



République Algérienne Démocratique et Populaire
Ministère de l'enseignement supérieur et de la
recherche scientifique

Université Larbi Tébessi - Tébessa



Faculté des Sciences Exactes et des Sciences de la Nature et de la Vie
Département : Mathématiques et Informatique

Mémoire de fin d'étude
Pour l'obtention du diplôme de **MASTER**
Domaine : Mathématiques et Informatique
Filière : Informatique
Option : Systèmes d'information

Thème

**LA gestion de conflit sémantique DANS une
MAISON intelligente : Enjeux & Défis**

Présenté Par :

Debaïlia Imen

Devant le jury :

| | | | |
|----------------------|-----|--------------------------|-----------|
| Mr. Hamidane Fathi | MAA | Université Larbi Tébessa | Président |
| Mr. Djeddai Ala | MCB | Université Larbi Tébessa | Examineur |
| Mr. Bourougaa Salima | MCB | Université Larbi Tébessa | Encadreur |

Date de soutenance : 14/09/2020

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

REMERCIEMENTS

Tout d'abord, longue vie à « Allah » qui nous a guidé sur le droit chemin tout au long du travail et nous a inspiré les bons pas et les justes réflexes. Sans sa miséricorde, ce travail n'aurait pas abouti.

*L'encadrement scientifique de ce travail a été assuré par **Dr. Bourougaa Salima**, maître de conférences classe B à la faculté des Sciences Exactes et des Sciences de la Nature et de la vie, Université Tébessa.*

Nous tenons vivement à lui exprimer nos profonde reconnaissances gratitude pour sa disponibilité, sa patience, sa compréhension, ses qualités humaines et ses intérêts portés pour notre sujet de travail. Nous le remercions de nous avoir fait confiance et d'avoir été présent aussi souvent que possible. Son soutien permanent et son dynamisme nous ont permis d'avancer plus loin dans notre travail.

Nos remerciements vont aussi aux membres de jury :

***Dr. Hamidane. F, & Dr. Djeddai. A**, d'avoir accepté d'examiner et critiquer ce mémoire et nous éclairer par ses précieux conseils.*

A tous les enseignants de département math & informatique

A mes parents & ma famille

A tous mes amis

DEDICACE

Je dédie ce travail

A mes parents

A ma famille

A mes sœurs

A mes frères

A toutes mes amies

A tous ceux qui m'ont aidé dans mes études.

A tous mes proches, et tous ceux qui m'aiment.

Jmen

| | |
|--|-----------|
| Introduction Générale..... | 1 |
| 1. Introduction | 2 |
| 2. Problématique | 2 |
| 3. Objectives de travaille | 3 |
| 4. Structure du mémoire | 3 |
| | |
| Partie 1 : | |
| Chapitre 1 : Internet des objets & Smart Cities..... | 4 |
| 1. Introduction | 5 |
| 2. Notion de l'IOT | 5 |
| 2.1. Définition de IOT | 6 |
| 2.2. Les composants de l'Internet des Objets..... | 8 |
| 2.3. Evolution de l'internet des objets | 8 |
| 2.4. Les applications actuelles de l'Internet des Objets..... | 10 |
| 2.5. Les domaines de l'Internet des Objets | 10 |
| 1. Les Villes Intelligentes | 10 |
| 2. Le Smart Grid..... | 11 |
| 3. Les Appareils Intelligents | 12 |
| 4. Les Maisons et Les Bâtiments Intelligents | 13 |
| 5. Le Système De Santé Electronique | 14 |
| 6. Le Transport et La Mobilité Intelligente | 15 |
| 7. Les Usines et La Fabrication Intelligente | 16 |
| 8. La Surveillance à Distance Des Patients | 16 |
| 3. CONCLUSION | 18 |
| | |
| Chapitre 2 : Les Ontologies & Le Smart Home | 20 |
| 2.1. Introduction | 21 |
| 2.2. Notion d'ontologie | 21 |
| 2.2.1. Définitions..... | 21 |
| 2.2.2. Que représente-t-on dans une ontologie ? | 23 |
| 2.2.3. Différentes sortes d'ontologies..... | 24 |
| 2.2.4. Les ontologies et les systèmes à bases de connaissances | 26 |

| | | |
|--------|---|----|
| 2.2.5. | Les ontologies et le web sémantique | 27 |
| 2.3. | Les ontologies : Différents besoins | 27 |
| 2.4. | Construire des ontologies | 28 |
| 2.5. | Méthodologies de construction d'ontologies..... | 29 |
| 2.6. | Formalismes de représentation | 31 |
| 2.7. | Les niveaux de description..... | 32 |
| 2.8. | Outils de développement d'ontologies | 33 |
| 2.9. | Langages d'interrogation d'ontologies..... | 37 |
| 2.10. | Editeur d'ontologies | 39 |
| 2.11. | La maison intelligente | 40 |
| 2.12. | CONCLUSION | 44 |

| | | |
|-------------------|---|----|
| Chapitre 3 | Les travaux connexes sur La gestion des..... | 45 |
| | Conflits sémantiques dans une maison intelligente..... | 45 |
| | Synthèse & Discussion | 45 |
| 3.1. | Introduction | 46 |
| 3.2. | Gestion de conflits..... | 46 |
| 3.2.1. | Définitions..... | 47 |
| 3.3. | Les classes des approches existantes | 50 |
| 3.4. | Les solutions existantes | 52 |
| 3.4.1. | Les solutions classiques | 52 |
| 3.4.2. | Les solution basée ontologie | 54 |
| 3.5. | Critiques sur les méthodes de de gestion de conflits..... | 57 |
| 3.6. | Synthèse et discussion..... | 57 |
| 3.7. | Conclusion..... | 60 |

Partie2 :

| | | |
|---------------------|--|----|
| Chapitre 4 : | Conception d'une ontologie | 62 |
| | Pour la résolution des conflits sémantique | 62 |
| | Dans une maison intelligente..... | 62 |
| 4.1. | Introduction | 63 |
| 4.2. | Définition de conflit..... | 63 |
| 4.3. | Agrégation des conflits | 64 |
| 4.3.1. | Le conflit direct..... | 64 |
| 4.3.2. | Le conflit indirect..... | 66 |

| | | |
|---|--|-----------|
| 4.4. | Motivation des ontologies | 71 |
| 4.5. | Processus de construction de l'ontologie | 71 |
| 4.5.1. | Spécification | 72 |
| 4.5.2. | Conceptualisation..... | 72 |
| 4.5.3. | Formalisation | 73 |
| 4.5.4. | Implémentation | 73 |
| 4.5.5. | Tests et évolution | 74 |
| 4.6. | Construction de l'ontologie | 74 |
| 4.6.1. | Spécification..... | 74 |
| 4.6.2. | Conceptualisation..... | 76 |
| 4.7. | CONCLUSION | 86 |
| | | |
| Chapitre 5 : Implémentation d'une ontologie..... | | 87 |
| pour la résolution des conflits sémantique | | 87 |
| dans une maison intelligente | | 87 |
| 5.1. | Introduction | 88 |
| 5.2. | Implémentation | 88 |
| 5.2.1. | Présentation de l'éditeur PROTEGE OWL..... | 88 |
| 5.2.2. | Définition de la hiérarchie des classes | 89 |
| 5.2.3. | Définition des propriétés | 90 |
| 5.2.4. | Définitions des restrictions | 92 |
| 5.2.5. | Définition des instances | 93 |
| 5.2.6. | Définition des règles SWRL | 94 |
| 5.2.7. | Test de l'ontologie..... | 95 |
| 5.3. | CONCLUSION | 98 |
| | | |
| Conclusion Générale | | 99 |
| 1. | Conclusion..... | 100 |
| 2. | Les perspectives | 100 |

Liste de figure

| | |
|--|----|
| Figure 1 – l’historique de l’Internet des objets. [Sagar15] | 5 |
| Figure 2 – Définition d’Internet des objets [Friess11] | 7 |
| Figure 3 – les constituants de la ville intelligent [Friess11] | 11 |
| Figure 4 – les constituants de la smart grid [Thomas 19] | 12 |
| Figure 5 – les constituants des appareils intelligents [Friess11] | 12 |
| Figure 6 – les constituants des maison intelligents [Friess11] | 14 |
| Figure 7 – un système de santé électronique [Friess11] | 15 |
| Figure 8 – les aspects de transport intelligent [Friess11] | 15 |
| Figure 9 – un schéma de l’industrie 4.0 | 16 |
| Figure 10 – le mécanisme de la surveillance distante des patients [Alok14] | 17 |
| Figure 2.1 : Les différents sortes d’ontologie [Bourougaa19] | 21 |
| Figure 2.2 : Les couches du Web sémantique[BS19] | 23 |
| Figure 2.3 : Domaines d’utilisation des Ontologies. | 24 |
| Figure2.4 : Processus de construction d’ontologie [Bourougaa19] | 25 |
| Figure 2.5 : La pyramide des langages du Web sémantique | 31 |
| Figure 2.6 : Exemple : Imen a 24 ans et habite Tébéssa | 31 |
| Figure 2.7 : Taxinomie de concept personne | 32 |
| Figure 4.1 : Exemple de différents conflits dans la maison intelligente | 64 |
| Figure 4.2 : Exemple de conflit direct | 67 |
| Figure 4.3 : Exemple de conflit indirect | 68 |
| Figure 4.4 : Un document RDF de spécification de l’ontologie. | 76 |
| Figure 4.5 : Diagramme de classification de concepts | 81 |
| Figure 4.6 : Diagramme de relations binaires | 82 |
| Figure 5.1 : Interface de Protégé OWL | 89 |
| Figure 5.2 : Création des classes | 90 |
| Figure 5.3: Création des propriétés pour une classe | 91 |
| Figure 5.4 : Création d’un attribut | 91 |
| Figure 5.5 : Création d’une relation | 92 |
| Figure 5.6 : Création d’une restriction sur une classe | 93 |
| Figure 5.7 : Création des instances | 94 |
| Figure 5.8 : Création des règles SWRL | 94 |
| Figure 5.9 : Vérification de l’URL | 96 |
| Figure 5.10 : Test de consistance | 97 |
| Figure 5.11: Test de classification | 97 |

Liste de tableau

| | |
|---|----|
| Tableau 1.1 – Définition d’Internet des objets | 7 |
| Tableau 1.2 – Les dimensions conceptuel et technique [futura20] | 7 |
| Tableau 1.3 – L’évolution technologique applicable dans l’iot [Alin10] | 9 |
| Tableau 2.1 : Une base de connaissances composée d’une TBox et d’une ABox | 29 |
| Tableau 3.1 : Synthèse des méthodes de gestion de conflits | 70 |
| Tableau 4.1 : Conflits & Causes | 70 |
| Tableau 4.2 : Conflits & Solutions | 71 |
| Tableau 4.3 : Glossaire de termes | 79 |
| Tableau 4.5 : Dictionnaire de concepts | 83 |
| Tableau 4.6 : Tableau des relations binaires | 84 |
| Tableau 4.7 : Tableau des attributs | 84 |
| Tableau 4.8 : Tableau des axiomes logiques | 95 |
| Tableau 4.9 : Tableau des instances | 95 |

إنترنت الأشياء هو الترابط بين الإنترنت والأشياء. يمهد الطريق للأنظمة لمساعدة الناس على العيش في بيئات مثل المنزل الذكي. إنه يوفر الوجود في كل مكان للأشياء القادرة على التفاعل مع بعضها البعض والتعاون مع جيرانها لتحقيق أهداف مشتركة. تشمل أنظمة إنترنت الأشياء على جهات فاعلة تتمتع بخبرات متنوعة للغاية. وبالتالي، يصبح من الضروري وجود أوصاف معيارية ودلالية لحل النزاعات المتعلقة بقابلية التشغيل البيئي وعدم التجانس الدلالي بين الموارد المتاحة المختلفة من ناحية، وبين مختلف المستخدمين من ناحية أخرى.

تعتبر الأنطولوجيا من أهم الحلول لمشكلة عدم التجانس، لأنها تقدم دلالات مشتركة قادرة على منع فشل الاتصال والتفاعل بين التطبيقات الذكية والمستخدمين بسبب عدم تجانسهم وتضارب مصالحهم.

لحل التعارضات في التطبيقات السياقية في بيئات المنزل الذكي "H-Ontology" لذلك ، اقترحنا نهجًا جديدًا الكلمات الرئيسية: إنترنت الأشياء، المنزل الذكي، علم الوجود، الصراع، عدم التجانس، قابلية التشغيل البيئي

L'Internet des objets est l'interconnexion entre l'Internet et des objets. Elle ouvre la voie pour des systèmes ayant pour but d'aider les gens à vivre dans un environnement ubiquitaire tel que la maison intelligente. Elle offre l'omniprésence d'objets capables d'interagir les uns avec les autres et de coopérer avec leurs voisins pour atteindre des objectifs communs. Les systèmes d'internet des objets fait intervenir des acteurs d'expertises très diverses. Ainsi, il devient indispensable d'avoir des descriptions standardisées et sémantiques pour résoudre les conflits liés à l'interopérabilité et l'hétérogénéité sémantique entre les différentes ressources disponibles d'une part, et entre les différents utilisateurs d'autre part.

L'ontologie est considérée parmi les solutions les plus importantes de problème d'hétérogénéité, car elle offre une sémantique partagée capable d'empêcher l'échec de communication et d'interaction entre les applications intelligentes et les utilisateurs dû à l'hétérogénéité de leurs propriétés conflictuelle.

De ce fait, nous avons proposé une nouvelle approche « H-Ontologie » pour résoudre les conflits pour les applications contextuelles dans les environnements domestiques intelligents.

Mots clés : Internet des objets, maison intelligente, Ontologie, Conflit, Hétérogénéité, interopérabilité

ABSTRACT

The Internet of Things is the interconnection between the Internet and things. It paves the way for systems to help people live in ubiquitous environments such as the smart home. It offers the ubiquitous presence of objects capable of interacting with one another and cooperating with their neighbors to achieve common goals. Internet of Things systems involve actors with very diverse expertise. Thus, it becomes essential to have standardized and semantic descriptions to resolve conflicts related to interoperability and semantic heterogeneity between the different available resources on the one hand, and between the different users on the other hand.

Ontology is considered among the most important solutions to heterogeneity problem, because it offers shared semantics capable of preventing communication and interaction failure between intelligent applications and users due to heterogeneity of their conflicting properties.

Therefore, we have proposed a new “H-Ontology” approach to resolve conflicts for contextual applications in smart home environments.

Keywords: Internet of things, smart home, Ontology, Conflict, Heterogeneity, interoperability



Introduction Générale

1. Introduction

La maison intelligente est un terme qui fait référence aux maisons modernes dotées d'appareils, d'éclairage et / ou d'appareils électroniques pouvant être contrôlés à distance par le propriétaire, souvent via une application mobile.

Les appareils intelligents pour la maison peuvent également fonctionner en conjonction avec d'autres appareils de la maison et communiquer des informations à d'autres appareils intelligents. Les appareils intelligents pour la maison peuvent inclure des appareils tels que des réfrigérateurs, des machines à laver, des sècheuses et des fours grille-pain, ainsi que des unités de chauffage et de climatisation et des appareils d'éclairage.

2. Problématique

La maison intelligente est composée d'objets plutôt hétérogènes. Quelques exemples d'appareils électroniques intelligents pour la maison sont les systèmes de divertissement audio et vidéo, les systèmes de caméra et de sécurité, et les ordinateurs, ordinateurs portables et autres appareils mobiles électroniques.

Dans le contexte d'un environnement domestique intelligent, l'interopérabilité signifie la capacité des systèmes, des applications et des services à fonctionner ensemble de manière fiable et prévisible. L'interopérabilité est définie comme la capacité de deux systèmes ou plus à échanger des informations et à utiliser les informations échangées. L'interopérabilité permet l'échange d'informations et de ressources entre des sous-systèmes définis. La chose la plus importante est de s'assurer que ces sous-systèmes interagissent de manière transparente et fournissent aux résidents des applications intégrées sans modifier la plate-forme ou les protocoles des systèmes sous-jacents.

Pour cela, dans une maison intelligente différents conflits sémantiques peuvent survenir entre les souhaits des différents utilisateurs. Ce là est dû à l'hétérogénéité, interopérabilité et la dynamique des contextes des différents objets composants de cette maison.

Alors, Comment peut on résoudre les conflits sémantiques dans un environnement intelligents ?

3. Objectives de travaille

Notre but général est de proposer une nouvelle approche pour résoudre les conflits pour les applications contextuelles dans les environnements domestiques intelligents et l'implémenté en utilisant les outils de l'ontologie. Ainsi que de faire un état de l'art : travaux connexes, et Détails des défis de recherche pour faire une discussion et étude comparative.

4. Structure du mémoire

Notre mémoire est composée de deux parties, chaque partie contient 2 comme suit :

Partie1 : état de l'art

Chap1 : Internet des Objets et Smart Cities

Chap2 : Les ontologies et le Smart Home.

Chap3 : Les travaux connexes sur la gestion des conflits sémantiques dans une maison intelligente : synthèse et discussion.

Partie 2 : contribution

Chap4 : Conception d'une ontologie pour la résolution des conflits sémantique dans une maison intelligente.

Chap5 : Implémentation d'une ontologie pour la résolution des conflits sémantique dans une maison intelligente.



Partie 1 :

Chap1 : Internet des Objets et Smart Cities

Chap2 : Les ontologies et le Smart Home.

Chap3 : Les travaux connexes sur la gestion des conflits sémantiques dans une maison intelligente : synthèse et discussion.

