fondamental pour capturer les informations sur la situation d'exécution.

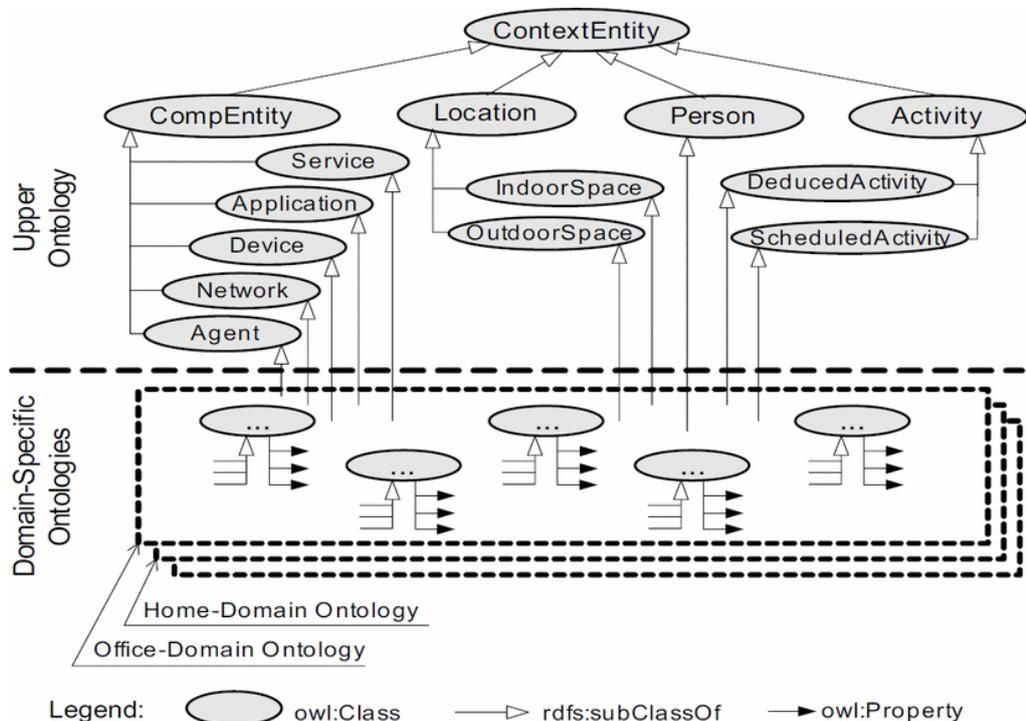


Figure 3.8 : Ontologie haut niveau du CONON [71].

La deuxième ontologie est une spécialisation de la première pour un domaine du contexte. Elle constitue un ensemble de sous-classes qui viennent se greffer à l'ontologie de base afin de détailler la modélisation suivant un environnement tel que la maison ou le lieu de travail. Alors que l'ontologie de premier niveau définit les termes qui caractérisent l'environnement d'une manière abstraite, la partie de l'ontologie spécifique au domaine est constituée d'un ensemble de termes concrets (GPS, mobile, maison, bureau, salleTD, etc.).

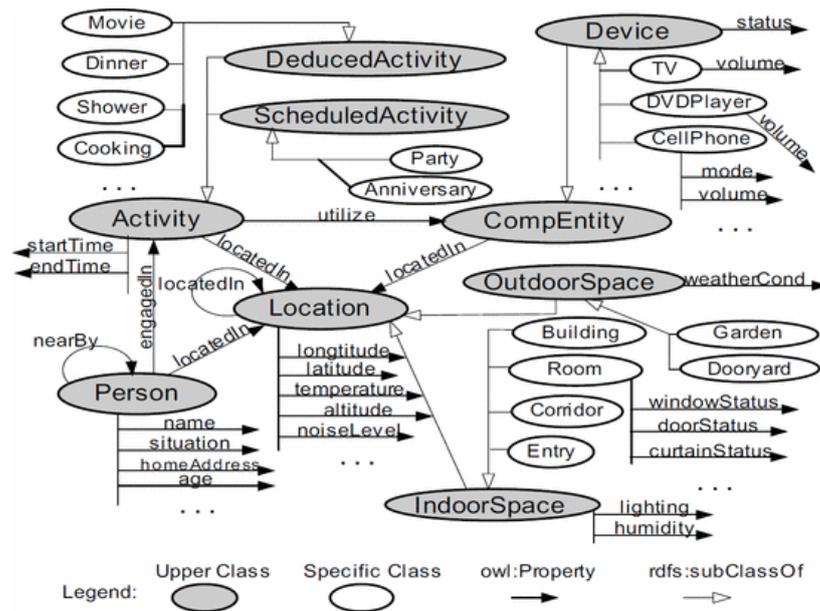


Figure 3.9 : exemple d'ontologie de CONON[71].

Dans l'approche CONON, le moteur de raisonnement est sollicité pour expliciter la description du contexte (par exemple déterminer de manière explicite qu'un utilisateur se trouve dans une salle, implique que cette même personne se situe également dans le bâtiment où se trouve cette salle !). En plus des règles de déductions de bases utilisées pour les ontologies, l'approche CONON permet d'ajouter des règles personnalisées. Ces règles sont écrites grâce à des prédicats associés à la logique de premier ordre. Elles sont introduites pour permettre de caractériser la situation de l'utilisateur.

3.3.3 Couplage des ontologies

Le couplage entre ces deux ontologies de façon à ce qu'elles représentent une ontologie unique et cohérente. Ce couplage pourrait être une fusion, un alignement ou une intégration [12]. Dans notre cas, nous avons fait cette intégration d'une façon manuelle. Les deux ontologies à unifier sont assez simples et ne nécessitent pas le recours à des méthodes d'intégration automatique ou semi-automatiques.

Nous avons décidé d'utiliser l'ontologie CONON comme ontologie de haut niveau

pour notre modèle. Nous étendrons cette ontologie avec les concepts relatifs au domaine de localisation et domaine médical. Nous intégrerons l'ontologie obtenue avec SOUPA afin d'avoir un modèle global capable de supporter les règles d'inférence.

Nous partons alors de l'ontologie de haut niveau CONON (*A*) comme point de départ. On y ajoute les concepts du domaine localisation et médical pour obtenir une ontologie (*B*). Cette ontologie est couplée avec l'ontologie SOUPA (*C*) afin d'obtenir notre modèle de contexte (*D*). Ce modèle permet la déduction du contexte au moyen des règles d'inférence.

Nous avons réalisé cette ontologie en plusieurs étapes.

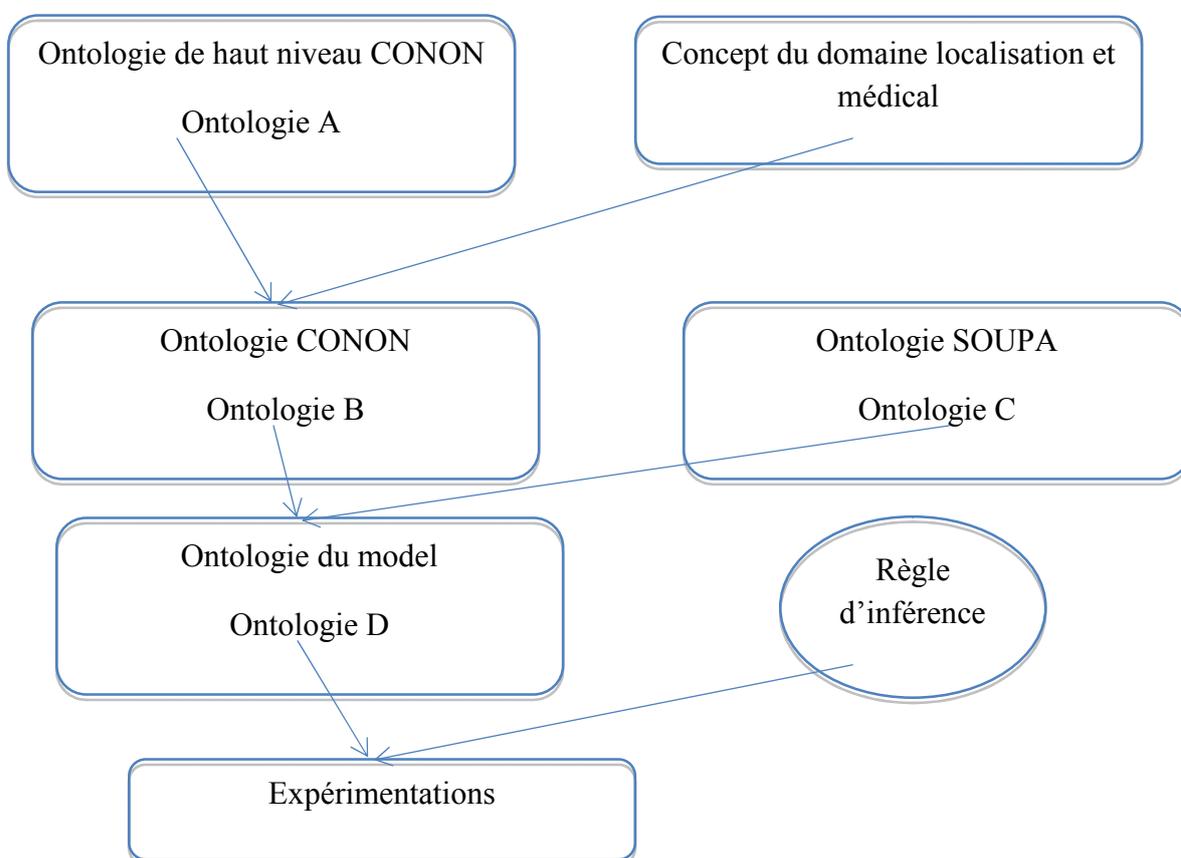


Figure 3.10 : diagramme du couplage.

L'ontologie représentée par la figure 3.9 est basée sur les concepts suivants : Personne, Activité, Entité Computationnelle et Localisation. Ces concepts sont reliés entre eux par des relations (Object Properties). Une activité se déroule (locatedIn) dans une

localisation et utilise une entité computationnelle. Une personne est propriétaire (ownsCompEnt) d'une entité computationnelle, est engagée (engagedIn) dans une activité, est située (situatedIn) dans une localisation, et possède (ownCompEnt) une entité computationnelle.

OWL permet la structuration des éléments de manière hiérarchique par la notion subClasseOfalors, la localisation est divisée en espace intérieur (Indoor) et extérieur (Outdoor). L'élément *Activité* (Activity) distingue entre les activités programmées (ScheduledActivity) et activités déduites (DeducedActivity). L'entité computationnelle (ComputationalEntity) peut être un périphérique (Device), un réseau (Network), un service (Service) ou une application (Applications).

Nous avons aussi organisé le concept Device en catégorisant les appareils par type : les appareils de communication pour l'exemple de localisation (téléphone mobile, GPS) et appareils médicaux pour l'exemple médical (tensiomètre, thermomètre, etc.) qui agissent comme capteur pour la mesure des données corporelles.

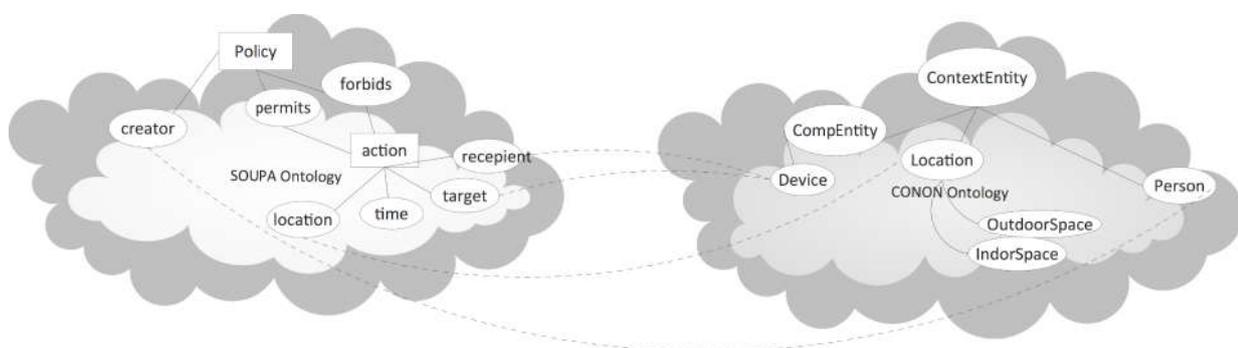


Figure 3.11 : couplage des ontologies.

3.4 Conclusion

On rappelle que l'approche qu'on a présentée à pour but de garder la sécurité des utilisateurs dans un système hybride, tout en utilisant les ontologies comme un moyen de représentations du contexte dans la réalisation de ce système.

Dans ce chapitre nous avons présenté l'architecture en trois couches pour l'internet des objets, par la suite nous avons présenté la description de l'architecture et son fonctionnement ainsi que le rôle de chacun de ces composants.

Le présent chapitre décrit, dans une dernière partie, les deux ontologies qui présentent

le contexte de notre architecture. Enfin, dans cette même partie, nous avons exposé le couplage entre ces deux ontologies.

Dans ce qui suit, nous présentons les tests et les expérimentations afin de valider notre approche. Nous montrons aussi les résultats des différents scénarios d'inférence.

CHAPITRE 4

EXPERIMENTATIONS

4.1 Introduction

Afin d'évaluer et valider la contribution présentée dans ce mémoire, une phase d'expérimentation s'avère indispensable. Cette phase a pour objectif d'étudier les performances de notre approche de sécurité de la vie privée dans l'internet des objets. En outre, ceci nous permettra aussi d'identifier les contraintes et les insuffisances de notre approche.

Ce chapitre est organisé comme suit ; nous commençons en premier lieu par l'environnement de développement qui va supporter l'application. Nous présentons les règles d'inférence et les résultats d'exécutions. L'étude de ces résultats nous permettra une évaluation plus précise de notre approche et l'examen de son intérêt du point de vue pratique.

4.2 Environnement de développement

Le travail est réalisé sur un mac possédant caractéristiques suivantes : un processeur Intel i7 Quad-core 2.8 GHz, et une mémoire de 16 GB. L'ensemble est piloté par Mac OS X 10.9.5 Mavericks.

4.2.1 Eclipse

Le prototype qui implémente notre approche a été développé entièrement en Java, sous l'environnement Eclipse [21]. Eclipse est un environnement de développement intégré (Integrated Development Environment), une plateforme de programmation développée par IBM, dont le but est de fournir un environnement robuste et convivial pour les développeurs de logiciels, outils, et systèmes informatiques. Eclipse utilise énormément le concept de modules nommés "plug-ins" dans son architecture. D'ailleurs, hormis le noyau de la plateforme, nommé "Runtime", tout le reste de la plateforme est développé sous la forme de plug-ins. Ce concept permet de fournir un mécanisme pour son extension, et ainsi, fournir des

opportunit  s a des tiers de d  velopper des fonctionnalit  s qui ne sont pas fournies en standard par Eclipse.

Les principaux modules de base d'Eclipse fournis concernent le langage de programmation Java. Les modules agissent sur des fichiers qui sont inclus dans le l'espace de travail (appel   workspace). L'espace de travail regroupe les projets qui contiennent une ou plusieurs arborescences de fichiers.

4.2.2 Prot  g   (  diteur d'ontologie)

Prot  g   [57] est un IDE d'ontologies distribu   en open source par l'universit   de Stanford (sp  cialis  e en informatique m  dicale). Prot  g  , c'est un   diteur "de base", modulaire et extensible via plugins. Gr  ce    ces plugins, il est capable de g  rer d'autres ontologies, comme OWL.

Connue de par ses multiples ressources tant logistiques que mat  riels, on a su l'adopter et l'utiliser au mieux, ce qui a r  pondu    nos attentes comme il le sera justifi   comme suit :

- Prot  g   permet un fonctionnement en mode client/serveur avec une gestion des droits ajustables en fonction des profils souhait  s.
- Prot  g   int  gre des outils de travail collaboratif comme le chat ou les annotations.
- Il int  gre une multitude de plugins permettant des manipulations sur les bases de connaissances cr  es.
- L'interop  rabilit   qui s'exprime sous de nombreux format d'exportations tel que : (HTML, OWL, RDF Sch  ma, XML, ACME).
- Supportant toute forme de plateforme confondu, et supportant JAVA tel (Linux (32 et 64 bits), Mac-OS X, Windows (32 et 64 bits)).

JESS et Prot  g  

La figure 4.1 montre comment utiliser Jess avec Prot  g   et OWL. Deux outils suppl  mentaires sont utilis  s pour que Jess puisse traiter des entr  es OWL : OWL2JESS et SWRL2JESS.

- OWL2JESS a le r  le de convertir l'ontologie OWL et des instances aux faits de JESS (en anglais FACTS). C'est la base de connaissance de moteur d'inf  rence.