Internet des objets & Smart Cities

1. Introduction

L'Internet des objets (IoT) est un système de dispositifs informatiques interdépendants, de machines mécaniques et numériques dotés d'identifiants uniques et de la possibilité de transférer des données sur un réseau sans nécessiter d'interaction entre humains ou entre humains.

Dans ce chapitre, nous présentons les différents concepts de (IoT), les environnements intelligents en particulier les maisons intelligentes.

2. Notion de l'IOT

Le terme Internet des objets (IoT), inventé par Ashton en 1999 [Ashton 09], a été une tendance technologique croissante ces dernières années. Au cours de la dernière décennie, l'IoT a gagné une attention considérable dans le monde universitaire et l'industrie informatique. L'IoT offre beaucoup plus de fonctionnalités par rapport à d'autres approches de mise en réseau comme le filaire / sans fil, le LAN, l'Ethernet, etc.

L'IoT promet un monde où tous les objets intelligents qui nous entourent sont connectés les uns aux autres et communiquent spontanément les uns avec les autres avec le minimum d'intervention humaine. En fin de compte, l'objectif de l'IoT est de créer un monde meilleur pour les êtres humains.

La figure 1 La figure suivante montre l'historique de l'Internet des objets

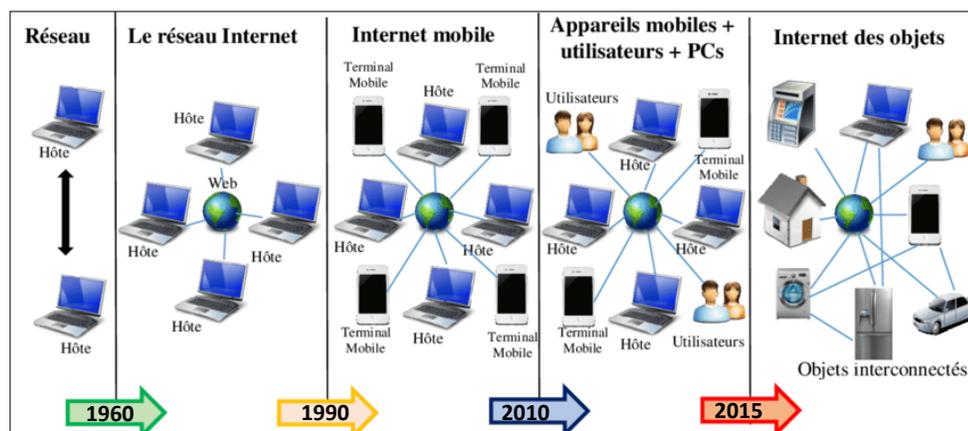


Figure 1 – l'historique de l'Internet des objets. [Sagar15]

Dans les années 1960, la communication entre deux ordinateurs a été rendue possible grâce au réseau informatique.

Au début des années 80, la pile TCP / IP a été introduite, puis l'utilisation commerciale d'Internet a commencé à la fin des années 80.

Plus tard, en 1991, le World Wide Web (WWW) a été introduit, ce qui a rendu Internet populaire et a stimulé la croissance rapide. WWW comprend également le Web of Things (WoT), qui fait partie de l'IoT.

Plus tard, divers appareils mobiles se sont connectés à Internet et ont formé l'Internet mobile. Ensuite, par les réseaux sociaux, les utilisateurs se connectent les uns aux autres via Internet. La prochaine grande chose est l'IoT, où les objets autour de nous peuvent se connecter les uns aux autres sur le réseau et communiquer avec l'aide d'Internet. [Sagar15]

2.1. Définition de IOT

Dans cette partie on va présenter les définitions de l'IOT. Il existe des dizaines de définitions de l'internet des Objets. Certaines claires et succinctes, mais trop spécifiques, elles sont échouées de prendre en compte la diversité des concepts de la terminologie (informatique diffuse, systèmes cyber-physique, intelligence ambiante, réseaux de capteurs, numérique connecté, ubiquité, etc..).

Quand les objets peuvent sentir et communiquer dans l'environnement, ils deviennent des outils pour comprendre la complexité et répondre à elle vite. Ce qui est révolutionnaire en tout c'est que ces systèmes d'information physiques commencent maintenant à être déployés, et certains d'entre eux même travaillent en grande partie sans intervention humaine [McKinsey10]. La figure 2 et le tableau 1 montrent respectivement la définition de River [Friess11] et les définitions de Atzori [Atzori08] et [CITC13]

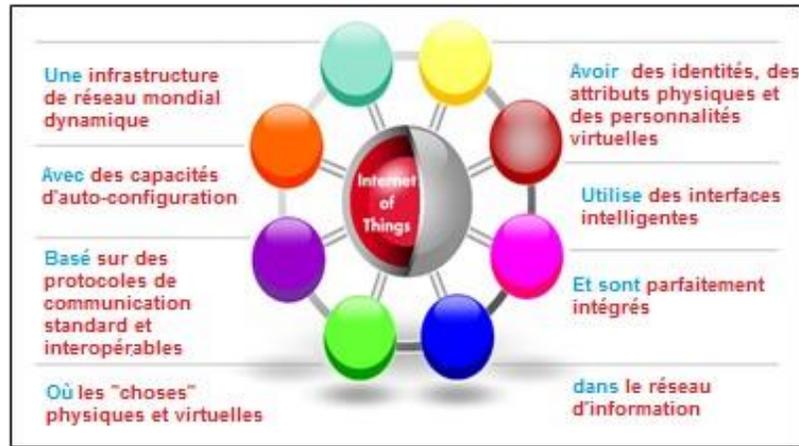


Figure 2 – Définition d’Internet des objets [Friess11]

| | |
|--|--|
| <p>Définition1 [Atzori08]</p> | <p>“ L’Internet des objets peut être réalisé dans trois paradigme : Internet-orientés (intergiciel), orientées objets (des sondes) et sémantique-orientées (la connaissance), l’utilité d’IoT peut être lâchée seulement dans un domaine d’application où les trois paradigmes intersectés ”</p> |
| <p>Définition2 [Citic13]</p> | <p>“ L’Internet des objets (IoT) est une infrastructure mondiale pour la société de l’information, qui permet de disposer de services évolués en interconnectant des objets (physiques ou virtuels) grâce aux technologies de l’information et de la communication interopérables existantes ou en évolution ”</p> |

Tableau 1.1 – Définition d’Internet des objets

En réalité, la définition de l’Internet des objets n’est pas figée. Elle est basée sur des dimensions conceptuel et technique :

| | |
|---------------------------------------|--|
| <p>Point de vue conceptuel</p> | <p>L’Internet des objets caractérise des objets physiques connectés ayant leur propre identité numérique et capables de communiquer les uns avec les autres. Ce réseau crée en quelque sorte une passerelle entre le monde physique et le monde virtuel</p> |
| <p>Point de vue technique</p> | <p>l’IoT consiste l’identification numérique directe, et normalisée (adresse IP, protocoles smtp, http...) d’un objet physique grâce à un système de communication sans fil qui peut être une puce RFID, Bluetooth ou Wi-Fi. Cela rend l’énergie, le transport, les villes et beaucoup d’autres secteurs plus intelligents</p> |

Tableau 1.2 – Les dimensions conceptuel et technique [futura20]

En résumé les définitions précédentes :

L'IOT est un système qui englobe les dispositifs de l'informatique, de réseau et des machines. Capables de se connecter et de transmettre des données à tout moment, n'importe où et vers n'importe quel chemin / réseau, sans nécessiter d'interaction humaine.

2.2. Les composants de l'Internet des Objets

Selon [Gattal17], Le concept d'Internet des Objets exige la coordination des dispositifs suivants :

- ✚ Une étiquette physique identifie chaque objet ou bien une étiquette virtuelle identifie chaque lieu.
- ✚ Un dispositif mobile (téléphone cellulaire, ordinateur portable...) doté d'un logiciel additionnel, lit les étiquettes physiques ou localise les étiquettes virtuelles.
- ✚ Un réseau sans fil relie le dispositif portable à un serveur contenant l'information relative à l'objet étiqueté.
- ✚ Les informations sur les objets sont gérées dans des pages existantes du web.
- ✚ Un dispositif d'affichage (écran d'un téléphone mobile) permet de consulter les informations relatives à l'objet ou à un ensemble d'objets.

2.3. Evolution de l'internet des objets

Selon [Alain10] L'Internet des Objets est un « cyberspace » dans lequel des objets autonomes, dotés d'une intelligence propre, sont capables de s'auto organiser en fonction des circonstances, des contextes ou des environnements cette intelligence leur permet de coopérer entre eux pour atteindre une finalité. Cette collaboration s'effectue de façon ascendante (bottom-up) à partir des événements subis par les objets. La variété et la multiplicité des liens ou interactions entre ces éléments constituent un système complexe capable d'intégrer de nouveaux acteurs autonomes.

Les démarches linéaires n'ont plus d'utilité dans un tel système globalisé où des milliards d'événements de natures diverses se produisent simultanément : l'Internet des Objets

nécessite le développement.

Le tableau suivant indique l'évolution technologique applicable dans le domaine d'internet des Objets

| | Objectifs | Technologies anciennes | Technologies récentes |
|----------------------------------|--|---|---|
| Identification de l'objet | Reconnaître chaque objet individuel de façon unique. | Codes à barres RFID simples URL Coordonnées GPS | Solutions RFID complexes Puces optiques ADN |
| Capture de données | Recueillir des données environnementales pour enrichir les fonctionnalités du dispositif. | Thermomètre Hydromètre | Capteurs miniaturisés Nanotechnologies |
| Connexion | Connecter les systèmes entre eux. | Câbles | Bluetooth, WiFi |
| Intégration | Intégrer les systèmes pour transmettre les données d'une couche à l'autre. | Middlewares | Middlewares évolués |
| Traitement de données | Stocker et analyser les données pour lancer des actions ou pour aider à la prise de décisions. | Excel ERP (Programme de Gestion Intégrée) CRM (Gestion de la relation Client) | Datawarehouse Web sémantique |
| Transmission | Transférer les données dans les mondes physiques et virtuels | Internet Ethernet | EPCglobal |

Tableau 1.3 – L'évolution technologique applicable dans l'iot [Alin10]

2.4. Les applications actuelles de l'Internet des Objets

Citons quelques applications déjà opérationnelles :

- ✚ Suivi de l'acheminement des objets (courriers postaux, transports routiers)
- ✚ Connaissance de composition des produits « des produits sensibles 'pharmaceutiques' »
- ✚ Suivi de l'utilisation et de l'état des objets après leur acquisition par le client.
- ✚ Sécurité des transfusions sanguines (contrôle de la compatibilité).

Évoquons des objets expérimentaux plus originales.

- ✚ Le Web napperon (prototype développé par Érasme – département du Rhône) facilite l'accès des personnes âgées ou handicapées à des applications informatiques : On associe des étiquettes RFID à des objets familiers (photos, CD, boîtes de médicaments...) qu'il suffit de placer sur un lecteur, en forme de napperon, pour afficher sur un écran aux informations relatives à cet objet.
- ✚ La montre verte couplée à un téléphone mobile, capte des données environnementales (niveau sonore, taux d'ozone...) en différents lieux avant de les transmettre à un central qui les traite en temps réel.

2.5. Les domaines de l'Internet des Objets

L'IoT couvrira un large éventail d'applications et touchera quasiment à tous les domaines que nous affrontons au quotidien, ceci permettra l'émergence d'espaces intelligents. Parmi ces espaces intelligents, on peut citer [Hajaj18] :

1. Les Villes Intelligentes

Beaucoup de grandes villes ont été soutenues par des projets intelligents, comme Séoul, New York, Tokyo, Shanghai, Singapour, Amsterdam et Dubaï. Les villes intelligentes (voir Figure 3) peuvent encore être considérées comme des villes de l'avenir et la vie intelligente, et par le taux d'innovation de la création de villes intelligentes d'aujourd'hui, il sera devenu

très faisable pour entrer la technologie IoT dans le développement des villes. La demande exige une planification minutieuse à chaque étape, avec l'appui de l'accord des gouvernements, citoyens à mettre en œuvre la technologie d'Internet des objets dans tous les aspects. Par l'IoT, les villes peuvent être améliorées à plusieurs niveaux, en améliorant les infrastructures, en améliorant les transports... etc



Figure 3 – les constituants de la ville intelligente [Friess11]

2. Le Smart Grid

L'un des domaines d'application de l'IoT est le secteur de la distribution d'énergie intelligente, dit « Smart Grid » (voir figure 4). En France, ERDF est très actif dans le développement de ce domaine, où un besoin clair en récupération d'information à différents points du réseau électrique est devenu nécessaire pour une meilleure intégration des différentes sources d'énergies et une meilleure gestion de la distribution jusqu'aux utilisateurs finaux.

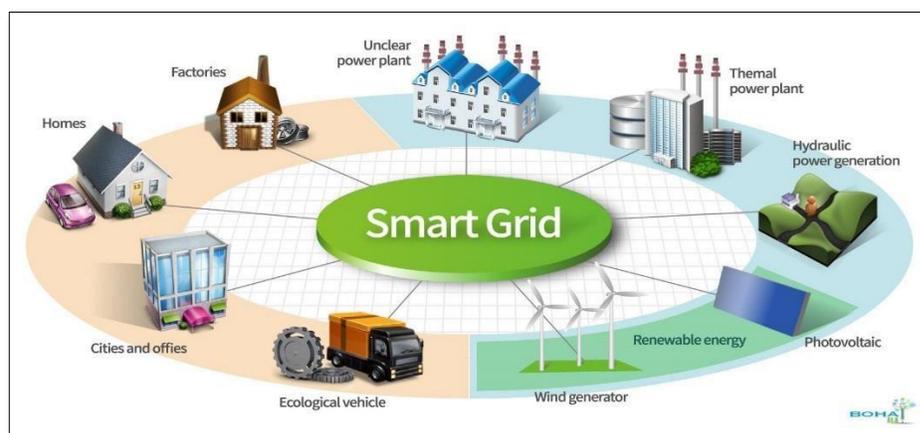


Figure 4 – les constituants de la smart grid [Thomas 19]

3. Les Appareils Intelligents

Des appareils intelligents (voir figure 5) dans les soins de santé sont utilisés pour stocker et gérer les paramètres de soins clés et pour gérer les données sur les maladies capturées. Ils sont principalement déployés pour fournir des solutions de conditionnement physique en suivant les activités ciblées et des dispositifs de diagnostic utilisés pour stocker des données de dispositifs. Principalement, ils sont utilisés comme des solutions de fitness pour suivi des activités du patient et des appareils de diagnostic intelligents tels que les Google verre.



Figure 5 – les constituants des appareils intelligents [Friess11]

4. Les Maisons et Les Bâtiments Intelligents

Les technologies Wi-Fi dans la domotique ont été principalement utilisées pour plusieurs raisons :

- ✓ Les appareils électroniques tels que les téléviseurs, les appareils mobiles, généralement pris en charge cette technologie.
- ✓ Le taux croissant d'adoption de dispositifs informatiques mobiles comme les téléphones intelligents, les tablettes.
- ✓ Les appareils mobiles garantissent que les consommateurs peuvent accéder aux « contrôleurs » des appareils portables connectés à un réseau.

A l'aide du concept de l'internet des objets, les maisons et les bâtiments peuvent exploiter alors de nombreux appareils et objets intelligemment (voir figure 6). Comme un exemple d'application intéressante de l'IoT dans les maisons intelligentes et les bâtiments on trouve : l'éclairage intelligent, le contrôle de l'air et de chauffage central, la gestion de l'énergie et la sécurité. Ainsi, les réseaux de capteurs sans fil avec intégration de la technologie de l'internet des objets fourniront une gestion intelligente de l'énergie dans les bâtiments. D'autre part, l'internet avec des systèmes de gestion de l'énergie aussi offre la possibilité d'accéder aux systèmes d'information et de contrôler l'énergie d'un bâtiment par un ordinateur portable ou un smartphone placé n'importe où dans le monde. [Friess11]

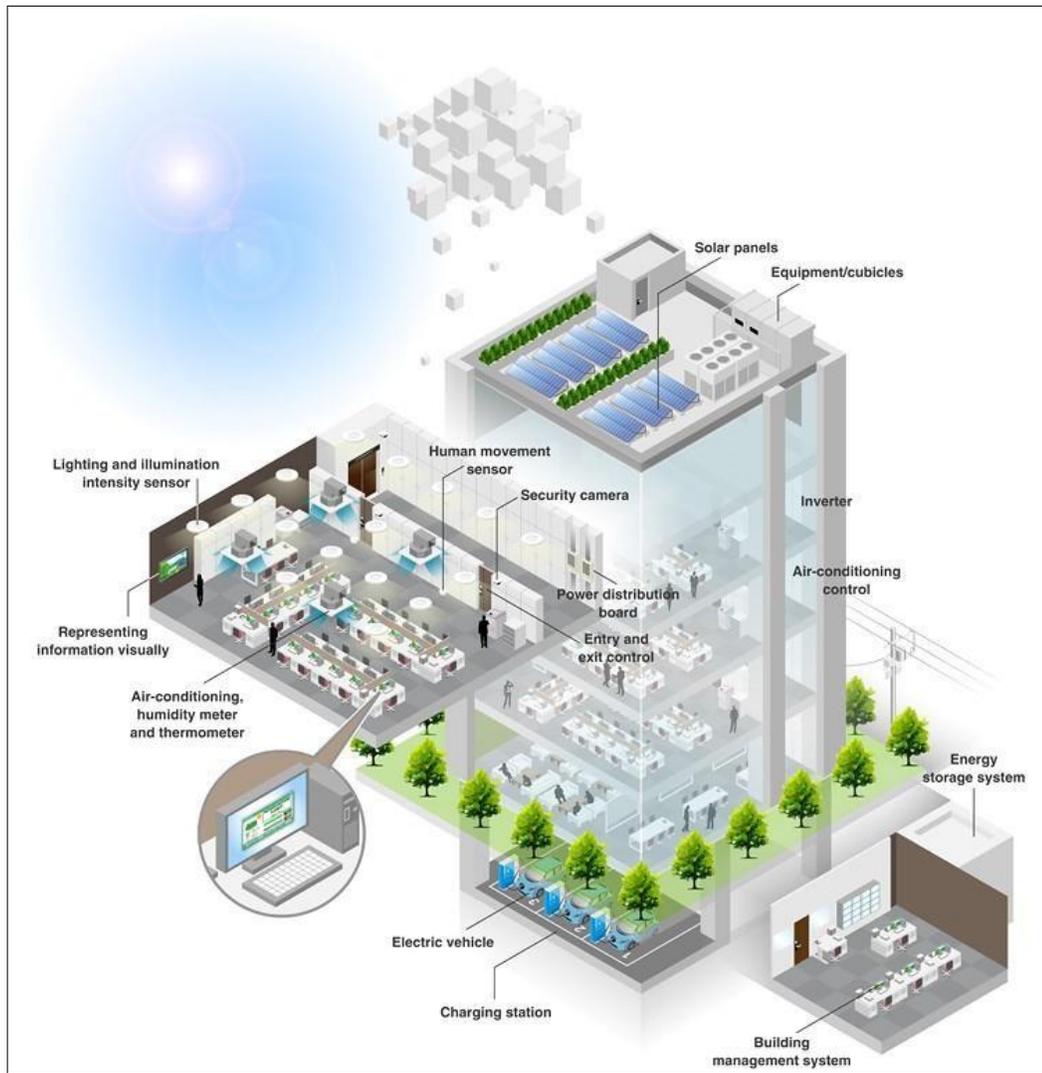


Figure 6 – les constituants des maison intelligents [Friess11]

5. Le Système De Santé Electronique

Dans le contexte de généralisation du traitement médical électronique, l'Internet des objets est fondamental. En effet, la conception d'un système intelligent de prise de décision clinique, matérialisé par le stockage des données collectées sur les patients et leur accessibilité universelle, procurerait au médecin un excellent appui durant la phase de traitement (voir figure 7). L'Internet des objets trouve donc tout son intérêt dans le domaine médical, et qui aussi peut améliorer le développement dans ce dernier.