- SWRL2JESS : Le but de cet outil est de convertir le langage SWRL en règle dans JESS. Avec SWRL, les utilisateurs peuvent créer un nouveau concept et l'ajouter à la base de connaissances.

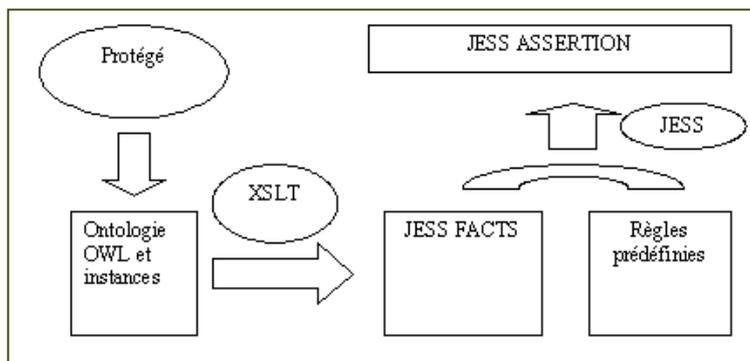


Figure 4.1 : Utiliser Jess avec OWL [57].

Edition des règles SWRL (pour protégé)

L'éditeur de règles SWRL opère dans Protégé OWL. Il fournit une interface très interactive pour l'édition des règles prenant en charge l'ensemble des fonctionnalités du langage SWRL. Les moteurs de règles tels que Jess peuvent être intégrés avec cet éditeur. Cette intégration permet de fournir un raisonnement à base de règles plus riche pour le Web sémantique.

4.2.3 SWRL

SWRL (Semantic Web Rule Language) [25] est un langage de règles pour le web sémantique combinant le langage OWL-DL et le langage RuleML (RuleMarkupLanguage). SWRL est un langage qui enrichit la sémantique d'une ontologie définie en OWL. SWRL permet contrairement à OWL de manipuler des instances par des variables ($?x$, $?y$, $?z$). SWRL ne permet pas de créer des concepts ni des relations. Il permet simplement d'ajouter des relations suivant les valeurs des variables et la satisfaction de la règle.

La syntaxe

La syntaxe SWRL définit une règle qui est une relation d'implication entre un antécédent (corps) et un conséquent (tête). Si les conditions spécifiées dans l'antécédent sont vérifiées c'est-à-dire si l'on a pu démontrer la partie corps, alors les conditions spécifiées dans le conséquent le sont aussi (antécédent \rightarrow conséquent).

L'antécédent et le conséquent d'une règle sont des conjonctions d'atomes. Les atomes peuvent avoir les formes:

- $C(x)$ où C est une description OWL, x est soit une variable, un individu OWL ou encore des valeurs de données (datavalues) de OWL.
- $P(x,y)$ où P est une propriété OWL (Object_property ou data_type_property), x est soit une variable ou un individu OWL et y est soit une variable, un individu OWL ou encore des valeurs de données de OWL.
- $\text{sameAs}(x,y)$, $\text{differentFrom}(x,y)$ où x et y sont des variables ou des individus OWL
- $\text{builtin}(r,x,\dots)$ où r est une relation built-in et x,\dots Sont des valeurs de données OWL

Le fonctionnement d'une règle est basé sur le principe de satisfiabilité de l'antécédent ou du conséquent. Pour une règle, il existe trois cas de figure :

- L'antécédent et le conséquent sont définis. Si l'antécédent est satisfait alors le conséquent doit l'être.
- L'antécédent est vide, cela équivaut à un antécédent satisfait ce qui permet de définir des faits.
- Le conséquent est vide, cela équivaut à un conséquent insatisfait, l'antécédent ne doit pas être satisfiable.

4.2.4 SQWRL

SQWRL [58] (SemanticQuery-enhanced Web RuleLanguage) est un langage d'interrogation qui peut être employé pour questionner des ontologies d'OWL, il fournit des opérations de SQL à la connaissance de format recherchée d'une ontologie d'OWL.

SQWRL ne change pas la sémantique de SWRL et n'emploie pas la syntaxe standard de présentation de SWRL, il est soutenu par le SWRLTab.

Aussi SQWRL est défini en utilisant une bibliothèque built-ins de SWRL qui transforme efficacement SWRL en langage d'interrogation.

4.3 Expérimentation

Nous avons effectué quelques simulations sur *Protégé* afin de s'assurer le fonctionnement correct des règles d'inférence. Ainsi, nous avons peuplé l'ontologie avec des individus (*Individuals*), nous avons défini les propriétés de ces individus et nous avons lancé le moteur d'inférence.

Nous avons défini le contexte représenté dans la figure 4.2, l'université de Blida qui possède ; Pavillon 1, Pavillon 2, Pavillon 3, Pavillon 4, Pavillon 5 et bibliothèque central. Nous avons aussi défini la politique PolicyLocation qui désactive la location pour notre

utilisateur dans L'université de Blida.

Nous avons aussi, à considérer un scénario médical dans le cas d'un patient équipé de dispositifs intelligents mobiles qui peuvent rapporter l'emplacement pour l'hôpital et désactiver la localisation dans les conditions normales. Mais quand elle/il est dans des circonstances urgentes, nous définissons la règle qui peut activer le service de localisation dans les cas critiques. La Figure 4.16 représente La fenêtre de notre application pour vérifie la localisation de l'utilisateur dans les cas critiques.



Figure 4.2 : Exemple de contexte (université de BLIDA)

Dans notre exemple de contexte comme indiqué dans Figure 4.2, nous avons défini la relation `haveUnderLocation` entre `(class :Location)` et `(class :Location)` comme `owl:inverseOf` relation, et la relation `haveUpperLocation` comme la propriété inverse de `haveUnderLocation`. Nous avons aussi défini la relation `haveUnderTime` entre `(class :Time)` et `(class :Time)` comme `owl:inverseOf` relation, et la relation `haveUpperTime` comme la propriété inverse de `haveUnderTime`. Figure 4.3 montre notre contexte où les lignes orange représentent la zone où l'utilisateur a désactivé le service de localisation avec la politique (`PolicyLocalisation`).



Figure 4.3 : Exemple de contexte avant l'inférence.

Après l'exécution des règles SWRL montrées dans le tableau 4.1. La figure 4.4 présente notre contexte où les lignes rouges représentent la zone où le service de localisation est désactivé suite à l'inférence.

La politique (PolicyLocalisation) a été enrichie sémantiquement, maintenant la (PolicyLocalisation) est désactivé la localisation de l'utilisateur dans l'université de Blida et aussi dans le Pavillon 1, Pavillon 2, Pavillon 3, Pavillon 4, Pavillon 5 et bibliothèque central.

La même pour le temps de la localisation, la politique (PolicyLocalisation) a été enrichie sémantiquement, maintenant la (PolicyLocalisation) est désactivé la localisation de l'utilisateur pendant la journée et aussi pendant le matin, le soir et la nuit.

La figure 4.14 et la figure 4.15 représentent l'interface de notre application qui vérifie la localisation de l'utilisateur dans chaque emplacement.

Dans cette étape nous avons montré que l'utilisation des règles d'inférence nous permettait d'enrichir le contexte et de définir de nouvelles politiques de sécurité.



Figure 4.4 : Exemple de contexte après l'inférence.

nom	SWRL	Explication
Rule1	policy:ForbiddenAction(?x) ^ action:time(?x,?y) ^ PrivacyOntology:haveUnderTime(?y, ?z) ->action:time(?x, ?z)	La règle pour interdire les actions qui ont le temps inclue dans d'autres actions interdites
Rule2	policy:PermittedAction(?x) ^ action:time(?x, ?y) ^ PrivacyOntology:haveUnderTime(?y, ?z) ->action:time(?x, ?z)	La règle pour permettre les actions qui ont le temps inclue dans d'autres actions permises
Rule3	policy:ForbiddenAction(?x) ^ context:locatedAt(?x, ?y) ^ PrivacyOntology:haveUnderLocation(?y, ?z) - >context:locatedAt(?x, ?z)	La règle pour interdire les actions qui ont l'emplacement inclus dans d'autres actions interdites
Rule4	policy:PermittedAction(?x) ^ context:locatedAt(?x, ?y) ^ PrivacyOntology:haveUnderLocation(?y, ?z) - >context:locatedAt(?x, ?z)	La règle pour permettre les actions qui ont l'emplacement inclus dans d'autres actions permises

Table 4.1 : règles SWRL pour l'exemple de localisation

nom	SWRL	Explication
Rule1	policy:policyOf(?x,? y) ^ person:Condition(?y, ?critical) ->policy:defaultPolicyMode(?x, ?Liberal)	La règle pour activer la localisation pour la personne (? Y) dans une condition critique

Table 4.2 : règle SWRL pour l'exemple médical.

Pour l'évaluation de la performance, les règles illustrées dans le tableau 4.1 et le tableau 4.2 ont été exécutées en utilisant le moteur d'inférence Jess avec une modification supplémentaire. L'ontologie est transformée pour l'ajouter dans la base de connaissances Jess.

4.3.1 Temps de Chargement

Nous avons mesuré le temps de chargement pour l'ontologie de haut niveau et l'ontologie de domaine spécifique dans les deux scénarios localisation et médical.

Figure 4.5 illustre le temps de chargement pour l'ontologie de haut niveau et l'ontologie de domaine spécifique. Un facteur important qui doit être noté est que le temps pour charger l'ontologie de domaine spécifique est environ quatre fois le temps pour charger l'ontologie de haut niveau.

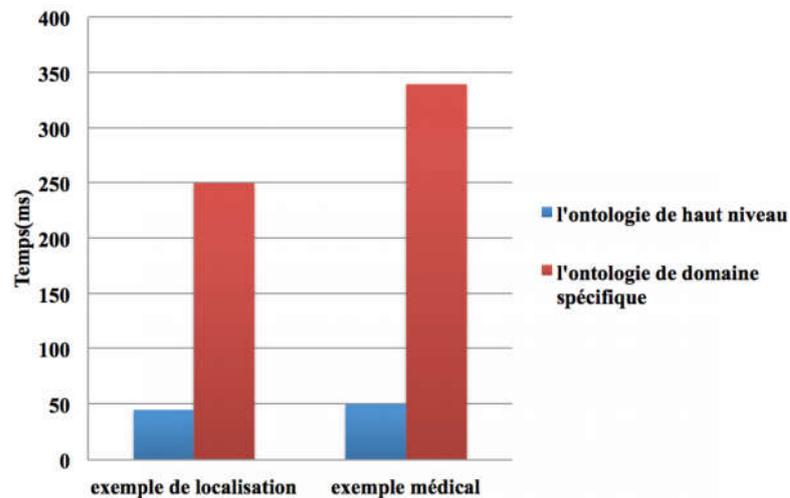


Figure 4.5 : Temps de chargement pour l'ontologie de haut niveau et l'ontologie de domaine spécifique.

4.3.2 Temps de transfert et d'exécution

D'après la figure 4.6 et la figure 4.7. Nous observons que le temps d'exécution a une relation linéaire avec le temps de transfert.

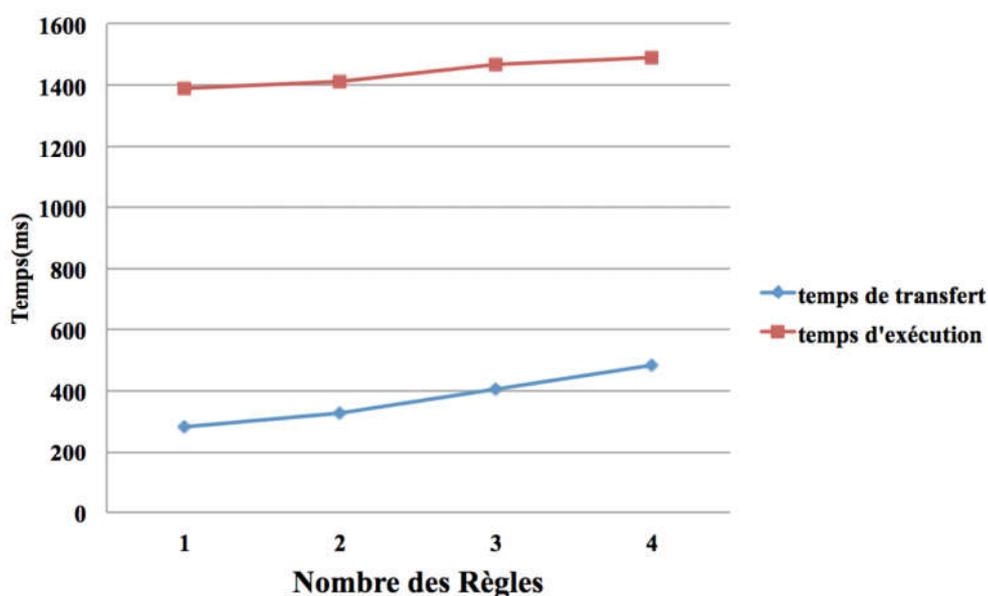


Figure 4.6 : Les résultats du Temps de transfert et d'exécution pour l'exemple de localisation

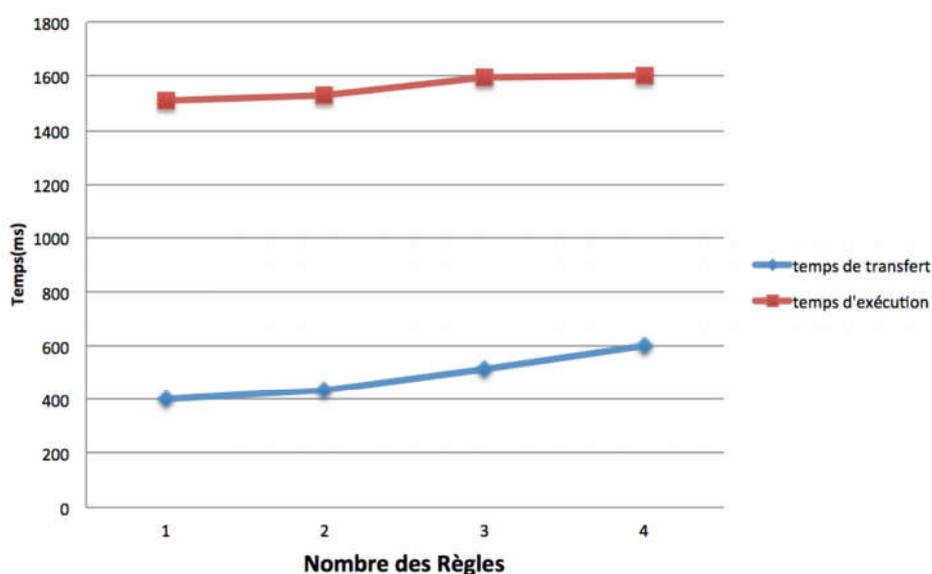


Figure 4.7 : Les résultats du Temps de transfert et d'exécution pour l'exemple médical.

4.3.3 Impact de la mise en cache du moteur d'Inférence

Un facteur important qui devrait être noté est que le temps de la deuxième exécution, comme on l'observe dans nos expériences, est proche de zéro, en raison de l'approche Jess où évolue la base de connaissances dans une première étape en vue de fournir l'exécution des règles au minimum temps sur la deuxième exécution.

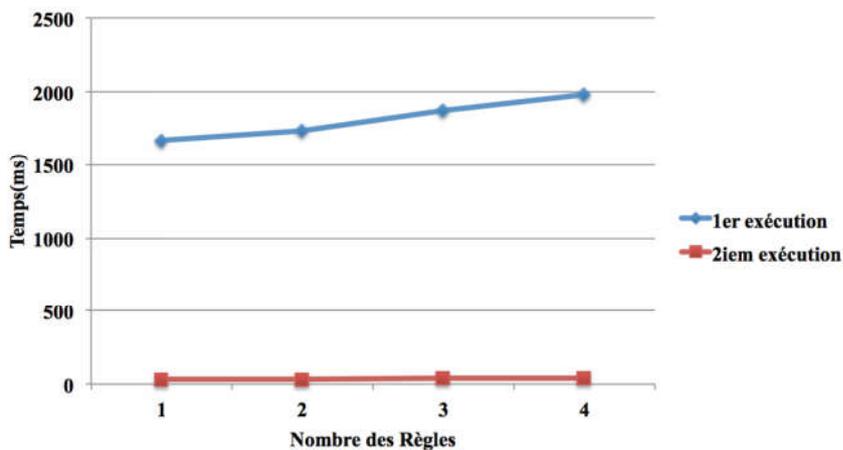


Figure 4.8 : Résultat montre la différence entre la première et la deuxième exécution pour l'exemple de localisation.