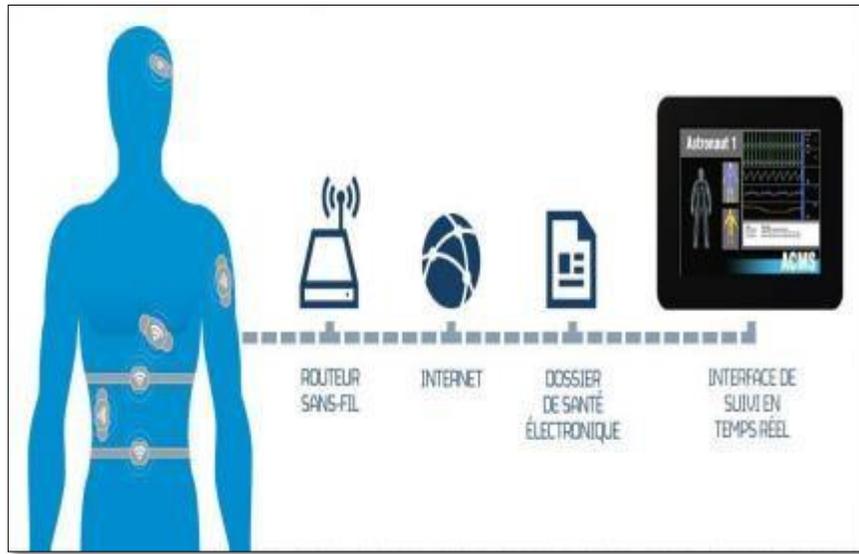


Figure 7 – un système de santé électronique [Friess11]

6. Le Transport et La Mobilité Intelligente

Le transport intelligent est confronté à deux conceptions principales ils sont l'analyse des transports et le contrôle des véhicules connectées. L'analyse de transport représente l'analyse de la prédiction de la demande et de détection anomalie. Le routage des véhicules et le contrôle de la vitesse en plus de la gestion du trafic sont tous connu comme le contrôle du transport qu'ils ont réellement étroitement lié au véhicule connecté, et globalement régie par la diffusion multi-technologie comme montre la figure 8

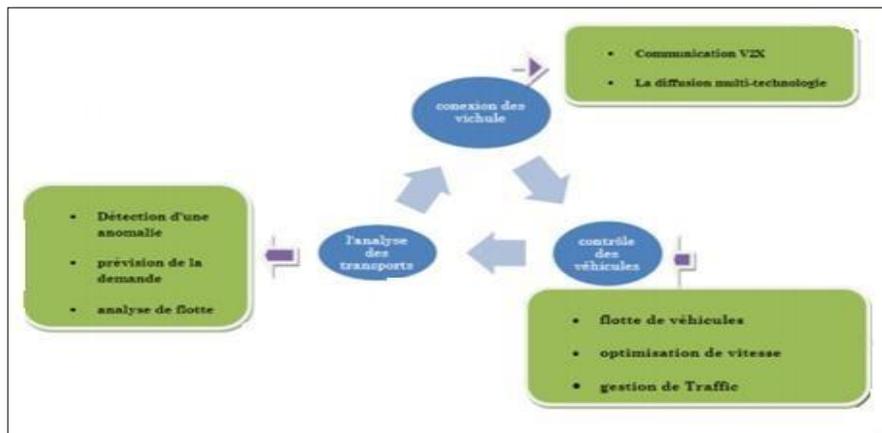


Figure 8 – les aspects de transport intelligent [Friess11]

7. Les Usines et La Fabrication Intelligente

L'usine intelligente a ajouté une nouvelle valeur dans la révolution de la fabrication en intégrant l'intelligence artificielle et la communication M2M avec le processus de fabrication. La révolution des industries et de la fabrication est devenue l'une des plus technologies développées de nos jours, la croissance de l'évolution de l'industrie a pris de nombreuses générations. L'industrie intelligente c'est la quatrième génération connue par l'industrie 4.0 est basée sur les systèmes de chiffrement physiques qui est capables de se connecter à Internet. Le concept de l'industrie 4.0 avec l'internet des objets peut atteindre de grandes attentes pour les accords de résolution des industries avec de nombreux aspects sont illustrés à la figure 9.

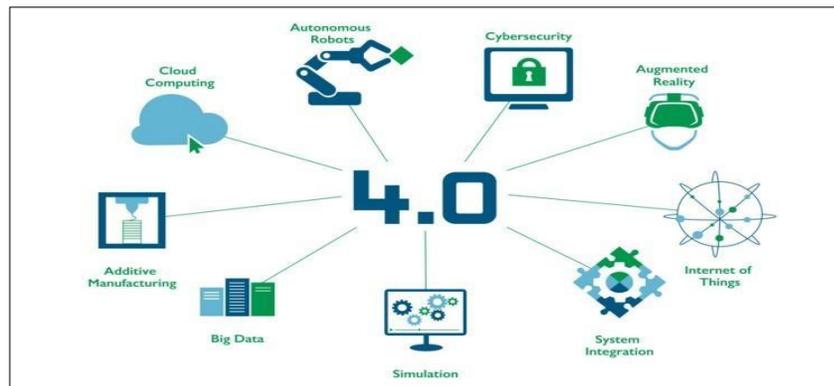


Figure 9 – un schéma de l'industrie 4.0

8. La Surveillance à Distance Des Patients

Ce domaine d'application est déployé pour surveiller à distance les paramètres essentiels du patient par l'utilisation de capteurs, de dispositifs et les objets qui les entourent (voir figure 10). En cela, les données critiques du patient sont transmises et partagées en temps réel entre le patient et les soignants. Sa principale pertinence est la gestion des maladies chroniques telles que le diabète, les maladies cardiaques... etc. [Alok14]



Figure 10 – le mécanisme de la surveillance distante des patients [Alok14]

3. CONCLUSION

Dans l'étude présentée dans ce premier chapitre, nous pouvons conclure que L'internet des objets est une nouvelle vision apparue suite à l'évolution de l'informatique. Elle vise en effet à fusionner les environnements informatiques avec le monde réel. L'objectif principal de cette vision est de fournir aux utilisateurs des services à forte valeur ajoutée de manière naturelle et transparente.

Les développeurs d'applications dans l'iot doivent répondre à des exigences qui rendent leurs tâches particulièrement difficiles et complexes. Ces exigences résultent de la nature des environnements ubiquitaires (hétérogène, dynamique, distribué, ouvert)

Les Ontologies & Le Smart Home

2.1. Introduction

Le terme « ontologie » est utilisé depuis le début des années 1990, et son champ d'application s'élargit considérablement.

Il est prévu que des applications internet pourront mener des raisonnements utilisant les connaissances stockées.

Dans ce chapitre, nous introduit les différentes définitions qui ont été attribuées à la notion d'ontologie. Nous montrerons aussi le but de l'utilisation des ontologies dans le domaine du Web sémantique ainsi que leur place dans les systèmes à bases de connaissances. Ensuite, nous montrerons le processus et les méthodologies servant leur construction, ainsi que les principaux de la maison intelligente .

2.2. Notion d'ontologie

Selon Bourougaa [Bourougaa 19], Introduit en Intelligence Artificielle (IA) il y a 23 ans, le terme d'ontologie est cependant usité en philosophie depuis le XIXème siècle. Dans ce domaine, L'ontologie est une étude de l'être en tant qu'être, c'est-à-dire, une étude des propriétés générales de ce qui existe.

C'est à l'occasion de l'émergence de l'Ingénierie des Connaissances que les ontologies sont apparues en IA, comme réponses aux problématiques de représentation et de manipulation des connaissances au sein des systèmes informatiques.

2.2.1. Définitions

Selon Bourougaa [Bourougaa 19], Le terme « ontologie » est employé dans des contextes très différents touchant la philosophie, la linguistique ou l'IA. Sachant que chaque définition est spécifier sa domaine Pour cela les définitions dépendent des domaines spécifiques :

| | |
|---|--|
| Définition1 [Neches91] | <i>“ une ontologie définit les termes et les relations de base du vocabulaire d’un domaine ainsi que les règles qui indiquent comment combiner les termes et les relations de façon à pouvoir étendre le vocabulaire “</i> |
| Définition2 [Gruber93] | <i>“ une ontologie est une spécification explicite d’une conceptualisation “</i> |

La définition de Gruber est la plus utilisée dans la littérature. Elle a été modifiée par Borst :

| | |
|--|---|
| Définition3 [Borst97] | <i>“ une ontologie est une spécification explicite et formelle d’une conceptualisation partagée “</i> |
|--|---|

Le terme « conceptualisation » réfère dans cette définition à une abstraction d’un phénomène du monde. Le terme « formelle » indique que les ontologies sont interprétables par la machine. Cependant, « spécification explicite » signifie que les concepts de l’ontologie et les contraintes liées à leur usage sont définis de façon déclarative. Enfin, le terme « partagé » signifie que l’ontologie capture la connaissance consensuelle. Mais cette définition laisse la porte ouverte à de nombreuses définitions.

| | |
|--|---|
| Définition4 [Grüninger95] | <i>“ une ontologie est une description formelle d’entités et leurs propriétés, relations, contraintes, comportement “</i> |
|--|---|

Cette définition de Grüninger est simplifiée dans [Ikeda97] où une ontologie est définie comme un ensemble de définitions de concepts et leurs relations. A ne pas confondre avec un modèle qui est un ensemble d’instances de ces concepts.

| | |
|--|---|
| Définition5 [Guarino97] | <i>“ Les ontologies sont des spécifications partielles et formelles d’une conceptualisation commune “</i> |
|--|---|

2.2.2. Que représente-t-on dans une ontologie ?

Les ontologies produisent un vocabulaire commun d'un domaine et définissent, de façon plus ou moins formelle, la signification des termes et des relations entre eux. Les connaissances intégrées dans les ontologies sont formalisées en mettant en jeu cinq types de composants : [Gomez99].

- **Concepts** : Un concept (termes ou classes de l'ontologie) est un constituant de la pensée (un principe, une idée, une notion abstraite) sémantiquement évaluable et communicable. Ces concepts peuvent être classifiés selon plusieurs dimensions :
 - ✓ Niveau d'abstraction (concret ou abstrait).
 - ✓ Atomicité (élémentaire ou composée).
 - ✓ Niveau de réalité (réel ou fictif).
- **Relations** : Représentent des associations existantes entre les concepts d'un domaine et elles se définissent formellement à partir d'un produit de n concepts :

$$R : C_1 \times C_2 \times \dots \times C_{n-1} \rightarrow C_n$$

Voici quelques relations les plus courantes dans la littérature :

1. **L'équivalence** : R est une relation d'équivalence si et seulement si : R est symétrique, réflexive et transitive.
2. **La cardinalité** : c'est le nombre possible de relations de ce type entre les mêmes concepts (ou instances de concept).
3. **L'incompatibilité** : Deux relations sont incompatibles si elles ne peuvent lier les mêmes instances de concepts.
4. **L'inverse** : Deux relations binaires sont inverses l'une de l'autre si, quand l'une lie deux instances I1 et I2, l'autre lie I2 et I1.
5. **L'exclusivité** : Deux relations sont exclusives si, quand l'une lie des instances de concepts, l'autre ne lie pas ces instances, et vice-versa.

- **Fonctions** : ce 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 :

$$F : C_1 \times C_2 \times \dots \times C_{n-1} \rightarrow C_n$$

- **Axiomes** : constituent des assertions, acceptées comme vraies, à propos des abstractions du domaine, traduites par l'ontologie. Ils ont pour objectif de représenter des concepts et des relations dans un langage logique permettant de représenter leur sémantique. Ils représentent les intentions des concepts et des relations du domaine et, de manière générale [Staab00].
- **Instances** : elles constituent la définition extensionnelle de l'ontologie ; ces objets véhiculent les connaissances (statiques, factuelles) à propos du domaine du problème.

2.2.3. Différentes sortes d'ontologies

Nous pouvons classer les ontologies selon plusieurs dimensions. Parmi celles-ci, nous en examinerons deux :

1) Objet de conceptualisation

Dans [Gomez99], Les ontologies sont classifiées selon leur objet de conceptualisation de la façon suivante :

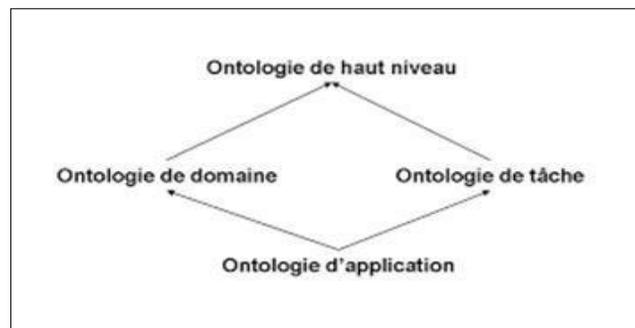


Figure 2.1 : Les différents sortes d'ontologie [Bourougaa19]

- **Ontologie de haut niveau** : décrit des concepts très généraux comme l'espace, le temps, la matière, les objets, les événements, les actions, etc. Ces concepts ne dépendent pas d'un problème ou d'un domaine particulier.
- **Ontologie de domaine** : Contrairement aux ontologies de haut niveau, les ontologies de domaine sont plus spécifiques. Elles synthétisent les connaissances spécifiques à un domaine particulier. Elles décrivent le vocabulaire ayant trait à un domaine générique (ex. : l'enseignement, la médecine) .
- **Ontologie de tâches** : Ce type d'ontologies est utilisé pour conceptualiser des tâches spécifiques dans les systèmes, telles que les tâches de diagnostic, de planification, de conception, de configuration, de tutorat.
- **Ontologie d'application** : Cette ontologie est la plus spécifique, elle contient des concepts dépendants d'un domaine et d'une tâche particuliers, qui sont généralement subsumés par des concepts de ces deux ontologies. Ces concepts correspondent souvent aux rôles joués par les entités du domaine lors de l'exécution d'une certaine activité [Guarino98].

2) Niveau de formalisme de représentation

Selon le niveau du formalisme de représentation, [Ushold96] propos une classification comprenant quatre catégories :

- **Informelles** : ontologies opérationnelles dans un langage naturel.
- **Semi-informelles** : utilisation d'un langage naturel structuré et limité.
- **Semi-formelles** : langage artificiel défini formellement.
- **Formelles** : utilisation d'un langage artificiel contenant une sémantique formelle, ainsi que des théorèmes et des preuves de propriétés telles la robustesse et l'exhaustivité [Gomez99].

2.2.4. Les ontologies et les systèmes à bases de connaissances

Les ontologies sont apparues en informatique, plus précisément en Ingénierie des Connaissances, dans le cadre des démarches d'acquisition des connaissances pour les systèmes à base de connaissances (SBC). Les SBC proposaient alors de spécifier, d'un côté des connaissances du domaine modélisé, et de l'autre, des connaissances de raisonnement qui manipule et utilise ces connaissances du domaine. L'idée de cette séparation modulaire était de construire mieux et plus rapidement des SBC en réutilisant le plus possible des composants génériques, que ce soit au niveau du raisonnement ou des connaissances du domaine [Bachimant03].