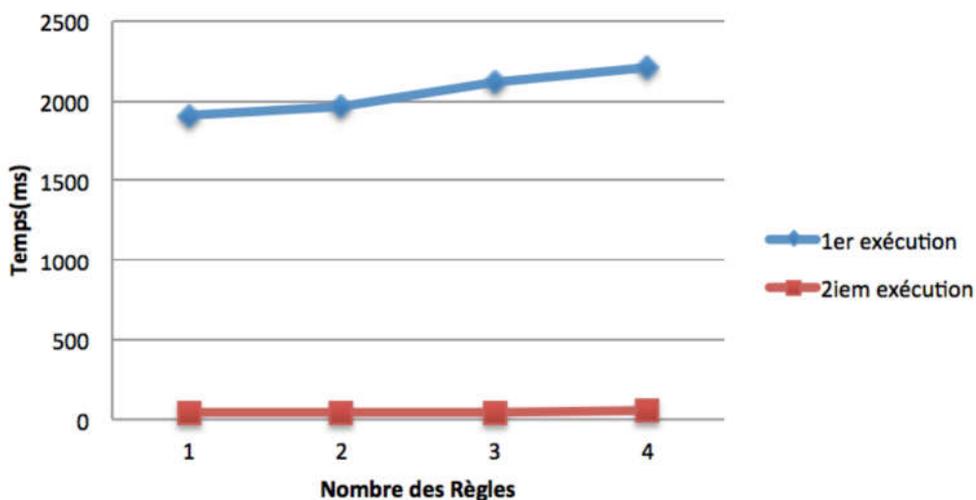


Figure 4.9 : Résultat montre la différence entre la première et la deuxième exécution pour l'exemple médical.

4.3.4 Temps de raisonnement

D'après la figure 4.10. Nous observons que le temps de raisonnement est presque la même pour les deux exemples.

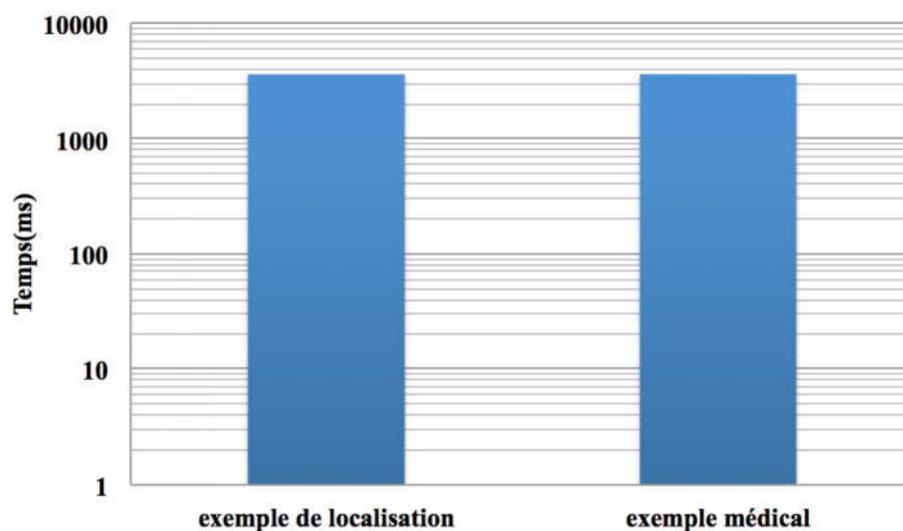


Figure 4.10 : Résultat du temps de raisonnement

4.3.4 Présentation de l'application

L'application que nous avons développée permet :

- Ouvrir notre ontologie et Afficher ces arbres : " OWLClasses ", " Properties " et " Individuals ".
- vérifier la localisation de l'utilisateur dans chaque emplacement de notre contexte.
- vérifier la localisation de l'utilisateur dans les cas critiques.

La figure suivante présente l'interface principale de notre application



Figure 4.11 : La fenêtre principale de notre application.

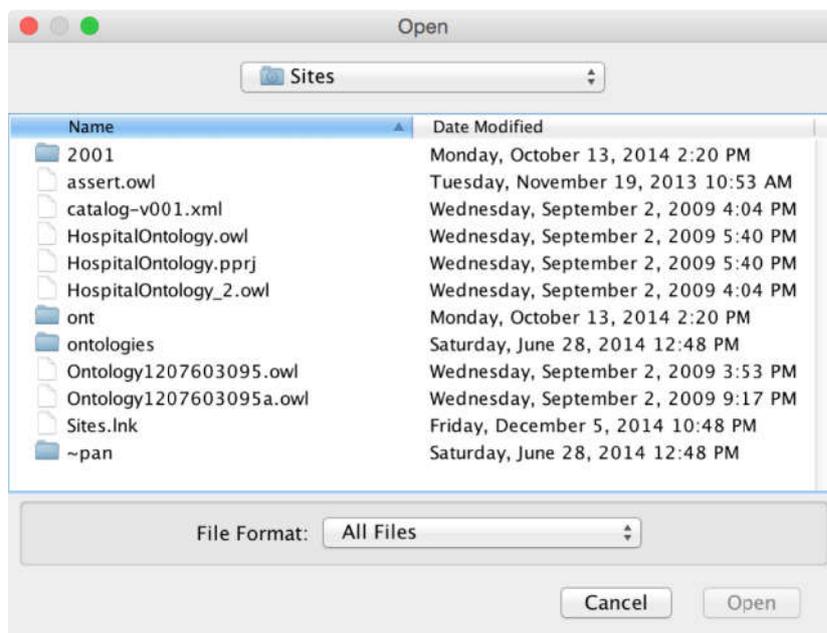


Figure 4.12 : La fenêtre pour sélectionner l'ontologie.

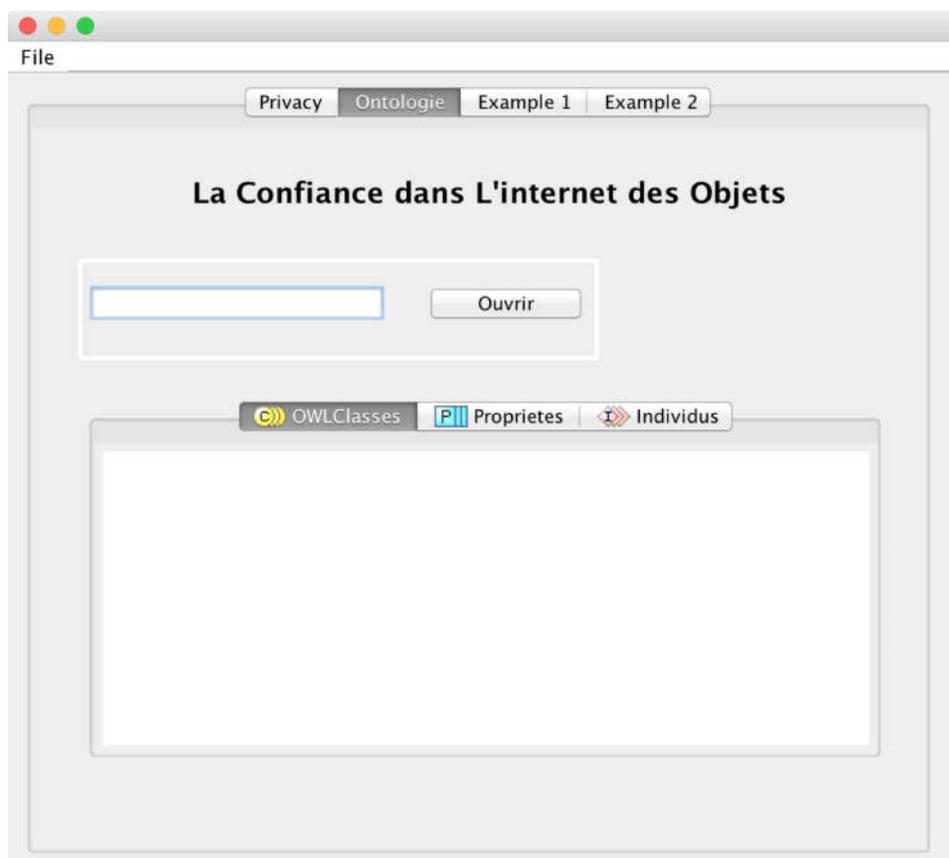


Figure 4.13 : La fenêtre d'ouverture de l'Ontologie.

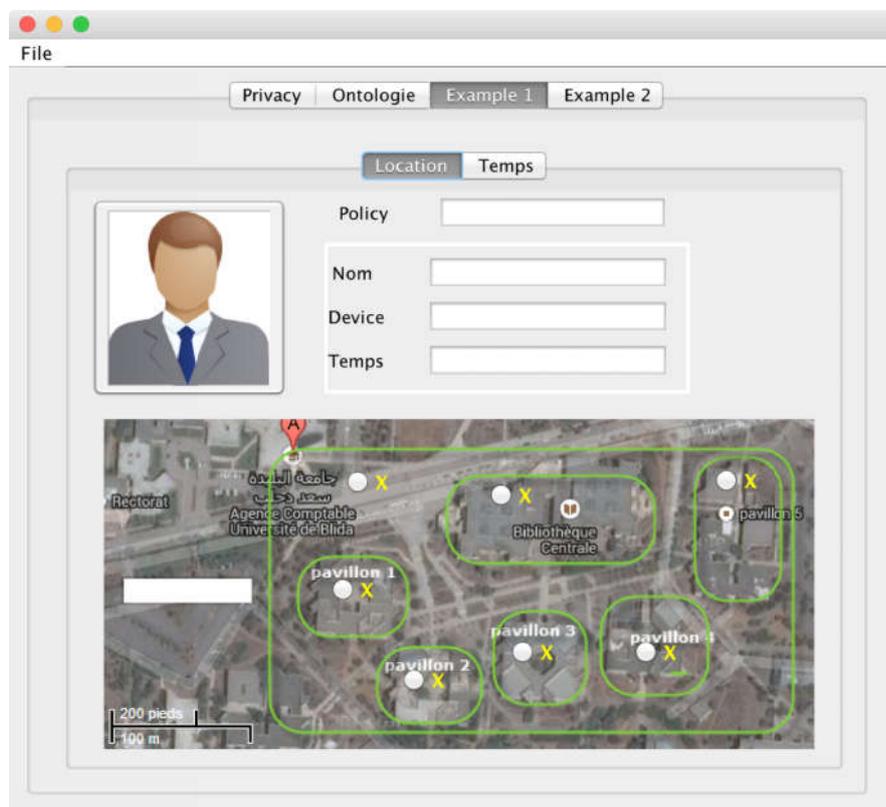


Figure 4.14 : La fenêtre de l'application pour l'exemple de localisation

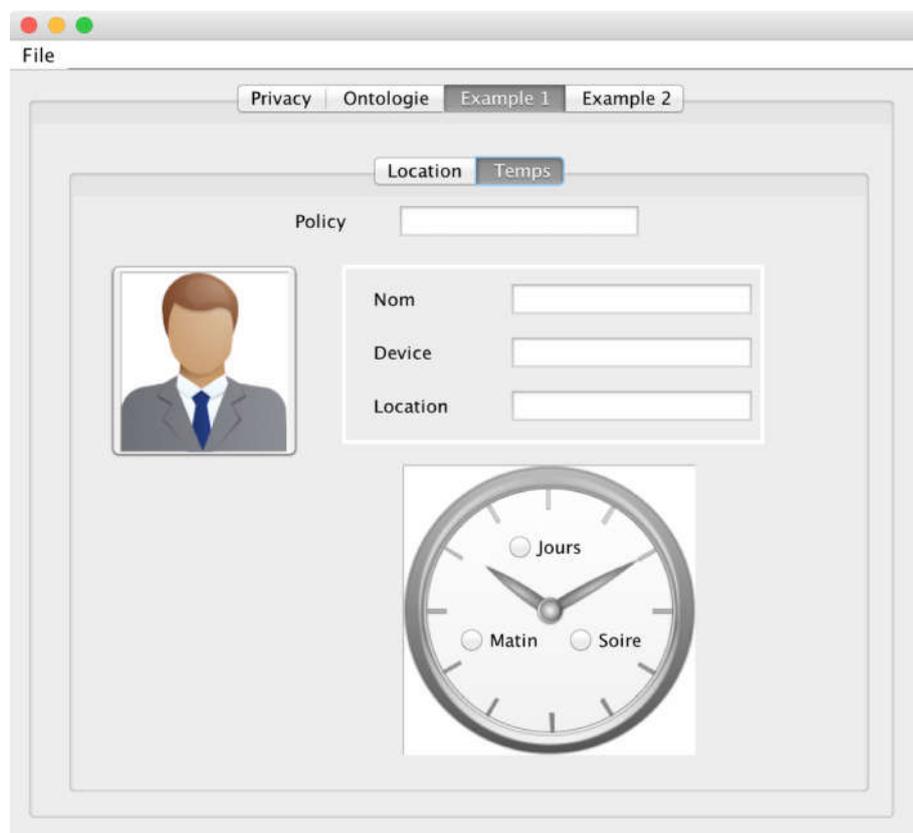


Figure 4.15 : La fenêtre de l'application pour l'exemple de localisation

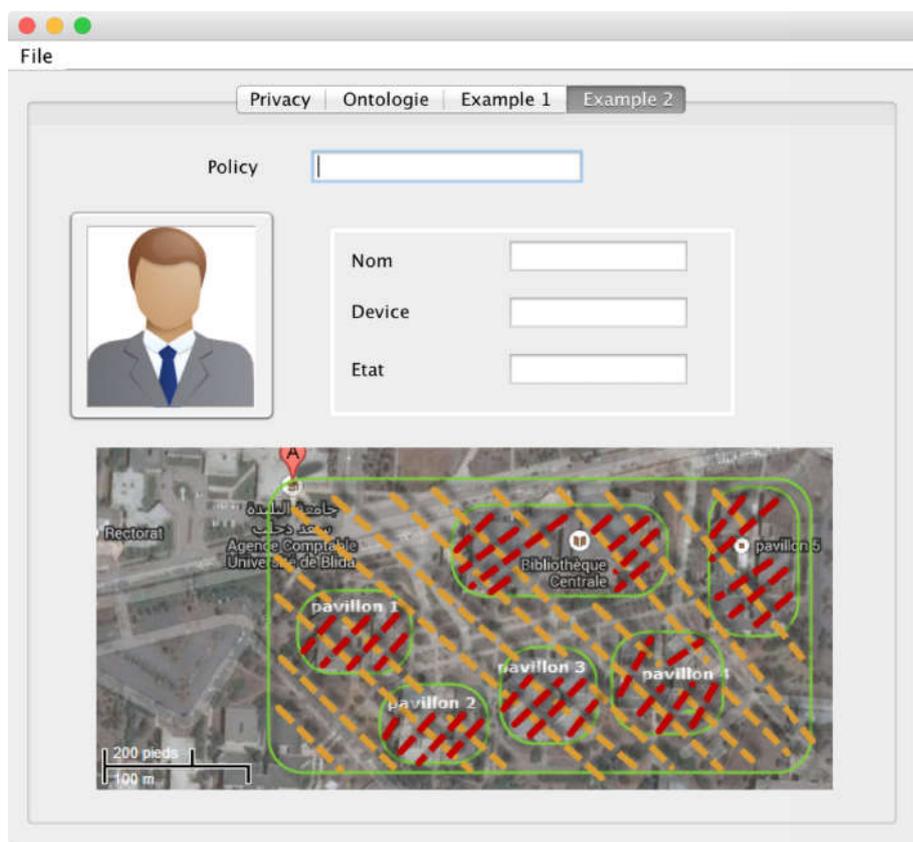


Figure 4.16 : La fenêtre de l'application pour l'exemple médical.

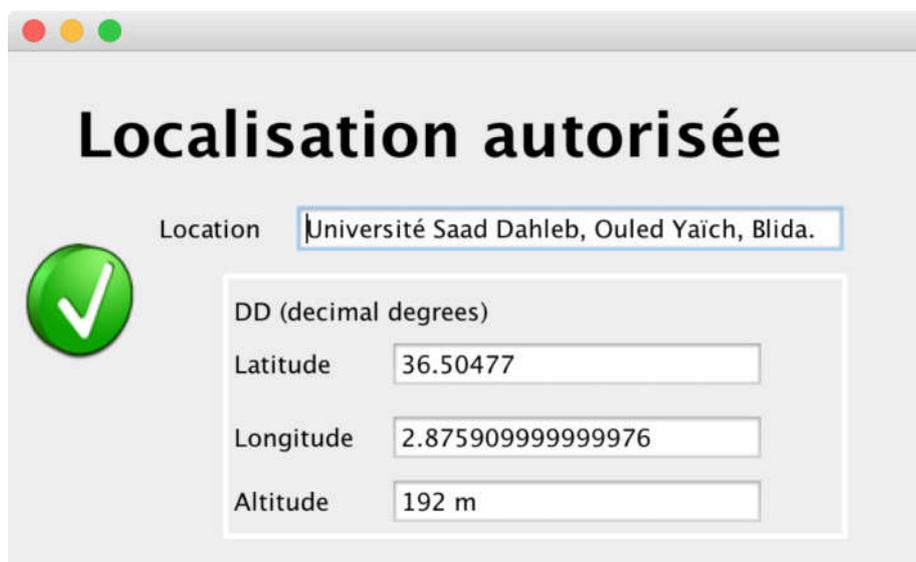


Figure 4.17 : La fenêtre de localisation autorisée



Figure 4.18 : La fenêtre de localisation non autorisée

4.4 Conclusion

Dans ce chapitre nous avons présenté le prototype que nous avons implémenté avec toutes les stratégies exposées dans les chapitres précédents. Nous avons commencé par la description de l'environnement de développement. Puis nous avons présenté le contexte de base de notre approche et les règles d'inférences. Nous avons démontré l'utilité et l'apport des ontologies dans la définition du contexte de sécurité de l'utilisateur dans le cadre d'une deux scenarios de localisation et médical. Nous avons montré et commenté les résultats d'expérimentations, obtenus par les différents scénarios. Nous avons constaté que ces résultats garantissent favorablement l'efficacité de notre approche. Pour terminer, nous avons présenté notre application.