2.2.5. Les ontologies et le web sémantique

Le Web sémantique [Lee02] a alors l'ambition de lever cette difficulté en associant aux ressources du Web des entités ontologiques comme références sémantiques, ce qui permettra aux différents agents logiciels d'accéder et d'exploiter directement le contenu des ressources et de raisonner dessus. Ce référencement sémantique peut aussi résoudre les problèmes d'interprétation des ressources informationnelles provenant des applications hétérogènes. [Ushold02].

L'architecture du Web sémantique repose sur une hiérarchie des langages d'assertion et de description d'ontologies ainsi que sur un ensemble de services pour l'accès aux ressources au moyen de leurs références sémantiques. [Oberle04].

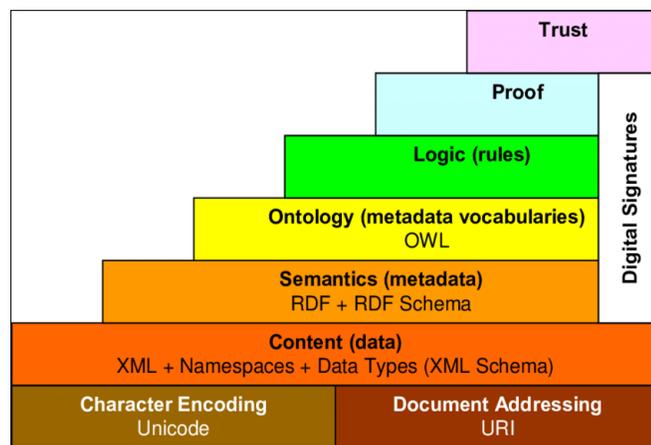


Figure 2.2 : Les couches du Web sémantique[BS19]

2.3. Les ontologies : Différents besoins

Les ontologies sont utilisées dans plusieurs domaines, les plus répandus sont :

1) **Communication :**

Les humains peuvent communiquer s'ils ont des connaissances ou des points de vue partagés. Ces dernières peuvent être obtenues si le domaine est décrit sans confusion terminologique ou conceptuelle pour être compris de la même façon par tout le monde.

2) **Interopérabilité :**

L'interopérabilité implique la possibilité de pouvoir demander et recevoir des services entre des systèmes interopérables.

3) **Ingénierie des systèmes :**

Le développement des systèmes basé sur les ontologies a donné un profit à l'ingénierie :

- **Réutilisabilité :** l'ontologie encode les informations relatives à un domaine de sorte que le partage et la réutilisation sont possibles.
- **Acquisition des connaissances :** Elle guide l'acquisition des connaissances.
- **Sûreté :** Elle rend l'automatisation du processus de vérification de consistance.
- **Spécification :** Aide le processus d'identification et de spécification des besoins.

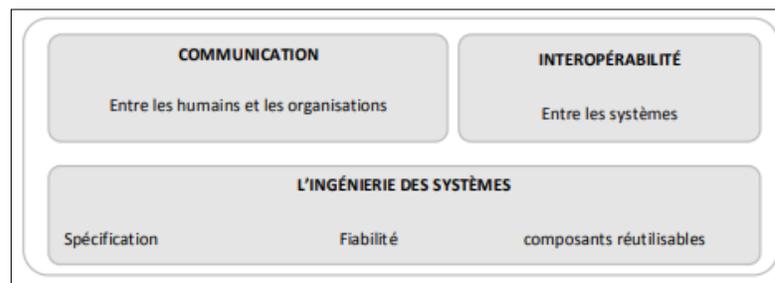


Figure 2.3 : Domaines d'utilisation des Ontologies.

2.4. Construire des ontologies

Le processus de construction d'une ontologie est une collaboration qui réunit des experts du domaine de connaissance, des ingénieurs de la connaissance, voire les futurs utilisateurs de l'ontologie. Cette collaboration ne peut être fructueuse que si les objectifs du processus ont été clairement définis, ainsi que les besoins qui en découlent. La Figure 4 représente le processus de construction d'ontologie.

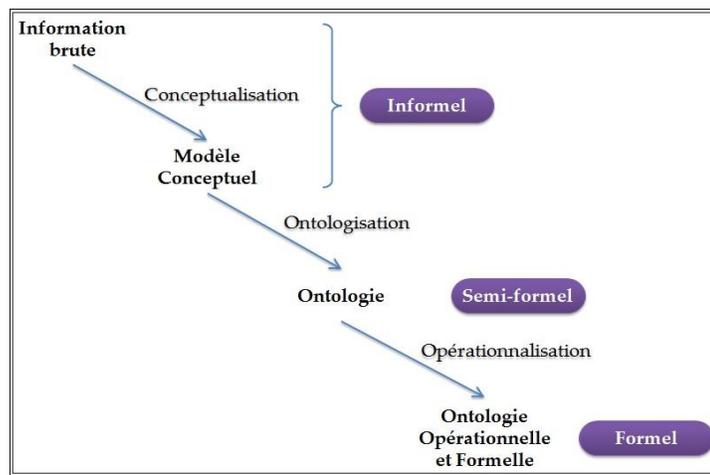


Figure2.4 : Processus de construction d'ontologie [Bourougaa19]

1) Evaluation des besoins

Le but visé par la construction d'une ontologie se décline en 3 aspects :

1. **L'objectif opérationnel** : il est indispensable de bien préciser l'objectif opérationnel de l'ontologie, en particulier à travers des scénarios d'usage.
2. **Le domaine de connaissance** : il doit être délimité aussi précisément que possible.
3. **Les utilisateurs** : ils doivent être identifiés autant que faire se peut, ce qui permet de choisir, en accord avec l'objectif opérationnel.

2) Conceptualisation

Cette étape permet d'aboutir à un modèle informel, donc sémantiquement ambiguë et généralement exprimé en langage naturel. Elle consiste, à partir des données brutes, à dégager les concepts et les relations entre ces concepts permettant de décrire de manière informelle les entités cognitives du domaine.

3) Ontologisation

L'ontologisation consiste en une formalisation partielle, sans perte d'information, du modèle conceptuel obtenu dans l'étape précédente. Ce qui permet de faciliter sa représentation ultérieure dans un langage complètement formel et opérationnel.

4) Opérationnalisation

Cette étape consiste à formaliser complètement l'ontologie obtenue dans un langage de représentation de connaissances formel (i.e. possédant une syntaxe et une sémantique) et opérationnel (i.e. doté de services inférentiels permettant de mettre en œuvre des raisonnements), par exemple, le modèle des Graphes Conceptuels ou la Logique de Descriptions.

2.5. Méthodologies de construction d'ontologies

Les méthodologies peuvent porter sur l'ensemble du processus et guider l'ontologiste dans toutes les étapes de la construction. Les méthodologies les plus connus sont :

1) TOVE :

TOVE (Toronto Virtual Enterprise) développé par l'université de Toronto, cette méthodologie repose sur les expériences de développement d'une entreprise [Gruber95, Uschold96].

Elle s'appuie également, pour le développement d'une ontologie, sur les principales étapes suivantes :

- Capturer des scénarios de motivations

- Spécifier la terminologie de l'ontologie
- Evaluer la complétude de l'ontologie.
- Formuler des questions de compétences informelles

2) ENTERPRISE :

Uschold [Uschold96], propose le squelette d'une méthode basé sur l'expérience de construction d'ontologies dans le domaine de la gestion des entreprises. La méthode ENTERPRISE repose sur les quatre étapes suivantes :

- Evaluer l'ontologie.
- Identifier le rôle et la portée de l'ontologie.
- Identifier les concepts et relations fondamentaux et des définitions provisoires.
- Rédiger une documentation et une trace des actions réalisées lors des différentes phases.

3) METHONTOLOGY :

La méthodologie de construction d'ontologies « METHONTOLOGY » se situe entre le GL (Génie Logiciel) et l'IC (Ingénierie des Connaissances). Elle identifie une séquence d'activités techniques à appliquer pour le développement de l'ontologie. L'approche METHONTOLOGY distingue les étapes suivantes :

1. Spécification :

Le développement d'une ontologie commence par la définition du domaine et portée de celle-ci. Cela est basé sur la réponse à certaines questions : Quel est le domaine que l'ontologie va couvrir ? À quoi cette ontologie va servir ?? Qui va utiliser et maintenir l'ontologie ? etc. [Gruber95].

2. Conceptualisation :

Elle consiste à identifier et à structurer les connaissances du domaine, à partir des sources d'informations.

3. Implémentation :

Cette étape consiste à formaliser le modèle conceptuel obtenu dans l'étape précédente par un formalisme de représentation d'ontologie telles que les logiques de description. Puis, à coder l'ontologie dans un langage d'ontologie formel.

4. Maintenance :

Cela peut s'agir d'une maintenance correctrice ou évolutive de l'ontologie (nouveaux besoins de l'utilisateur), ce qui permet la validation et l'évolution de celle-ci.

2.6. Formalismes de représentation

Représenter des connaissances propres à un domaine particulier consiste à décrire et à coder les entités de ce domaine de manière à ce qu'une machine puisse les manipuler afin de raisonner. Comme alternative à la logique classique, l'IA a proposé divers formalismes de représentation : ceux qui ont été le plus utilisés pour représenter les ontologies sont :

1. Frames :

Le formalisme frames est introduit par M. Minsky [Minsky75]. Dans ce formalisme, la structure de données enregistrement représente une situation et un objet. L'idée est de collecter toutes les informations nécessaires concernant une situation et de les mettre dans une place, appelée frame. [Gruber95].

2. Graphes conceptuels :

Le modèle des Graphes Conceptuels (GC), introduit par J. SOWA [Sowa84] au début des années 80, est un modèle opérationnel de représentation de connaissances, qui appartient à la famille des réseaux sémantiques.

Il y a deux sortes d'arcs : **les arcs de propriété** qui affectent les propriétés à des concepts ou à des individus **et les arcs IS-A** qui introduisent les relations hiérarchiques entre des concepts ou entre des individus.

3. Logique de description LD :

L'objectif principal des LDs consiste à pouvoir raisonner efficacement pour minimiser les temps de réponse. Par conséquent, la communauté scientifique a publié de nombreuses recherches qui portent sur l'étude du rapport expressivité/performance des différentes LDs [Nardi03].

2.7. Les niveaux de description :

La modélisation des connaissances d'un domaine avec les LDs se réalise en deux niveaux. Le premier, le niveau terminologique ou TBox, décrit les connaissances générales d'un domaine alors que le second, le niveau assertionnel ou ABox, représente une instantiation spécifique.

| TBox | ABox |
|---|-------------------------------------|
| Femelle \sqsubseteq T \sqcap \neg Mâle | Humain(Ane) Femelle(Ane) |
| Mâle \sqsubseteq T \sqcap \neg Femelle | Femme(Sophie) |
| Animal \equiv Mâle \sqcup Femelle | Humain(Robert) |
| Humain \sqsubseteq Animal | \neg Femelle(Robert) |
| Femme \equiv Humain \sqcap Femelle | Homme(David) |
| Homme \equiv Humain \sqcap \neg Femelle | relationParentEnfant(Sophie, Anne) |
| Mère \equiv Femme \sqcap \exists relationParentEnfant | relationParentEnfant(Robert, David) |
| Père \equiv Homme \sqcap \exists relationParentEnfant | |
| MèreSansFille \equiv Mère \sqcap \forall relationParentEnfant. \neg Femme | |
| relationParentEnfant \sqsubseteq T _R | |

Tableau 2.1 : Une base de connaissances composée d'une TBox et d'une ABox

1. Le niveau terminologique (TBox)

Les concepts atomiques et rôles atomiques constituent les entités élémentaires d'une TBox. Les noms débutant par une lettre majuscule désignent les concepts, alors que ceux débutant par une lettre minuscule dénomment les rôles (par exemple : les concepts Femelle, Mâle, Homme et Femme, et le rôle relationParentEnfant).

2. Le niveau assertionnel (ABox)

Une ABox contient un ensemble d'assertions sur les individus. Chaque ABox doit être associée à une TBox, car les assertions s'expriment en terme de concepts et de rôles de la TBox. Une ABox désigne des individus caractérisés par des assertions d'individus nommés.

3. L'inférence :

L'inférence s'effectue au niveau terminologique ou assertionnel (factuel) :

- **L'inférence au niveau terminologique comprend quatre principaux problèmes** : la satisfiabilité, la Subsumption, l'équivalence, et la disjonction.
- **L'inférence au niveau assertionnel comprend quatre principaux problèmes aussi**: la Cohérence, la vérification d'instance, La vérification de rôle, et le problème de récupération. [Baader03a]

2.8. Outils de développement d'ontologies

Les outils de développement d'ontologies qui existent sur le marché aujourd'hui sont divers et variés. Cet état de choses suscite beaucoup d'interrogations lorsque vient le moment d'en choisir un pour construire une nouvelle ontologie [Gomez-Pérez02] : L'outil offre-t-il une assistance au développement ? L'outil dispose-t-il d'un moteur d'inférence ? Quels langages d'ontologies l'outil supporte-t-il ? L'outil permet-il d'importer/exporter des ontologies ? L'outil offre t-il un support à la réutilisation d'ontologies existantes ? L'outil permet-il de documenter les ontologies construites ? L'outil offre-t-il un support graphique à la construction des ontologies ? L'outil est-il stable, convivial, « mature » ?

1) Langages de spécification d'ontologies :

Dans le contexte du Web sémantique, plusieurs langages d'ontologies ont été développés pendant les dernières années. Certains d'entre eux sont basés sur la syntaxe de XML, tels que XOL, SHOE, OML, RDF, et RDF Schéma. Les deux derniers sont créés par des groupes de travail du W3C. Trois autres langages sont établis sur RDF(S) pour améliorer ses caractéristiques : OIL, DAML+OIL et OWL. La Figure 5 représente la pyramide des langages du Web sémantique.

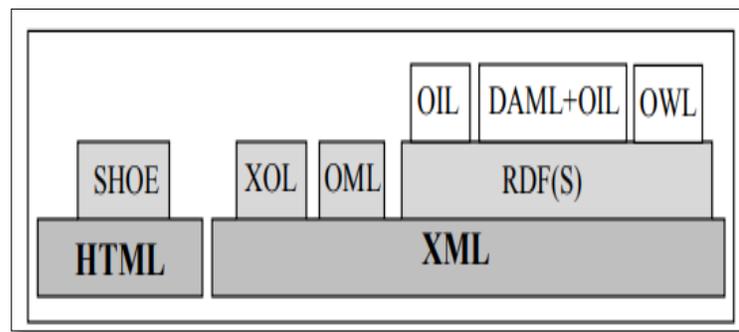


Figure 2.5 : La pyramide des langages du Web sémantique

1. RDF:

RDF [W3C04a] est un langage pour la représentation de métadonnées à propos des ressources. Le modèle RDF permet cette représentation par des assertions sous la forme d'un triplet (ressource, propriété, valeur), ou encore (sujet, prédicat, objet) :

- **Ressources** : les ressources sont tous les objets *décrits* par RDF. Ces ressources peuvent être aussi bien des pages Web que tout objet ou personne du monde réel. Les ressources sont alors identifiées par leur URI (Uniform Resource Identifier).
- **Propriétés** : est un attribut, un aspect, une caractéristique qui s'applique à une ressource. Il peut également s'agir d'une mise en relation avec une autre ressource.
- **Valeurs** : les valeurs en question sont les valeurs particulières que prennent les propriétés. La valeur pouvant être une autre ressource ou bien un littéral.

```
<rdf :RDF>
<rdf :Description about='Imen'>
<rdf :Property about='ville'> Tébessa </rdf :Property>
<rdf :Property about='age'> 24 </rdf :Property>
</rdf :Description>
</rdf :RDF>
```

Figure 2.6 : Exemple : Imen a 24 ans et habite Tébessa

2. RDF(S) :

RDFS [W3C04b] est un langage permettant de définir des schémas de méta-données. Il définit le sens, les caractéristiques et les relations d'un ensemble de propriétés. La principale nouvelle notion est la distinction entre une classe (concept d'une ontologie) et une instance (individu d'une ontologie). Quelques notions définies sont : (rdfs : Class), (rdfs : subclassOf), (rdfs : domain), et (rdfs : range).

Sur l'exemple de Sami, nous définissons le concept de personne, une taxinomie de concepts, et l'instance Imen.



Figure 2.7 : Taxinomie de concept personne

3. OWL :

OWL [W3C04c] est un langage fondé sur la syntaxe RDF/XML et héritier des travaux de DAML+OIL. OWL introduit l'aspect sémantique qui manque RDF, et offre, par ses primitives plus riches, au machine une capacité d'interprétation plus grande que celle de RDF et RDFS.

2) Les moteurs d'inférences :

La plupart de ces moteurs acceptent en entrée des fichiers OWL et sont conçus pour raisonner sur les logiques de descriptions. Une fois l'ontologie chargée, ces moteurs effectuent les inférences sur la TBox et la ABox. Les principaux moteurs d'inférence pour les logiques de description sont : Racer [Haarslev01], Pellet [Sirin06].

1. Racer :

Racer est le moteur d'inférence le plus connu et l'un des plus utilisés dans le domaine pour ces performances et sa stabilité. Il est fondé en 2004 par Volker Haarslev, Kay Hidde, Ralf Möller et Michael Wessel qui travaillaient à l'université de Hambourg. Racer travaille sur les ontologies modélisées par son langage, mais il accepte des ontologies décrites en RDF ou OWL.

Racer possède quelques avantages :

- ✓ Racer permet l'ajout d'assertions et d'individus dans les ABox après le chargement de l'ontologie.
- ✓ Racer permet l'utilisation de règles SWRL.

Racer possède quelques points négatifs :

- Racer suppose que toutes les propriétés sur les datatypes sont fonctionnelles (pas de valeurs multiples pour un datatype property).
- Racer ne permet pas l'utilisation de type de défini par l'utilisateur, et il n'existe pas de version libre d'utilisation. Cependant il est possible d'obtenir une licence gratuite dans le cadre de la recherche scientifique.

2. Pellet :

Le moteur Pellet [Sirin06] est beaucoup plus récent. Pellet est un des projets du MINDSWAP Group, un groupe de recherche sur le web sémantique de l'université du Maryland. Il est disponible en Open Source et offre des évolutions fréquentes. Pellet travaille sur des ontologies décrites en RDF ou OWL et permet les requêtes avec RDQL et SPARQL sur la ABox et la TBox.

Pellet possède quelques avantages :

- ✓ Pellet est open-source et développé en Java.
- ✓ Pellet est un raisonneur OWL DL complet.
- ✓ Pellet propose en cas d'incohérence dans l'ontologie des réparations possibles.

Pellet possède quelques points négatifs :

- Pellet possède une documentation pauvre en comparaison de celle de Racer. En effet racer est le plus utilisé et donc le plus documenté par des particuliers.
- Pellet n'offre pas de système de souscription à un concept.

2.9. Langages d'interrogation d'ontologies :

Dans cette section, nous présentons les trois langages d'interrogation basés sur la reconnaissance de graphe RDF : RDQL, SPARQL et nRQL. Le principe est de décrire un graphe RDF à l'aide de variables. RDQL et SPARQL sont les langages d'interrogation utilisables sur Pellet et nRQL est le langage d'interrogation de Racer.

1. RDQL :

RDQL (RDF Data Query Language) [W3C04d] est un langage d'interrogation de données définies en RDF. Ce langage n'est pas standardisé, il existe de nombreuses implémentations. Sa syntaxe est très proche de SQL (Structured Query Language) :

```
SELECT variable [, variable]*  
FROM documents_rdf [, documents_rdf]*  
WHERE modele_de_triplets  
AND  
restrictions_booléennes  
USING  
définition_des_raccourcis
```

- La clause **SELECT** définit la liste des variables que l'on désire obtenir. Une variable est composée de caractères alphanumériques et commence par un '?'.
• La clause **FROM** définit l'emplacement des documents RDF utilisés pour la requête.
• La clause **WHERE** définit le triplet RDF (sujet - prédicat - objet), les éléments de ce triplet sont décrits soit par les valeurs de l'ontologie interrogée soit par des variables.
• La clause **AND** définit les restrictions booléennes de la requête. Une restriction booléenne est constituée de valeurs ou variables composé à l'aide d'opérateurs .

Voici un exemple de requête RDQL, qui sélectionne l'âge d'une personne dans le document `universite_tebessa.rdf` sachant que la personne possède un âge inférieur ou égal à 24 ans, que son nom est Imen et son adresse email est xxx@gx.com.

```
SELECT ?name, ?email, ?age  
FROM <http://www.universite_tebessa.rdf> WHERE (  
?email, vcard:EMAIL, "xxx@gx.com") AND (?age <=  
24) && (?name EQ "Imen") ) WHERE (?email, vcard  
:EMAIL, "xxx@gx.com")  
USING vcard FOR <http://www.w3.org/2001/vcard-rdf/3.0/>
```

2. SPARQL :

SPARQL [W3C06] est une amélioration de RDQL, ce langage est en cours de standardisation au niveau du W3C. SPARQL offre une syntaxe quasi identique à RDQL, mais y ajoute notamment les opérateurs **UNION** et **OPTIONAL** dans la clause **WHERE**. L'opérateur **UNION** définit la disjonction de triplets RDF.

3. nRQL :

nRQL (new Racerpro query Language) [Haarslev01] est le langage d'interrogation de Racer. Comme RDQL et SPARQL, nRQL est basé sur la recherche de graphes RDF. Sa syntaxe est proche des deux autres, sauf pour sa notation préfixée des opérateurs. Ci-dessous, nous présentons une requête en nRQL cherchant tous les oncles (variable z) dans l'ontologie myOntology.

```
RETRIEVE ($ ?z)
(AND ( $ ?x $ ?y | myOntology#aEnfant |
($ ?z $ ?x | myOntology#estFrereDe |))
```

2.10. Editeur d'ontologies

Aujourd'hui, le nombre d'éditeurs d'ontologies a considérablement augmenté. Il existe différents éditeurs qui offrent différentes fonctionnalités. Parmi ces outils on trouve :

1. OILEd

OILEd [Bechhofer01] a été conçu pour éditer des ontologies dans le langage de représentation OIL, il est souvent considéré comme une simple interface de la logique de description SHIQ. Cet éditeur offre également les services d'un raisonneur, FaCT qui permet de tester la satisfiabilité des définitions de classes et de découvrir des subsumptions restées implicites dans l'ontologie. L'outil dispose de mécanismes pour la classification et le contrôle de la cohérence des ontologies.

2. OntoEdit

OntoEdit [Sure02] est un environnement de construction d'ontologies indépendant de tout formalisme. Des outils graphiques d'ontologies sont inclus dans l'environnement. ONTOEDIT intègre un serveur destiné à l'édition d'une ontologie par plusieurs utilisateurs. Un contrôle de la cohérence de l'ontologie est assuré à travers la gestion des ordres d'édition.

3. Protégé2000

Protégé2000 [Noy01] est une interface modulaire, développée au Stanford Medical Informatics de l'Université de Stanford⁷, permettant l'édition, la visualisation, le contrôle (vérification des contraintes) d'ontologies, l'extraction d'ontologies à partir de sources textuelles, et la fusion semi-automatique d'ontologies. Le modèle de connaissances de Protégé2000 est issu du modèle des frames et contient des classes (concepts), des slots (propriétés) et des facettes (valeurs des propriétés et contraintes), ainsi que des instances des classes et des propriétés. De nombreux plug-ins sont disponibles ou peuvent être ajoutés par l'utilisateur.

2.11. La maison intelligente

La maison intelligente ou en anglais Smart Home est l'exemple **d'environnement ubiquitaire** auquel nous nous intéressons dans cet mémoire. Son objectif principal est d'améliorer la qualité de vie des habitants en fournissant des services à haute valeur ajoutée.

Dans cette section, nous présentons les définitions de la maison intelligente proposées dans la littérature. Ensuite, nous discutons ses défis, ses domaines d'applications et ses caractéristiques.

3.1. Définition

| | |
|--------------------------------|--|
| [Anne92; Galliot12] | "Une maison sera disqualifiée au regard du classement dans la catégorie des maisons intelligentes si : - l'information ne peut pas circuler librement de l'intérieur de la maison vers le monde extérieur, et vice-versa ; - si la maison fonctionne avec des ordinateurs intégrés qui ne peuvent pas se parler entre eux ; - si elle est équipée avec des fonctions sophistiquées difficiles à utiliser. " |
| [Aldrich03] | " une résidence équipée de technologies informatiques qui anticipent et répondent aux besoins de ses occupants en essayant de promouvoir leur confort, leur bien-être, leur sécurité et leur détente grâce à la gestion de la |

| | |
|---------------------|---|
| | technologie à l'intérieur de la maison et des connexions avec le monde extérieur. ‘’ |
| [Froncois05] | ‘’ une maison qui dispose de fonctionnalités susceptibles de simplifier la vie de ses habitants au quotidien, de réaliser des économies d'énergie et d'apporter un certain niveau de confort et de sécurité. Elle est ouverte aux évolutions futures par la nature même de ses infrastructures de câblage et par son ouverture au monde numérique. ‘’ |

3.2. Les défis

Toutefois, avant que le concept de la maison intelligente soit implanté en situation réelle, un ensemble de défis doivent être relevés. Ces défis ont été au cœur des travaux de plusieurs chercheurs.

Dans [Edwards01], l'auteur présente 2 classes des défis couvrent principalement les problèmes qui résultent de la manière attendue des maisons intelligentes à être déployées et habitées :

- ✚ **Besoins techniques** : la maison intelligente doit relever tous les défis associés à son aspect technique tels que la fiabilité, l'interopérabilité et la capacité de fonctionner en présence d'ambiguïtés.
- ✚ **Besoins humains** : une maison intelligente doit intégrer un ensemble de fonctionnalités et de services qui doivent être faciles à comprendre et à utiliser par son habitant.

Aussi, ces fonctionnalités doivent avoir des interfaces attractives respectant ainsi l'ergonomie qui doit être adaptée à un environnement domestique. De plus, la maison intelligente doit respecter le contexte social de ses habitants.

3.3. Domaines d'application

Les maisons intelligentes permettent d'améliorer les conditions de vie en augmentant le confort de l'habitant. Parmi ce domaine d'application on trouve :



Grâce aux technologies intégrées dans les maisons, il est possible aujourd'hui de piloter l'éclairage, la température ainsi que différents appareils électroniques.



Il est désormais possible de contrôler les différentes ressources énergétiques telles que les lumières et les chauffages et d'adapter leurs utilisations en fonction des besoins et des rythmes des habitants dans le but de réduire les dépenses d'énergie.



Grâce aux solutions domotiques dédiées à la sécurité, il est possible de protéger la maison contre les différents dangers et menaces tels que les intrusions, les incendies, les inondations, etc. L'habitant est ainsi capable de surveiller sa maison à distance et il est averti immédiatement en cas de problème.



Les innovations technologiques intégrées dans les maisons aujourd'hui permettent de redonner une véritable autonomie aux personnes âgées et handicapées, de suivre leur santé et d'améliorer leur qualité de vie. Actuellement, plusieurs projets de recherche se sont consacrés à étudier les besoins des personnes âgées fragilisées qui souffrent d'une perte d'autonomie. De nombreuses études se sont penchées sur les technologies dédiées à la santé des personnes âgées.

3.4. Caractéristiques d'un environnement ubiquitaire

Parmi les propriétés permettant de qualifier un environnement d'ubiquitaire en trouve :

❖ Ouverture

Les environnements ubiquitaires sont des environnements ouverts permettant la cohabitation entre de multiples acteurs de marché qui deviennent de plus en plus actifs suivant la forte croissance technologique. En effet, ces environnements peuvent intégrer non seulement des dispositifs délivrés par des vendeurs et des fabricants différents mais

aussi des services issus de fournisseurs multiples. Dans un environnement ouvert, ces dispositifs et services doivent être conçus pour être interopérables ; c'est-à-dire que leurs spécifications doivent être intégralement connues et accessibles dans le but de fonctionner et de communiquer avec d'autres dispositifs ou applications.

❖ **Distribution**

Les environnements ubiquitaires intègrent un nombre de plus en plus important de dispositifs communicants : certains sont les dispositifs que nous utilisons tous les jours tels que les téléphones portables, les ordinateurs ou les tablettes ; d'autres sont des dispositifs qui permettent d'interagir avec l'environnement comme des capteurs qui permettent de capter des données de l'environnement et de connaître son état ou des actionneurs qui permettent d'agir sur l'environnement et de changer son état. En général, les applications ubiquitaires sont amenées à inter-opérer avec ces dispositifs, utiliser leurs fonctionnalités explicitement exposées et les coordonner pour rendre des services à l'utilisateur. Ces services sont ainsi le résultat d'une coopération entre les dispositifs dispersés dans son environnement, il s'agit donc d'un environnement distribué.

❖ **Hétérogénéité**

Avec l'essor de l'Internet et la forte croissance technologique, nos environnements deviennent de plus en plus riches en dispositifs fortement hétérogènes. Cette hétérogénéité résulte essentiellement d'une concurrence sévère entre les différents fabricants et vendeurs de dispositifs. La diversité des dispositifs dans les environnements ubiquitaires n'est pas limitée seulement au motif de leurs utilisations et leurs performances mais elle inclut aussi les technologies et les protocoles de communication, les modèles de données et les langages de programmation.

Le problème majeur qui résulte de cette hétérogénéité est la complexité de développement des applications ubiquitaires. En effet, ces applications sont amenées à inter-opérer avec un ensemble de ressources fortement hétérogènes. Les développeurs doivent ainsi intégrer une grande variété de types de ressources et de protocoles de communication et traiter des formats différents des données collectées de ces ressources, dans leurs applications.

❖ **Dynamisme**

Le dynamisme peut, par exemple, ajouter de nouveaux dispositifs et en retirer d'autres. En

outre, l'aspect dynamique des environnements ubiquitaires peut être aussi accordé à la mobilité des dispositifs. En effet, les dispositifs mobiles tels que les ordinateurs portables, les tablettes ou les téléphones mobiles peuvent être déplacés par l'utilisateur. Ce dernier peut ainsi transporter ces appareils et changer leur localisation à tout moment. Par conséquent, cette localisation dépend fortement des envies et du rythme de vie de l'utilisateur. Cet aspect de mobilité des dispositifs soulève le problème de la connectivité au réseau. Ainsi, ces appareils mobiles peuvent apparaître et disparaître du réseau de manière imprévisible et en fonction de la localisation de leur utilisateur. Par ailleurs, la variation de l'environnement peut être marquée par les propriétés des dispositifs. Par exemple, le niveau de batterie d'un capteur de mouvement renseignant sur sa durée de vie est indispensable pour son bon fonctionnement. En-dessous d'un seuil, des dysfonctionnements sont possibles et le dispositif peut ne plus être accessible. [Balddauf07] [Roy92] [Bettini10] [Chen00]

2.12. CONCLUSION

Les ontologies apparaissent désormais comme une clé pour la manipulation automatique de l'information au niveau sémantique. Au fur et à mesure des recherches, des idées se dégagent autour du contenu des ontologies, des méthodes à utiliser pour les construire et des modèles et langages servant à leur représentation.

Au long de ce chapitre, nous avons essayé d'éclaircir la notion d'ontologie et de maison intelligent en présentant certaines définitions. Nous avons découvert les méthodologies les plus représentatives de leur construction et quelques domaines de leur utilisation. Nous avons montré après les principaux formalismes de représentation de connaissances à savoir les frames, les graphes conceptuels et les logiques de descriptions. Nous avons aussi présenté les outils nécessaires à leur développement à savoir les langages de représentation et finalement nous présentons les maisons intelligents leur domaines d'application et leur caractéristique .

Dans le chapitre suivant, nous allons présenter une synthèse sur les différentes solutions de gestion de conflits dans les maisons intelligents base sur les ontologies.

**Les travaux connexes sur La gestion des
Conflits sémantiques dans une maison intelligente
Synthèse & Discussion.**

3.1. Introduction

L'objectif de l'informatique ubiquitaire est de fournir aux utilisateurs des services intelligents basés sur les informations obtenues à partir de ressources informatiques distribuées mais invisibles. Ces services ne nécessitent pas d'interface ou de procédures penchées lourdes pour que les utilisateurs les utilisent. En particulier, les applications contextuelles offrent des services appropriés aux utilisateurs en utilisant les informations contextuelles de l'environnement, y compris les utilisateurs. Ces informations sont obtenues à partir de divers capteurs ou ressources informatiques répartis dans notre vie quotidienne. Toutefois, des conflits se produisent dans les applications contextuelles lorsque plusieurs utilisateurs partagent les applications ou que ces applications partagent les ressources limitées de l'environnement.

L'étude présentée dans ce chapitre s'articule autour de la gestion la gestion des conflits dans une maison intelligente et une synthèse sur les approches proposées pour les gérer.

3.2. Gestion de conflits

Le problème des conflits est un sujet de plus en plus important de nos jours. La première notion de conflit, appelée en anglais *feature interaction*, a été introduite dans le domaine des télécommunications où un service est défini comme une collection de fonctionnalités. Le problème de conflit se manifeste lorsqu'une fonctionnalité, qui fonctionne correctement en isolation, se comporte d'une manière incorrecte en présence d'autres fonctionnalités.

D'une manière générale, un conflit se produit lorsque le comportement d'une fonctionnalité est modifié, du fait de sa coexistence avec d'autres fonctionnalités. La gestion des conflits a été au cœur de plusieurs travaux de recherche où de multiples définitions des conflits ont été proposées.

3.2.1. Définitions :

Dans ce qui suit, nous citons les définitions qui font aujourd'hui figure de référence dans le domaine des télécommunications :

| | |
|--|---|
| Définition1 [Kimble 95] | <i>“ A feature interaction occurs when the behavior of one feature is affected by the behavior of another feature or another instance of the same feature “</i> |
| Définition2 [Muffy 00] | <i>“Feature interaction is the term used to describe interference between services or features “</i> |

L'exemple de conflit le plus connu dans le domaine des télécommunications est celui qui peut se produire entre les fonctions de transfert d'appel et l'appel en attente. Un conflit se produit entre ces deux fonctions lorsqu'elles sont activées sur la même ligne téléphonique et un deuxième appel (ressource partagée) arrive alors que le premier appel n'est pas encore terminé. Dans ce cas, le système doit impérativement prendre une décision pour résoudre ce conflit : soit que le deuxième appel sera renvoyé vers un numéro secondaire (transfert d'appel) ou que le client sera averti qu'un autre appel est arrivé (appel en attente).

La décision optimale qui doit être prise par ce système téléphonique dépend principalement des besoins du client.

D'une manière générale, le problème des conflits peut être identifié dans tous les domaines où certaines ressources peuvent être partagées par plusieurs consommateurs (entités logicielles) qui doivent cohabiter. Ceci est résumé dans les phrases suivantes :

| | |
|---|---|
| Définition1 [Plath 98] | <i>“ These interactions can usually be traced down to the fact that _two 'features' manipulate the same entities in the base system, and in doing so violate some underlying assumptions about these entities that the other 'features' rely on “</i> |
| Définition2 [Muffy 00] | <i>“The subject has relevance to any domain where separate software entities control a shared resource “</i> |

Aujourd'hui, le problème de conflit est devenu l'un des enjeux majeurs dans les environnements informatiques ubiquitaires où le nombre d'applications et leur complexité sont en plein essor. En effet, les environnements ubiquitaires sont des environnements ouverts par définition. Ils doivent, par conséquent, assurer la cohabitation des applications fournies par des acteurs différents. Ces applications partagent les dispositifs, tant capteurs que actionneurs, présents dans leur environnement.