CONCLUSION GENERALE

Dans ce mémoire, nous avons étudié le problème de sécurité et vie privée dans l'internet des objets. Nous avons commencé par la définition de concepts de base. Nous avons étudié les différentes définitions qui portent sur les notions d'ontologie et sensibilité au contexte et l'informatique en nuage et passé en revue les principales définitions concernent l'internet des objets. A la fin. Nous sommes ensuite focalisés sur la présentation de notre approche de sécurité et nous avons exposé les techniques et les méthodes utilisées pour combiner ces techniques ainsi que les trois principales couches existantes citées dans la littérature.

Le choix de l'approche ontologique dans la modélisation du contexte était motivé par la possibilité d'appliquer du raisonnement sur le modèle. En effet, l'ontologie fournit une forte expressivité qui permet de spécifier les concepts et leurs relations et qu'il fournit une formalisation des objets du monde réels dans un langage compréhensible par la machine. Les ontologies sont aussi extensibles et permettent de rajouter d'autres concepts ou d'être couplées avec d'autres ontologies.

La spécification et la mise en œuvre d'une architecture ouverte basée sur les ontologies pour les utilisateurs dans le contexte de l'internet des objets, même en se limitant aux propriétés de sécurité, est une tâche complexe. De ce fait, nous nous sommes limités, dans ce mémoire, au développement de l'ontologie qui s'inscrit dans le cadre de la sécurité liées aux vie privée des utilisateurs, avec la proposition, d'une manière générale, de quelques scénarios qui montrent comment l'utiliser.

Pour des besoins de validation de l'approche proposée, nous avons développé une interface graphique pour la simulation des scénarios. Cette interface permet de lancer la simulation d'un ou plusieurs scénarios à la fois.

Le travail que nous avons réalisé dans ce mémoire de magister représente un début de recherche dans un domaine intéressé qui peut être enrichi par plusieurs autres travaux futurs comme :

- Le renforcement et l'enrichissement de l'ontologie réalisée afin de capturer le maximum d'information de contexte.
- Définition des nouvelles règles d'inférence nous permettra d'enrichir le contexte et de définir de nouvelles politiques de sécurité.
- l'implémentation de ce système dans un environnement réel avec les capteurs et les diapositives nécessaires pour l'assistance des personnes malades.
- l'intégration complète de paradigme Cloud a notre architecture pour bénéficier de ces multiples avantages et sa puissance dans la mise en œuvre des systèmes complexes.

REFERENCES

1. Alain Michard, XML langage et applications, édition Eyrolles, 2001.
2. A. Harter, A. Hopper, P. Steggles, and A. Ward. The anatomy of a context-aware application. In *Mobile Computing and Networking*, pages 59–68, 1999.
3. A. Schmidt, K. A. Aidoo, A. Takaluoma, U. Tuomela, K. Van Laerhoven, and W. Van. Advanced interaction in context. In *Proceedings of the First International Symposium on Handheld and Ubiquitous Computing, HUC'99*, pages 89–101, Karlsruhe, Germany, September 1999.
4. Anind K. Dey, Gregory D. Abowd, and Daniel Salber. A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. *Human-Computer Interaction*, vol. 16, no 2, p. 97-166, 2001.
5. Apache Jena Java framework for building Semantic Web and Linked Data applications, <https://jena.apache.org/>.
6. Asuncion Gomez Perez, V. Richard Benjamins, *Overview of Knowledge Sharing and Reuse Components: Ontologies and Problem-Solving Methods*, 1999.
7. B. Schilit, M. Theimer, and B. Welch. Customising mobile applications. *USENIX Symposium on Mobile and Location-Independent Computing*, pages 129–138, 1993.
8. B.N. Schilit and M.M. Theimer. Disseminating active map information to mobile hosts. *Network, IEEE*, 8(5): 22-32, 1994.
9. Benoît Rivollet, Claude-Emmanuel Serre, *Les Réseaux De Capteurs*, Note de veille – CAPTIVEN, Juillet 2013.
10. Biegel, G., et V. Cahill. « A Framework for Developing Mobile, Context-aware Applications ». In the 2nd IEEE Conference on Pervasive Computing and Communication (March, 2004). p. 361-365, 2004.
11. Bonino, D.; Castellina, E.; Corno, F., "The DOG gateway: enabling ontology-based intelligent domotic environments," *Consumer Electronics, IEEE Transactions on*, vol.54, no.4, pp.1656,1664, November 2008.

12. Bruijn J., Ehrig M., Feier C., Recuerda F. M., Scharffe F. et Weiten M. Semantic Web Technologies, trends and research in ontology-based systems, chapter Ontology Mediation, Merging, and Aligning, WILEY, pp. 95–113. 2006.
13. Chalmers, « A historical view of context ». Computer supported cooperative work : CSCW, vol. 13, p. 223-247,2004.
14. Charline DIETRICH, les produits blancs connecté sont-ils un avenir auprès des particuliers, THESE Professionnelle, Pole universitaire Léonard de Vinci,2011.
15. Chen, Harry, Tim Finin, and Anupam Joshi. "A pervasive computing ontology for user privacy protection in the context broker architecture." University of Maryland, Baltimore County TR-CS-04-08 TechReport ,2004.
16. Coutaz J., Rey G., “Recovering foundations for a theory of contextors”, 4ICCADUI, Valenciennes, France, 2002.
17. D. Fensel, C. Bussler, A. Maedche: Semantic Web Enabled Web Services. International Semantic Web Conference, 2002.
18. Daniele Miorandi, Sabrina Sicari, Francesco De Pellegrini, ImrichChlamtac, Internet of things: Vision, applications and research challenges, Ad Hoc Networks, Volume 10, Issue 7, Pages 1497-1516, September 2012.
19. Dey, K. A., Abowd, D. G. Towards a better understanding of context and context-awareness. In Computer Human Interactions (CHI2000) Workshop on the What, Who, Where and How of Context-Awareness, 2000.
20. Djida Bahloul, une approche hybride de gestion des connaissances basée sur les ontologies: application aux incidents informatiques, Thèse de Doctorat, L’Institut National des Sciences Appliquées de Lyon,2006.
21. Eclipse IDE for Java Developers, <http://www.eclipse.org>.
22. Evren Sirin, Bijan Parsia, Bernardo Cuenca Grau, Aditya Kalyanpur, Yarden Katz, « Pellet: A practical OWL-DL reasoner » Web Semantics: Science, Services and Agents on the World Wide Web, Volume 5, Issue 2, Pages 51-53, June 2007.
23. Fi-ware internet of things (IoT) service enablement, <http://www.fiware.org> ,2011.
24. Florence AMARDEILH. Web Sémantique et Informatique Linguistique : propositions méthodologiques et réalisation d’une plateforme logicielle. Thèse de doctorat, Université Paris X - Nanterre, 2007.
25. Francis Lapique , Langages de requêtes pour base de connaissances,2006.
26. Frédéric Furst, Contribution à l’ingénierie des ontologies : une méthode et un outil d’opérationnalisation, Thèse de Doctorat, Université de Nantes, 2004.