Cependant, elles peuvent agir de manière contradictoire sur les actionneurs partagés et elles risquent d'être en conflit. De par le nombre important des ressources dispersées dans les environnements ubiquitaires, leur aspect dynamique et leur nature hétérogène, la gestion des conflits est devenue un défi très complexe et très difficile à relever dans le domaine de l'informatique ubiquitaire.

Dans le cadre de travaux de recherche, plusieurs définitions de conflit ont été proposées. Ces définitions sont particulièrement dédiées à définir la notion de conflit dans les environnements ubiquitaires. Dans ce qui suit, nous analysons les principales définitions dans la littérature :

| | |
|---|---|
| <p>Définition1 [Verena 07]</p> | <p><i>“ A conflict occurs for an application or a user, if a context change _ caused by an application _ leads to a state of the environment which is considered inadmissible by the application or user. “</i></p> |
| <p>Définition2 [Henner 11]</p> | <p><i>“A resource usage raises a potential conflict when two or more controllers may access it. These controllers may be defined within an application or across applications “</i></p> |

D'après ces définitions, nous pouvons identifier, dans les environnements ubiquitaires multi-applications, deux types de conflits. Le premier type de conflit se manifeste au niveau des actionneurs dispersés dans l'environnement. Il se produit lorsque deux applications agissent sur le même actionneur de manière différente, par exemple allumer et éteindre la même lampe. Le deuxième type de conflit se présente au niveau de l'environnement. Il se produit lorsque deux applications agissent sur deux actionneurs différents mais leurs effets sur l'environnement sont contradictoires ou s'interfèrent. Par exemple, un conflit d'environnement peut avoir lieu lorsque deux applications agissent sur la température de la même pièce de manière contradictoire (augmenter et réduire la température).

A titre d'illustration, nous considérons une plateforme domotique qui héberge deux applications de sécurité : une application de détection d'incendie et une application de détection d'intrusion. Dans le cas d'un incendie, la porte d'entrée de la maison doit être déverrouillée par l'application de détection d'incendie alors que la nuit, la porte doit être verrouillée par l'application de détection d'intrusion. La nuit, dans le cas d'un incendie, ces deux applications agissent sur la porte de façon contradictoire. Un conflit sera ainsi détecté entre ces deux applications au niveau de la porte d'entrée.

Considérons un deuxième exemple de conflit où la plateforme domotique intègre deux applications dédiées au divertissement. La première application est Movies Application qui permet à l'utilisateur de regarder un film lancé sur le téléviseur installé par exemple dans le salon. La deuxième application est Music Application qui s'exécute en utilisant un hautparleur installé dans la même pièce. Dans cet exemple, les deux applications n'utilisent pas les mêmes dispositifs.

Néanmoins, elles affectent le niveau sonore du salon et les sons issus de ces applications s'interfèrent. Il s'agit donc de conflit de l'environnement.

En conclusion, le problème de conflits est devenu aujourd'hui l'un des problèmes majeurs dans les environnements informatiques ubiquitaires particulièrement dans la maison intelligente qui doit assurer la cohabitation d'un ensemble d'applications appartenant à un champ croissant de domaines. Certes, plusieurs travaux de recherche se sont penchés sur ce problème et ils ont proposé des solutions pour prévenir, détecter et résoudre ces conflits.

Une étude des différentes approches proposées pour la gestion des conflits dans le domaine de l'informatique ubiquitaire sera menée dans les titres suivantes.

3.3. Les classes des approches existantes

Le problème des conflits a été identifié pour la première fois dans le domaine des télécommunications où plusieurs chercheurs se sont penchés sur ce problème proposant ainsi plusieurs approches pour l'analyse et l'identification des conflits. La plupart de ces approches ont été divisées par Cameron et Velthuijsen [Jane 93], puis affinées par Bouma et Velthuijsen [Bouma94] en trois classes : **la prévention, la détection et la résolution**. La prévention consiste à chercher les moyens pour prévenir les conflits. La détection suppose que les conflits seront présents et permet de déterminer les méthodes pour les identifier et les localiser. La résolution suppose que les conflits seront présents et détectés et se penche sur les mécanismes pour minimiser leurs effets négatifs potentiels.

- **La prévention** : la prévention est une étape importante pour la gestion des conflits qui se manifeste par la recherche de moyens et de techniques pour les prévenir. En général, les approches de prévention des conflits visent à développer des plateformes ou des environnements de création logicielle qui mènent à l'implantation des services moins exposés aux conflits.

- **La détection** : la détection est essentielle pour la gestion des conflits qui permet d'identifier et de localiser les conflits. En effet, étant donné qu'il est peu probable d'éviter tous les conflits, il s'avère primordial de fournir des techniques qui permettent de les détecter. Les approches de détection de conflits peuvent être appliquées offline (pendant la spécification, la conception, l'implémentation) et/ou online (pendant la phase de tests ou après le déploiement). La détection des conflits offline permet d'identifier et localiser les conflits potentiels qui peuvent se produire entre les services. La complexité de détection dépend du nombre de services impliqués et des situations dans lesquels ils peuvent entrer en conflit. En revanche, la détection online des conflits permet de considérer seulement les services invoqués dans des situations bien particulières.
- **La résolution** : Une fois que les conflits sont détectés, il faut les résoudre. En effet, la résolution des conflits se penche sur les mécanismes et les stratégies qui permettent de minimiser ou d'éliminer leurs effets indésirables potentiels. Tout comme la détection, la résolution des conflits peut être effectué pendant le processus de création des services (offline) ou lorsque les services sont opérationnels (online).

Cette classification des approches proposées pour la gestion de conflits en trois classes a été largement utilisée dans plusieurs domaines où le problème des conflits a été au cœur de plusieurs travaux de recherches. Ces approches peuvent intervenir dans les différentes étapes du cycle de vie logiciel. En effet, elles peuvent être appliquées offline (pendant la spécification, la conception, l'implémentation) ou online (pendant la phase de tests ou après le déploiement) et même elles peuvent être des approches hybrides (combinaison d'offline et online).

Dans [Alma12], l'auteur définit les trois classes de gestion des conflits d'une manière plus générale et indépendamment du domaine : Le processus d'identification des conflits est appelé détection. Le processus qui permet de minimiser ou d'éliminer les effets indésirables d'un conflit est appelé résolution. La prévention est présentée comme étant une autre approche permettant d'éliminer

les conflits et qui repose principalement sur les directives (guidelines) et les architectures pour les prévenir.

3.4. Les solutions existantes

3.4.1. Les solutions classiques

Une multitude de travaux ont été menés pour identifier et résoudre les conflits dans les espaces intelligents. Un groupe de recherche se concentre sur les applications indépendantes.

Par exemple, MusicFX est un arbitre pour sélectionner des stations de musique pour plusieurs résidents dans un centre de remise en forme et génère automatiquement une séquence de stations de musique préférées pour les clients en fusionnant les préférences des utilisateurs des stations de musique [McCathy98].

Un autre exemple MavHome, crée un cadre de gestion des ressources sensible au contexte qui ajuste automatiquement le niveau des services pour résoudre les conflits de ressources, tels que les lumières ou la climatisation, en trouvant une valeur d'équilibre dans chaque appareil en fonction des emplacements prédits des utilisateurs [Das06].

Cependant, ces approches ne traitent que d'une résolution automatique des conflits au sein d'une application partagée par plusieurs utilisateurs. Bien que cette approche à initiatives mixtes résout les conflits d'applications contextuelles en exploitant la participation de l'utilisateur, elle suppose que les applications sont indépendantes les unes des autres [Shin08].

Par conséquent, ces approches sont insuffisantes pour traiter les conflits d'applications interdépendantes dans un espace intelligent.

Une autre approche a traité des conflits multi-utilisateurs en utilisant plusieurs appareils ou applications. Le système comportemental réactif (ReBA) est une application contextuelle pour un environnement de bureau qui réagit aux appareils du bureau et résout les conflits en les hiérarchisant [Kulkarni02]. Le mécanisme de mise en file d'attente centralisé détecte les conflits comme une incohérence des ressources partagées et les résout en attribuant priorité basée sur le profil de service [Haya06]. Ainsi, la priorité la plus élevée dans une file d'attente est sélectionnée pour utiliser les ressources partagées en cas de conflit. Cependant, ces approches sélectionnent toujours automatiquement l'une des applications en fonction de la priorité, ce qui signifie que les autres utilisateurs avec une priorité inférieure sont incapables d'utiliser les applications ou de fournir des informations sur la prise de décision.

Même si ce système résout automatiquement les conflits en divisant les ressources pour parvenir à une utilisation efficace [Park05], une telle division est rarement applicable et ne peut pas satisfaire tous les utilisateurs ayant des préférences différentes.

D'autres travaux prennent en compte la participation de l'utilisateur à la résolution des conflits sur les applications dépendantes. Cette approche de gestion des conflits centrée sur l'utilisateur gère les conflits entre plusieurs utilisateurs accédant aux services de médias en faisant des recommandations à l'utilisateur pour des solutions possibles [Shin07]. De même, les décisions dans les systèmes de recommandation de groupe reposent principalement sur les utilisateurs. Le Travel Decision Forum, par exemple, recommande et médiatise les problèmes de décision de groupe en avatars pour les utilisateurs distants. [Jameson04].

La stratégie de recommandation de programmes télévisés recommande des choix de programmes télévisés à un groupe d'utilisateurs en fusionnant leurs profils [Yu06]. GroupLens prédit l'intérêt des utilisateurs pour les articles de presse en exploitant les notes fournies par d'autres utilisateurs similaires. Ces approches encouragent les utilisateurs à participer à la résolution des conflits, mais cela nécessite une intervention de l'utilisateur chaque fois que des conflits surviennent dans les applications.

En outre, d'autres travaux ont tenté de résoudre les conflits en raisonnant sur l'ontologie.

Afin de protéger la confidentialité des utilisateurs, la méthode de flou de contexte contrôle le niveau de détails en raisonnant sur l'ontologie [Wishart05]. La résolution des conflits de localisation résout également les conflits de différents contextes grâce à un raisonnement ontologique [Niu08]. Bien que ce raisonnement basé sur l'ontologie soit utile pour obtenir une solution généralisée, la vraie solution dans les conflits multi-utilisateurs vient toujours de la négociation entre utilisateurs. La section suivante présente un ensemble de travaux basée sur les ontologies.

3.4.2. Les solutions basée sur les ontologies

Le problème de conflits est l'un des principaux défis dans les environnements smart home où le nombre d'applications et leur complexité sont en plein essor. En fait, ces environnements sont par définition ouverts. Il s'avère nécessaire d'assurer la coexistence d'applications fournies par différents acteurs. Néanmoins, le nombre élevé de ressources dispersées dans ces environnements, leur aspect dynamique et leur nature hétérogène font de la gestion des conflits un défi très complexe. Certes, de nombreuses recherches ont déjà porté sur cette problématique et plusieurs approches ont été proposées. Dans cette section, nous décrivons les travaux récents liés à nos recherches utilisant également l'ontologie.

Parmi les solutions proposées Tam [Van10] présente une réalisation du système contextuel de la maison intelligente basé sur le contexte d'ontologie. Dans cette approche, le temps est également un contexte important et naturel pour de nombreuses applications. De plus, la priorité est le paramètre utilisé pour classer les préférences des utilisateurs. Par conséquent, le temps et la priorité de la personne sont ajoutés en tant qu'informations contextuelles importantes et naturelles pour de nombreuses applications. Plus important encore, lorsque les contextes informatique, utilisateur et physique sont enregistrés sur une période de temps, le résultat est un historique du contexte, qui

pourrait également être utile pour certaines applications. L'ontologie est utilisée par les UTOPIA ¹.

Dans [Wang19], Tao présente une approche pour gérer les conflits, est un modèle de contexte formel basé sur l'ontologie utilisant OWL pour résoudre des problèmes tels que la représentation sémantique du contexte, le raisonnement contextuel et le partage des connaissances, la classification du contexte, la dépendance du contexte et la qualité du contexte. Le principal avantage de ce modèle est la capacité de raisonner sur différents contextes. Cette ontologie est utilisée par l'architecture SOCAM ²

Dans son article [Chen12], Liming propose une approche axée sur les connaissances pour la reconnaissance en temps réel et continue des activités basées sur des flux de données multicapteurs dans les maisons intelligentes. L'approche va au-delà des méthodes traditionnelles centrées sur les données pour la reconnaissance des activités de trois manières. Premièrement, il utilise largement les connaissances du domaine dans le cycle de vie de la reconnaissance des activités. Deuxièmement, il utilise des ontologies pour la modélisation et la représentation explicites du contexte et des activités. Troisièmement et enfin, il exploite le raisonnement sémantique et la classification pour l'inférence d'activité, permettant ainsi à la fois une reconnaissance d'activité à grain grossier et à grain fin. Cette approche est utilisée par une architecture de système générique basée sur les connaissances et décrivons le processus de reconnaissance sous-jacent basé sur l'ontologie.

Dans [Zhang05], les développeurs proposant un modèle générique en couches (la pile de contexte) pour guider la conception du système sensible au contexte. Ils étudient l'utilisation des technologies du Web sémantique dans la modélisation et le raisonnement contextuels. Un modèle de contexte basé sur l'ontologie Web (CONON) est défini pour faciliter la représentation explicite du contexte, le partage de contexte sémantique et le raisonnement logique. Ils montrant comment l'utilisation du raisonnement logique permet aux services contextuels de déduire un contexte implicite de haut niveau d'un contexte explicite de bas niveau. En particulier, ils intégrant l'ontologie

¹ Ubiquitous Computing Architecture

² Service-Oriented Context-Aware Middleware

temporelle dans leur modèle d'ontologie pour faciliter l'annotation et le raisonnement sur les informations de contexte temporel. Suivant le modèle général en couches, ils présentant une architecture de service contextuelle basée sur OSGi3 pour les maisons intelligentes.

Omer et al. [Sezer16] développent une ontologie de capteur de maison, qui est une ontologie spécialisée basée sur l'ontologie des réseaux de capteurs sémantiques (SSN). Ils développent également un environnement de simulation pour un boîtier de maison intelligente utilisant leur ontologie et présentant les premiers résultats de performance.

Dans [DIRKT17], les auteurs étudie les recherches effectuées sur les modèles d'informations sémantiques pour les systèmes d'automatisation des bâtiments. Il analyse les domaines d'information couverts pour fournir le contexte, le vocabulaire fourni pour décrire les dispositifs et fonctions d'automatisation des bâtiments, et la structure de ces modèles. L'intention est de découvrir les bonnes pratiques et d'essayer d'identifier les tendances, les points communs, les différences et les prochaines étapes possibles.

Nicolas en [Verstaeve18], aborde le problème de la gestion autonome de l'architecture IoT au moyen d'ontologies, il propose une architecture innovante qui s'appuie sur des ontologies pour permettre à une application contextuelle de s'auto-composer à la demande. Cette architecture est déployée sur deux campus intelligents.

Le groupe de [Shin05] proposant une gestion de conflit pour résoudre les conflits pour les applications contextuelles dans les environnements domestiques intelligents sous le titre de *Conflict_Manager*. Afin de résoudre les conflits entre les utilisateurs, le gestionnaire de conflits attribue la priorité à chaque utilisateur afin que l'utilisateur avec la priorité la plus élevée puisse être sélectionné en exploitant l'historique des conflits des utilisateurs. De plus, Conflict Manager détecte et résout les conflits entre les applications en utilisant les préférences des utilisateurs et les propriétés des services. Pour montrer l'utilité de la méthode de résolution des conflits proposée, ils appliquant

³Open Services Gateway initiative

la méthode de résolution des conflits proposée à ubiHome⁴, Les résultats expérimentaux ont prouvé que Conflict Manager permet aux applications contextuelles d'offrir des services personnalisés à plusieurs utilisateurs en résolvant les conflits de service entre les applications et entre les utilisateurs.

Dans [Szilagyi16], Ioan & Patric présentant un aperçu de certaines des technologies du Web sémantique utilisées dans les systèmes IoT, ainsi que certaines des ontologies bien acceptées utilisées pour développer des applications et des services pour l'IoT. Ils présentant enfin la Pile Web Sémantique pour l'Internet des Objets soulignant certaines de ses lacunes dans le développement d'une application ou d'un service IoT.