27. G. Chen and D. Kotz. A survey of context-aware mobile computing research. Technical report, Technical Report TR2000-381, Dept. of Computer Science, Dartmouth College, 2000.
28. G. K. Mostéfaoui, G. Pasquier-Rocha, and P. Brézillon. Context-aware computing : A guide for the pervasive computing community. In ICPS, pages 39–48, 2004.
29. Guinard, Dominique, and VladTrifa.: "Towards the web of things: Web mashups for embedded devices." Workshop on Mashups, Enterprise Mashups and Lightweight Composition on the Web (MEM 2009), in proceedings of WWW (International World Wide Web Conferences), Madrid, Spain. 2009.
30. Harry Chen , Filip Perich , Tim Finin , Anupam Joshi. "Soupa: Standard ontology for ubiquitous and pervasive applications." Mobile and Ubiquitous Systems: Networking and Services, 2004. MOBIQUITOUS 2004. The First Annual International Conference on. IEEE, 2004.
31. Hofer, T., W. Schwinger, M. Pichler, G. Leonhartsberger, J. Altmann et W. Retschitzegger.«Context-awareness on mobile devices - the hydrogenapproach». In the 36thAnnual Hawaii International Conference onSystem Sciences (HICSS'03). p. 292-302,2003.
32. Internet of Things in 2020. Roadmap for the Future, 1.1 ed.: 27: Info D.4 Networked Enterprise & RFID; Info G.2 Micro & Nanosystems in co-operation with the working group RFID of the EPOSS, août 2008.
33. ITU Internet Reports, The Internet of Things, International Telecommunication Union, November 2005.
34. J. Cardoso, "Approaches to Developing Semantic Web Services", International Journal of Computer Science, 1 (1), 2006.
35. Jess the Rule Engine for the Java, <http://www.jessrules.com/jess/index.shtml> .
36. Jin-cui YANG, Bin-xing FANG, Security model and key technologies for the Internet of things, The Journal of China Universities of Posts and Telecommunications, Volume 18, Supplement 2, Pages 109-112,December 2011.
37. K. Henricksen, J. Indulska, and A. Rakotonirainy. Modeling context information in pervasive computing systems, pages 167–180, Zurich, Switzerland, 2002.
38. Kabbaj Abdeslam, Hadouch Mustapha.« Cloud Computing » IUT Nancy Charlemagne,2011.
39. Kovatsch, M. Lanter, M. Duquennoy, S. "Actinium: A RESTful runtime container for scriptable Internet of Things applications", 2012 3rd International Conference on the

- Internet of Things (IOT), pp.135, 142, 24-26, Oct. 2012.
40. L. Yu "Introduction to semantic web and semantic web services ", Chapman & Hall/CRC, 2007.
 41. L. Zheng, H. Zhang, W. Han et al., "Technologies, applications, and governance in the internet of things," in Internet of Things Global Technological and Societal Trends, O. Vermesan and P. Friess, Eds., pp. 141–176, River Publishers, 2011.
 42. Lortal Gaëlle, Etat de l'art Ontologies et Intégration-Fusion d'ontologies, 2002.
 43. Luis Miguel Vaquero Gonzalez, Luis Rodero Merino, Juan Caceres, and Maik Lindner. A break in the clouds: towards a cloud definition. Computer Communication Review, 50–55, 2009.
 44. Maeddine Touak, L'Internet des Objets : définitions, grands projets et politique japonaise, l'Institut National des Sciences Appliquées de Lyon, 2010.
 45. Mark Weiser, The computer for the 21st century, Scientific American Ubicomp, 1991.
 46. Matthias Baldauf, Schahram Dustdar, and Florian Rosenberg. "A survey on context aware systems". Int. J. Ad Hoc Ubiquitous Computer, 263- 277, June 2007.
 47. MELL, Peter et GRANCE, Tim. The NIST definition of cloud computing, 2011.
 48. Michel Riveill, « Ubiquitous computing: les challenges logiciels ». Université de Nice.
 49. Mike Dean, Guus Schreiber, Sean Bechhofer, Frank van Harmelen, Jim Hendler, Ian Horrocks Deborah. McGuinness Peter. Patel-Schneider et Lynn Andrea Stein, OWL Web Ontology Language - Reference, <http://www.w3.org/TR/2004/REC-owl-ref-20040210/>, 2005.
 50. Mohsin Saleemi, Natalia Diaz Rodriguez, Johan Lilius, and Ivan Parres. "A framework for context-aware applications for smart spaces". Filth international conference and 4th international conference on Smart Spaces and next generation wired/wireless networking. Berlin, Heidelberg: Springer, Lecture Notes in Computer Science, vol. 6869, p. 14-25, 2011.
 51. Mustapha Baziz, Application des Ontologies pour l'Expansion de Requêtes dans un Système de Recherche d'Informations, Université Paul Sabatier - Institut National Polytechnique de Toulouse, 2008.
 52. Nicola Bui, Internet of Things Architecture-Project, Deliverable D1.1 - SOTA report on existing integration frameworks/architectures for WSN, RFID and other emerging IoT related Technologies, European Commission, 2011.
 53. Objets communicants et internet des objets, livre blanc, association instituts carnot, 2011.

54. Oscar BOTERO. «Conception, analyse et évaluation d'un Réseau RFID Hétérogène. » Thèse de doctorat, Télécom & Management SudParis- Université Pierre et Marie Curie, Mai 2012.
55. Patrick Reignier, Intelligence Ambiante Pro-Active de la Spécification à l'Implémentation, Université Joseph-Fourier - Grenoble I, 2010.
56. Pierre-JeanBenghozi, Sylvain Bureau, Françoise Massit-Folea. L'Internet des objets. Quels enjeux pour les Européens ?. Rapport de la chaire Orange "innovation and regulation", Ecole polytechnique et TELECOM Paris Tech. 2008.
57. protégé-Ontology Editor and Knowledge Acquisition System, <http://protege.stanford.edu/>
58. protégéWiki:SQWRL , <http://protege.cim3.net/cgi-bin/wiki.pl?SQWRL>
59. R.K. Thomas and R. Sandhu. Models, protocols, and architectures for secure pervasive computing: Challenges and research directions. In Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second IEEE Annual Conference on, pages, 2004.
60. RDF. Resource Description Framework Primer. W3C Recommendation, <http://www.w3.org/TR/rdf-primer/>, 2004.
61. S. Hammoudi et D. Lopes, "Introduction aux Services Web", 2003.
62. S. Staab, A. Maedche, Axioms are objects too: Ontology engineering beyond the modeling of concepts and relations, Research report 399- Institute AIFB- Karlsruhe, 2000.
63. Sasa Radomirovic, Towards a Model for Security and Privacy in the Internet of Things,1st International Workshop on the Security of the Internet of Things (SecIoT'10),December2010.
64. Sylvain CHERRIER. « Web Sémantique et Informatique Linguistique : propositions méthodologiques et réalisation d'une plateforme logicielle. » Thèse de doctorat, Université Paris X - Nanterre, 2007.
65. T. Berners-Lee, J. Hendler et O. Lassila. The Semantic Web. Scientific American, 284(5):34-43, Mai 2001.
66. Tim Bray Jean Paoli C. M. Sperberg-McQueen Eve Maler, Extensible Markup Language (XML) 1.0, W3C Recommendation <http://www.w3.org/TR/REC-XML>,2000.
67. Vassilis Foteinos, Dimitris Kelaidonis, George Poullos, Vera Stavroulaki, Panagiotis Vlacheas, Panagiotis Demestichas, Raffaele Giaffreda, Abdur Rahim Biswas, Stephane Menoret, Gerard Nguengang, et al., A cognitive management framework for

- empowering the internet of things, *The Future Internet*, Springer, pp. 187-199, 2013.
68. Volker Haarslev , Kay Hidde , Ralf Möller , and Michael Wessel , *The RacerPro knowledge representation and reasoning system*, IOS Press,2012.
 69. Willy Malvault, *Vers une architecture pair-à-pair pour l'informatique dans le nuage*, LIG Laboratoire d'Informatique de Grenoble - SARDES, Université de Grenoble, 2011.
 70. Windson VIANA DE CARVALHO,« Mobilité et sensibilité au contexte pour la gestion de documents multimédias personnels : CoMMedia», Thèse de Doctorat, université de Grenoble, 2010.
 71. X. H. Wang, D. Q. Zhang, T. Gu, H. K. Pung: *Ontology Based Context Modeling and Reasoning using OWL*. Proc. of 2nd IEEE Annual Conf. on Pervasive Computing and Communications, Workshop Context Modeling and Reasoning, IEEE Computer Society, 2004.
 72. Xavier Lacot, *Introduction à OWL : un langage XML d'ontologies Web*,2005.
 73. Y. BELAID, "SERVICE WEB – SOAP", Centre d'enseignements de Grenoble, 2009.
 74. Zuraini Zainoland Keiichi Nakata. *Generic context ontology modelling: A review and framework*. In 2nd International Conference on Computer Technology and Development (! CCTD 2010), Piscataway, NJ: IEEE, p. 126-130, 2010.