3.5. Critiques sur les méthodes de gestion de conflits

La plupart de solutions proposées se sont fixées comme un seul objectif, la détection de conflits dans les maisons intelligentes en négligent la résolution de ces conflits. C'est le point qui sera utilisé dans nos recherches

3.6. Synthèse et discussion

Le tableau 3.1 présente les solutions décrites précédemment en les positionnant par rapport aux catégories des approches utilisées pour la gestion des conflits. La détection et la résolution des conflits avant leur occurrence peuvent être considérées comme une stratégie de prévention. Pour cela, ce tableau permet de classer les approches proposées en deux classes : la détection (les techniques qui permettent d'identifier et de localiser les conflits) et la résolution (les techniques qui permettent d'éviter ou d'éliminer les effets indésirables potentiels des conflits). D'autres approches sont optimistes et se focalisent sur leur résolution à l'exécution. Par exemple, dans [Verena07], la résolution est principalement basée sur l'adaptation de l'application à l'origine du conflit. Cette approche repose sur la stratégie du premier arrivé, premier servi. Contrairement à [Verena07],

⁴Ubiquitous Home : un banc d'essai pour maison intelligente

[Henner11] utilise des priorités pour résoudre les conflits lors de l'exécution. Cependant, lorsqu'un conflit est détecté, l'application avec le niveau de priorité le plus bas ne peut pas satisfaire les exigences de l'utilisateur. En outre, toutes ces solutions présentent une limite majeure : elles nécessitent l'intervention humaine. En effet, certaines solutions exigent l'intervention du développeur d'applications pour gérer les conflits. Par exemple, dans [Verena07] les développeurs doivent faire un travail supplémentaire pour la gestion des conflits. Pour chaque application, ils doivent décrire les effets sur l'environnement et spécifier les conflits. D'autres nécessitent l'intervention de l'utilisateur pour résoudre les conflits. Par exemple, dans [Miki15], la résolution est la responsabilité de l'utilisateur.

| Approches | Ontologie | Contribution | Exemple d'utilisation | Résolution de Conflits |
|------------------------|--|--|---|------------------------|
| Résolution de conflits | Gestionnaire de conflits [Shin05] | Résoudre les conflits pour les applications contextuelles dans les environnements domestiques intelligents en utilisant les préférences des utilisateurs et les propriétés des services | Appliquant la méthode de résolution des conflits proposée à ubiHome | ✓ |
| Détection de conflits | Ontologie contextuelle pour la maison intelligente [Van10] | Le temps et la priorité de la personne sont ajoutés en tant qu'informations contextuelles pour des applications. Lorsque les contextes informatique, utilisateur et physique sont enregistrés sur une période de temps, le résultat est un historique du contexte, | Appliquée sur UTOPIA (UbiquiTous cOmPutIng Architecture) | x |

| | | | | |
|--|---|---|--|----------|
| | <p>Un modèle de contexte formel basé sur l'ontologie utilisant OWL [Wang19]</p> | <p>représentation du contexte sémantique, raisonnement contextuel et partage des connaissances, classification du contexte, dépendance au contexte et qualité du contexte.</p> <p>Le principal avantage de ce modèle est la capacité de raisonner sur divers contextes</p> | <p>Architecture SOCAM (Service-Oriented Context-Aware Middleware)</p> | <p>×</p> |
| | <p>Une approche axée sur les connaissances pour la reconnaissance des activités dans les maisons intelligentes [Chen12]</p> | <p>Modélisation et représentation explicites du contexte et des activités</p> | <p>Une architecture système générique</p> | <p>×</p> |
| | <p>Un modèle de contexte basé sur l'ontologie Web (CONON) [Zhang05]</p> | <p>Pour faciliter la représentation explicite du contexte, le partage du contexte sémantique et le raisonnement logique. Montrant comment l'utilisation du raisonnement logique permet aux services contextuels de déduire un contexte implicite de haut niveau d'un contexte explicite de bas niveau</p> | <p>Une architecture de service contextuelle basée sur OSGi pour les maisons intelligentes.</p> | |
| | <p>Une ontologie de capteur de maison intelligente [Sezer16]</p> | <p>Basé sur l'ontologie des réseaux de capteurs sémantiques (SSN)</p> | <p>Un environnement de simulation</p> | <p>×</p> |
| | <p>Modèles d'information sémantique pour les</p> | <p>Analyse les domaines d'information couverts pour fournir le contexte, le</p> | <p>/</p> | <p>×</p> |

| | | | |
|--|---|--|----------|
| <p>systèmes d'automatisation des bâtiments [DIRKT17]</p> | <p>vocabulaire pour décrire les dispositifs et fonctions d'automatisation des bâtiments, et la structure de ces modèles</p> | | |
| <p>Une architecture contextuelle basée sur les ontologies pour les applications de campus intelligentes [Verstaevel18]</p> | <p>Une architecture innovante qui s'appuie sur des ontologies pour permettre aux applications contextuelles de se composer à la demande</p> | <p>Cette architecture est déployée sur deux campus intelligents</p> | <p>x</p> |
| <p>Ontologies et Web sémantique pour l'Internet des objets [Szilagyi16]</p> | <p>La Pile Web Sémantique pour l'Internet des Objets soulignant certaines de ses lacunes dans le développement d'une application ou d'un service IoT.</p> | <p>Utilisé pour développer des applications et des services pour l'IoT</p> | <p>x</p> |

Tableau 3.1 : Synthèse des méthodes de gestion de conflits

3.7. Conclusion

Le problème des conflits a été au cœur de plusieurs travaux de recherche dans le domaine de l'informatique ubiquitaire. Au long de ce chapitre, nous avons présenté une synthèse sur les solutions existantes de gestion de conflits dans les environnements des maisons intelligentes basé sur les ontologies.

Le chapitre suivant présente notre contribution au problème posé par ce mémoire, à savoir la construction d'une ontologie pour le domaine d'environnements intelligents. Pour cela, nous allons

détailler notre contribution par des scénarios proposés, ensuite, nous allons suivre un processus de construction de l'ontologie qui est l'objectif de ce mémoire.



Partie 2 :

Chap4 : Conception d'une ontologie pour la résolution des conflits sémantique dans une maison intelligente.

Chap5 : Implémentation d'une ontologie pour la résolution des conflits sémantique dans une maison intelligente.

Conception d'une ontologie
Pour la résolution des conflits sémantique
Dans une maison intelligente

4.1. Introduction

Ce chapitre présente notre contribution au problème posé par ce mémoire qui est la construction d'une ontologie dédiée aux résolutions des conflits, dont le domaine est la maison intelligente. Pour ce faire, nous nous sommes basé sur la méthodologie METHONTOLOGY qui est le support de base pour la conceptualisation de l'ontologie à créer, à travers un ensemble de représentations intermédiaires semi-formelles. La logique de descriptions, est le formalisme adopté pour l'expression de l'ontologie semi-formelle.

4.2. Définition de conflit

Un conflit est un clash d'intérêts. Il survient pour une application ou un utilisateur, si un changement de contexte - causé par une application - conduit à un état de l'environnement qui est considéré comme inadmissible par l'application ou l'utilisateur. Aussi, l'utilisation des ressources soulève un conflit potentiel lorsque deux contrôleurs ou plus peuvent y accéder. la figure 4.1 montre des exemples les plus connus de conflits dans une maison intelligent.

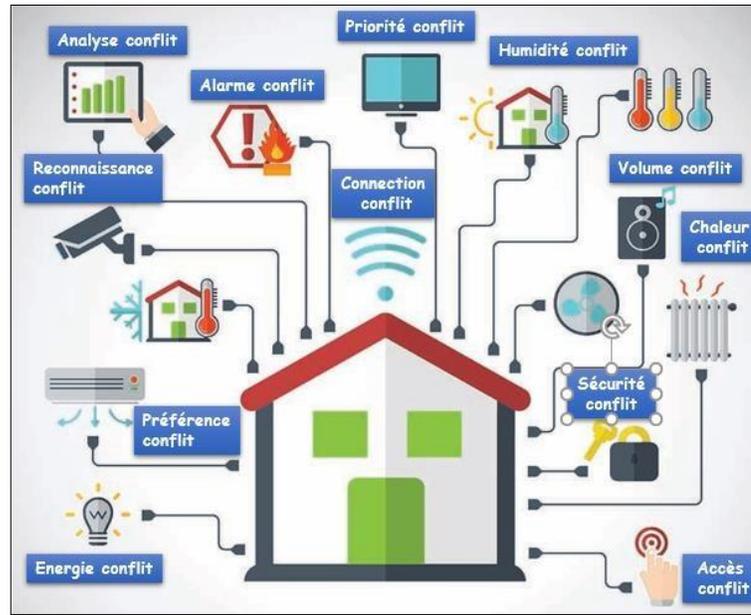


Figure 4.1 : Exemple de différents conflits dans la maison intelligente

4.3. Agrégation des conflits

Le contexte est conçu comme un ensemble de services disponibles dans l'environnement intelligent. Ces services peuvent être compatibles avec les appareils ou fournir des fonctions utiles. L'un des aspects les plus importants du contexte est que les services publiés peuvent être partagés par des applications installées sur la plateforme.

Ces applications utilisent les services fournis en fonction de leurs besoins afin d'atteindre des objectifs différents et parfois incompatibles.

On distingue deux types de conflits au niveau des services de contexte :

4.3.1. Le conflit direct

Qui survient lorsque plusieurs applications invoquent le même service de manière incompatible. Nous comprenons dans le terme incompatible des actions contradictoires (ouvrir vs. Fermer une porte), des configurations différentes (chauffer à 18°C et chauffer à 20 °C), etc

Trois sous catégories peuvent générer :

a. Conflits entre application :

Les conflits qui surviennent lorsque plusieurs applications (utilisateurs) accèdent au même service.

Priorité conflits

- Priorité d'accès aux applications entre les utilisateurs
- Priorité entre les services demandés par les applications

b. Conflits entre services :

Les conflits qui surviennent lorsque divers services partagent des ressources limitées pour répondre à des applications. On peut citer par exemple :

Energie conflits

Connection conflits

- Wi-Fi, Bluetooth ou sensor conflits

c. Conflits hybride entre applications et services :

Les conflits qui surviennent lorsqu'un service n'interagit pas avec l'application.

Services conflits

- Panne d'une machine (problème de batterie)
- Absences d'un service

Reconnaissance conflits

- Reconnaissance d'empreinte, de voix, de visage.

- Reconnaissance d'une action, ordre.

En outre, les conflits directs peuvent se présenter lorsque deux ou plusieurs applications invoquent directement le même service. Deux situations de conflits directs peuvent avoir lieu au niveau d'un contexte orienté services :

plusieurs applications consomment des services abstraits différents agissant sur le même service représentant un dispositif.

4.3.2. Le conflit indirect

Ce type de conflit se produit lorsque plusieurs applications invoquent des services différents en entraînant des actions incompatibles sur un service partagé.

Pour illustrer les conflit direct et indirect , nous examinons deux exemples simples.

Tout d'abord, considérons une plateforme ubiquitaire qui héberge deux applications de sécurité, appelées "Application d'intrusion" et " Application d'incendie". Leurs exigences sont les suivantes : l'application d'intrusion verrouille les portes d'entrée la nuit lorsque les habitants dorment ; l'application d'incendie s'assure que toutes les portes sont déverrouillées, lorsqu'un incendie est détecté pour permettre l'évacuation des habitants. Dans le cas d'un incendie la nuit, les deux applications agissent sur les portes (par le même service du contexte) de façon contradictoire. Ce conflit doit être détecté et, évidemment, la priorité doit être accordée à l'application d'incendie (c'est l'application la plus critique). Cet exemple Proposition de conflit direct où deux applications utilisent le même service (le service représentant les portes d'entrée d'une maison) est illustré par la figure 4.2.

Un autre exemple sur les conflits indirect concernant les lumières dans une maison. Les services sont disponibles pour contrôler les lampes d'une manière individuelle et

aussi pour contrôler la luminosité plus globalement dans une pièce, une maison entière. Voilà la description avec deux applications :

Une application de gestion de la consommation énergétique et une application de gestion d'éclairage. Le but de la première application est d'optimiser la consommation d'énergie de la maison. Son implémentation fait appel à des services contrôlant des zones entières telles que le service luminosité fourni par la plateforme. L'objectif de la deuxième application est d'allumer les lampes installées dans une pièce lorsqu'une présence est détectée. Plus précisément, cette application utilise des services qui représentent les lampes dans la plateforme. Il apparaît clairement que ces deux applications ont des objectifs contradictoires :

L'application de gestion de la consommation énergétique veut éteindre les lumières tandis que l'application de gestion d'éclairage cherche à les allumer. Bien qu'elles n'utilisent pas les mêmes services contextuels, le conflit doit être détecté et résolu. Cet exemple de conflit indirect où deux applications utilisent les mêmes ressources mais pas les mêmes services est illustré par la figure 4.3.

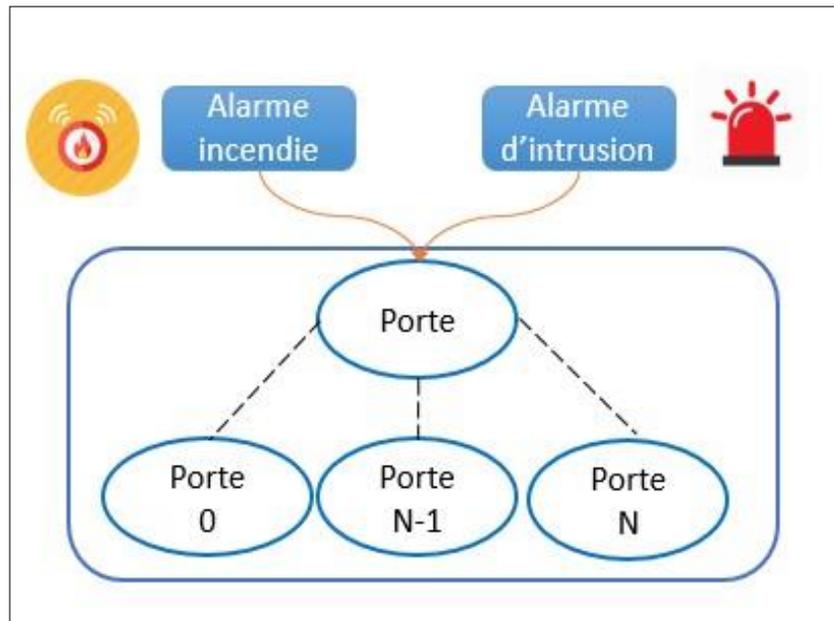


Figure 4.2 : Exemple de conflit direct

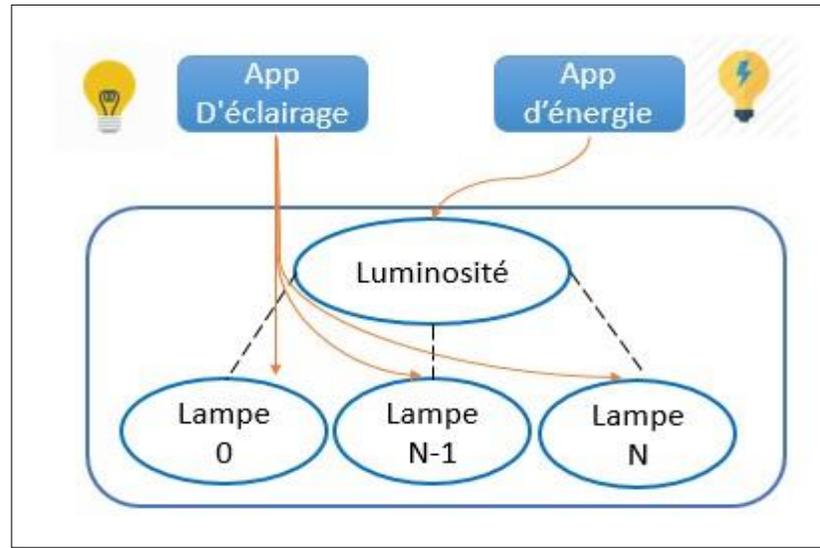


Figure 4.3 : Exemple de conflit indirect

Pour conclure, d'une manière générale, les conflits directs peuvent se produire lorsque deux ou plusieurs applications invoquent directement le même service du contexte. Ce service peut être un service abstrait ou tout simplement un service qui représente un dispositif dans la plateforme.

Le tableau 4.1 et le tableau 4.2 représentent respectivement le conflit avec la cause, et le conflit avec la solution où toutes les solutions que nous avons proposées s'atourne autour de la solution de priorité, de préférence ou bien d'une solution médiatisée :

❖ Conflits et causes

| Catégorie | Sous-catégorie | N° Conflit | Le Conflit | La cause de conflit |
|----------------|----------------|------------|--|--|
| Conflit | Entre services | 1 | Services conflits : Contradiction entre un service qui partage des ressources limitées. | Par exemple, les services de télévision et de cinéma peuvent être déclenchés |

| | | | | |
|---|--|---|--|--|
| | | | | en même temps lorsqu'un utilisateur entre chez lui |
| | Entre applications | 2 | Contradiction entre le service demandé et les requêtes des application (utilisateurs) opposées | <p>Ce conflit peut survenir dans le cas où deux (ou plusieurs) utilisateurs demandent un Service en même temps.</p> <p>Par exemple : l'utilisateur user_1 demande d'augmenter le volume du Tv, en même temps user_2 demandes de réduire le volume de Tv.</p> |
| | Hybride entre utilisateur et application | 3 | a. Confits de service : Contradiction entre les préférences demandées et la possibilité de reprendre. | L'utilisateur peut demander un service et le système ne l'offre pas, par exemple : l'utilisateur avec son application demande au système d'afficher la température de la chambre alors que la batterie de thermomètre est en panne. |
| 4 | | b. Contradiction entre les Requête des Services Préférés ET les droits d'accès de l'utilisateur | Ce conflit peut survenir dans le cas où l'utilisateur demande un Service qui ne convient pas avec ses droits d'accès. | |
| 5 | | c. Contradiction de reconnaissance entre le service et application | Les conflits qui surviennent lorsqu'un service n'interagit pas avec l'application. par exemple dans le cas où le service d'ouverture de la porte qui fonctionne avec le voix d'utilisateur est en échec à cause de la grippe de personne . | |

| | | | | |
|-------------------------|--|---|---|--|
| Conflit indirect | | 6 | Contradiction entre un service qui agit sur un service correspondant à un dispositif consommé par une autre application | Comme le cas où le service de la tv demande les baffles supplémentaires, alors que se dernière son utilise par le service de radio |
| | | | | |

Tableau 4.1 : Conflits & Causes

❖ Conflits et Solutions

| N de Conflit | Solution de conflits |
|--------------|--|
| 1 | Dans ce cas, le système informe l'utilisateur et lui demande au nouveau d'effectuer l'opération pour exécuter le premier choix, si l'utilisateur ne reprend pas dans un temps précis, le système va consulter l'historique de service d'utilisateur. |
| 2 | Afin de résoudre les conflits entre les utilisateurs, le gestionnaire de conflits attribue la priorité à chaque utilisateur afin que l'utilisateur avec la priorité la plus élevée puisse être sélectionné en exploitant l'historique des conflits des utilisateurs. |
| 3 | Dans ce cas, le système soit : <ul style="list-style-type: none"> · Il affiche le dernier résultat stocké · Ou, il informe utilisateur de réessayez dans quelques instants |
| 4 | Le système revient vers l'utilisateur pour l'informer qu'il n'est pas autorisé à accéder à ces services et demande donc des suggestions pour ce problème. Si l'utilisateur ne fait aucune suggestion, le système sera arrêté. |
| 5 | Après trois échecs de tests le service demande le code de système. |

| | |
|---|--|
| 6 | Comme la solution 1, le système va consulter l'historique de service d'utilisateur pour résoudre le conflit, si l'historique est vierge le système s'arrête. |
|---|--|

Tableau 4.2 : Conflits & Solutions

4.4. Motivation des ontologies

Afin de gérer les conflits dans une maison intelligente, nous avons choisi les ontologies comme une solution pour décrire les services dans une maison intelligente. De nombreuses recherches ont été menées pour développer une ontologie afin de prendre en charge l'interopérabilité et le partage d'informations contextuelles.

Notre choix d'utilisation de l'ontologie est dû aux raisons suivantes [Bourougaa16] :

- ✚ L'ontologie est moyenne pour décrire les informations pédagogiques.
- ✚ L'ontologie permet le partage des données.
- ✚ Facilement utilisé par d'autres applications et étendre la description initiale lorsque de nouveaux besoins surviennent.
- ✚ À l'aide du moteur d'inférence, les langages d'ontologie peuvent créer des modèles expressifs, évolutifs, réutilisables, partageables et inférés.
- ✚ Owl Language fournit une méthode simple et efficace basée sur le modèle de description XML pour partager les données décrites et ajouter des axiomes pour décrire des relations spécifiques entre les informations.

4.5. Processus de construction de l'ontologie

Nous utilisons un processus de construction dans le développement de l'ontologie partant de connaissances brutes et arrivant à une ontologie d'application opérationnelle représentée par le langage OWL. Les grandes étapes de ce processus sont inspirées de la méthodologie de construction d'ontologies « METHONTOLOGY ». Ce processus est composé de cinq étapes :

1. Spécification des besoins.

2. Conceptualisation.
3. Formalisation.
4. Implémentation.
5. Test & évolution de l'ontologie.

4.5.1. Spécification :

Cette étape consiste à établir un document formel de spécification des besoins représenté dans le langage RDF. Ce dernier permet de décrire l'ontologie à construire à travers les cinq aspects suivants :

1. **Le domaine de connaissance** : déterminer aussi précisément que possible le domaine que va couvrir l'ontologie.
2. **L'objectif** : le but de l'ontologie à créer pour le domaine considéré.
3. **Les utilisateurs** : identifier au maximum les futurs utilisateurs de l'ontologie à créer.
4. **Les sources d'informations** : déterminer les sources d'informations d'où les connaissances seront obtenues, par exemple, les experts du domaine, les documents techniques, etc.
5. **La portée de l'ontologie** : déterminer à priori la liste des termes (les plus importants) pour le domaine à représenter.

4.5.2. Conceptualisation :

C'est l'étape la plus importante dans le processus de construction de l'ontologie. Elle est inspirée de la méthodologie METHONTOLOGY qui consiste à identifier et à structurer, à partir des sources d'informations, les connaissances du domaine. Elles

permettent d'aboutir à un ensemble de représentations intermédiaires semi-formelles indépendamment des langages de formalisations à utiliser pour représenter l'ontologie.

A la fin de cette phase, nous obtenons une ontologie conceptuelle.

Pour cela on distingue les principales tâches suivantes :

- Construction du glossaire de termes.
- Construction du diagramme de classification des concepts.
- Construction du diagramme de relations binaires.
- Construction du dictionnaire de concepts.
- Décrire les relations dans une table de relations binaires.
- Spécifier des contraintes sur les attributs dans une table d'attributs.
- Spécifier des axiomes sur les concepts dans une table d'axiomes logiques.
- Décrire les instances des concepts dans une table d'instances.

4.5.3. Formalisation :

Cette phase consiste à formaliser l'ontologie conceptuelle obtenue dans la phase précédente afin de faciliter sa représentation ultérieure dans un langage complètement formel et opérationnel. Notre choix est porté sur le formalisme de représentation est la logique de descriptions. Elle est constituée de deux parties :

une partie terminologique (TBOX) permettant de décrire les concepts et les rôles et d'une partie assertionnelle (ABOX) décrivant les instances.

4.5.4. Implémentation :

L'ontologie que nous avons obtenue dans la phase précédente est appelée une ontologie formelle. Le but de cette étape sera donc de coder l'ontologie formelle en OWL DL qui

dispose de fonctionnalités sémantiques plus riches que ses prédécesseurs RDFS ou DAML+OIL. A la fin de cette phase, nous obtenons une ontologie opérationnelle.

Afin de faciliter le processus de codification, nous utilisons PROTEGE OWL version 3.4.1 qui dispose d'une interface permettant l'édition, la visualisation et le contrôle d'ontologies, et contient des classes (concepts), des propriétés et valeurs des propriétés et contraintes, ainsi que des instances des classes et des propriétés.

4.5.5. Tests et évolution

Cette étape consiste à exploiter les services d'inférence fournis par la logique de description afin de supporter le processus de construction et d'améliorer la qualité de l'ontologie. Pour ce faire, nous proposons l'utilisation de l'outil RACER, un système de la logique de descriptions.

Ce dernier, permet de lire un document au format OWL (ontologie OWL) et de le représenter sous forme d'une base de connaissances LD et de fournir des services d'inférence pour les niveaux TBOX et ABOX.

4.6. Construction de l'ontologie

Dans cette section, nous construisons notre ontologie qui concerne la gestion des conflits sémantique dans la « Smart Home ». A cette fin, nous suivrons les étapes du processus de construction d'ontologie proposé dans la section précédente.

4.6.1. Spécification :

Le développement de l'ontologie est débuté par la phase de spécification qui consiste à établir un document de spécification des besoins. Au sein de ce document, nous dériverons l'ontologie à construire à travers les cinq aspects suivants :

- **Le domaine de connaissance** : l'ontologie, que nous venons de construire, s'inscrit dans le cadre de conflits sémantiques dans les maisons intelligentes. En effet, elle peut prendre ses concepts depuis conflits possibles dans la maison intelligente.

- **L'objectif** : l'objectif majeur de l'incorporation des ontologies au sein d'un système de gestion de conflits est de celer l'hétérogénéité concernant les applications (utilisateurs) et les services.
- **Les utilisateurs** : cet aspect présente l'ensemble des utilisateurs pouvant exploiter l'ontologie. Dans notre cas, les utilisateurs de l'ontologie représentent les habitats de la maison intelligente.
- **Les sources d'informations** : les sources d'informations sur lesquelles nous nous sommes basés pour arriver à la construction de l'ontologie sont des documents techniques de la maison intelligente, des ontologies pour les maisons intelligentes, et des ontologies pour la gestion des conflits.
- **La portée de l'ontologie** : Cet aspect consiste à déterminer à priori la liste des termes de l'ontologie (les plus importants), parmi ces termes, nous pouvons citer : smart home, smart tv, smart radio, contexte, porte ... etc.

Nous résumons cette phase dans un document RDF présenté dans la Figure suivante. Il peut inclure aussi d'autres aspects tels que : la date de création de l'ontologie, ses créateurs, son niveau de formalité ...etc.

```
1 <rdf:RDF>
2 <rdf:Description about=" URI of ontology" >
3 <Domaine> Gestion des conflits sementiques dans la maison intelligente </ Domaine>
4 <Date> 01/09/2020 </ Date>
5 <Développé-par>
6 <rdf:Sequence>
7 <rdf:_1 I. DEBALLIA, Université Larbi Tebessi de Tébessa >
8 <rdf:_2 S. BOUROGAA, Université Larbi Tebessi de Tébessa>
9 </rdf:Sequence>
10 </ Développé-par >
11 <Objectif> une ontologie pour gerer les conflits sementique possible dans un envirement
12 intellgent qui est la maison intelligente .L'objectif de la construction de cette ontologie
13 est de fournir une sémantique claire, interprétable, et partagée entre les applications et
14 les services de la maison intelligente.
15 </ Objectif>
16 <Niveau de formalité> formel </ Niveau de formalité >
17 <Termes>
18 <rdf:Sequence>
19 <rdf:_1 Smart_Home> <rdf:_2 Tv> <rdf:_3 Temperature > ...
20 </rdf:Sequence>
21 </ Termes>
22 <Sources>
23 <rdf:Sequence>...
24 </rdf:Sequence>
25 </ Sources>
26 </rdf:description>
27 </rdf:RDF>
```

Figure 4.4 : Un document RDF de spécification de l'ontologie.

4.6.2. Conceptualisation :

Une fois que la majorité des connaissances est acquise, on doit les organiser et les structurer en utilisant des représentations intermédiaires semi-formelles qui sont faciles à comprendre et sont indépendantes de tout langage d'implémentation. Cette phase comporte plusieurs étapes qui sont :

- Construction de glossaire de termes.
- Construction de diagramme de classification de concepts.
- Construction de diagramme de relations binaires.
- Dictionnaire de concepts.
- Tableaux des relations binaires.
- Tableaux des attributs.
- Tableaux des axiomes logiques.
- Tableaux des instances.

1. Construction de glossaire de termes :

Ce glossaire contient la définition de tous les termes relatifs au domaine (concepts, instances, attributs, relations) qui seront représentés dans l'ontologie finale, par exemple, dans notre cas les termes ***Tv*** et ***Alarm*** sont des concepts, ***Habite a*** et ***Concerne un conflit*** représentent des relations, etc. Le Tableau suivant fournit une liste détaillée des différents termes utilisés dans l'ontologie.

| Terme | Synonyme | Acronyme | Sub_terme |
|------------------|----------|----------|--|
| Contexte | - | - | - |
| Enivrement | - | - | - |
| Location | - | - | - |
| Conflit direct | - | - | Conflits entre application Conflits entre Services |
| Conflit indirect | - | - | - |
| Connectivité | - | - | Connectivité (image, vidéo, Sound) |
| Dispositive | - | - | - |
| Télévision | - | Tv | - |
| Alarme | Réveille | - | - |
| Radio | - | - | - |
| Climatiseur | - | Clim | - |
| Port | - | - | - |
| Power | - | - | - |
| Permission | - | Perm | - |
| Touche | - | - | - |

| | | | |
|----------------------------|---|--------|--|
| Position | - | - | - |
| Gravité | - | - | - |
| Utilisateur | - | user | - |
| Activité | - | - | - |
| Reconnaissance | - | - | - |
| Détection | - | - | - |
| Mouvement | - | - | - |
| Maison intelligente | - | Maison | - |
| Appareille mobile | - | Phone | - |
| Tablette | - | Tab | - |
| Capteur | - | - | Capteur de température Capteur de lumière Capteur de pression Capteur d'humidité préférence Capteur de mouvement |
| Profil | - | - | - |
| Bruit | - | - | - |
| Environnement | - | - | - |
| Préférence | - | - | - |
| Commande | - | - | - |
| Allumer | - | - | - |
| Wifi | - | Réseau | - |
| Bluetooth | - | - | - |
| Camera | - | cam | - |
| cause_conflit | - | - | - |
| solution_confit | - | - | - |
| Demander_Baffe | - | - | - |

| | | | |
|-----------------------------|---|---|---|
| Habite_a | - | - | - |
| A_une_activité | - | - | - |
| Declanché_par | - | - | - |
| Causé_par | - | - | - |
| A_une_solution | - | - | - |
| Concernt_un_conflit | - | - | - |
| Reconnaissance | - | - | - |
| Augmenter | - | - | - |
| Réduire | - | - | - |
| Ouvrire_Port | - | - | - |
| Afficher_Température | - | - | - |
| Est_detecte_par | - | - | - |
| Utilisé_par | - | - | - |
| Avoir | - | - | - |
| Respect | - | - | - |
| Partie_de | - | - | - |
| Exist | - | - | - |

Tableau 4.3 : Glossaire de termes

2. Construction du diagramme de classification de concepts :

Dans cette étape, nous construisons le diagramme de classification de concepts. La hiérarchie de classification de concepts démontre l'organisation des concepts de l'ontologie en un ordre hiérarchique qui exprime les relations sous classe. Le tableau suivant présente les mains class et leur description.

| Main classe | Description |
|----------------------------|---|
| Maison intelligente | C'est le lieu des conflits en question. |
| Activité | L'activité peut être : détection, reconnaissance, fermeture, ouverture, contrôle et affichage de l'application sensible au contexte. |
| Contexte | Le contexte fait référence à toute information qui peut être utilisée pour caractériser l'état d'une entité, notamment l'éclairage, le niveau de bruit, la reconnaissance vocale, l'emplacement, la commande vocale, etc. |
| Dispositive | le dispositif fait référence à toute machine ou pièce connecté dans la maison intelligente |
| Conflit | Représente les conflits qui peuvent survenir lors d'une requête d'un utilisateur. |
| Utilisateur | L'habitat de la maison intelligente. |

Un concept universel « **Thing** », qui généralise tous les concepts racines des différentes hiérarchies de concepts est utilisé pour former une seule hiérarchie globale.

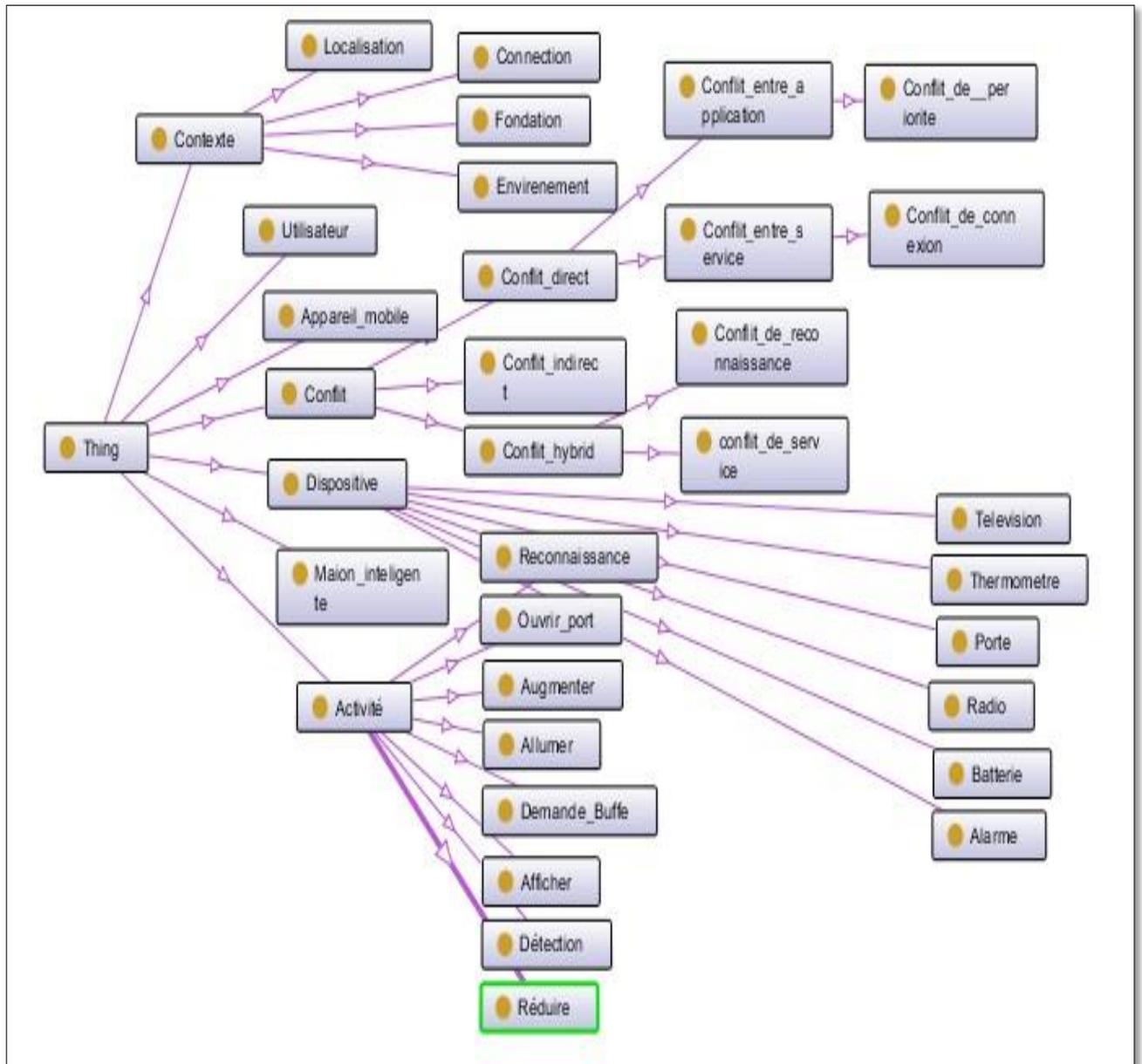


Figure 4.5 : Diagramme de classification de concepts

3. Construction de diagramme de relations binaires :

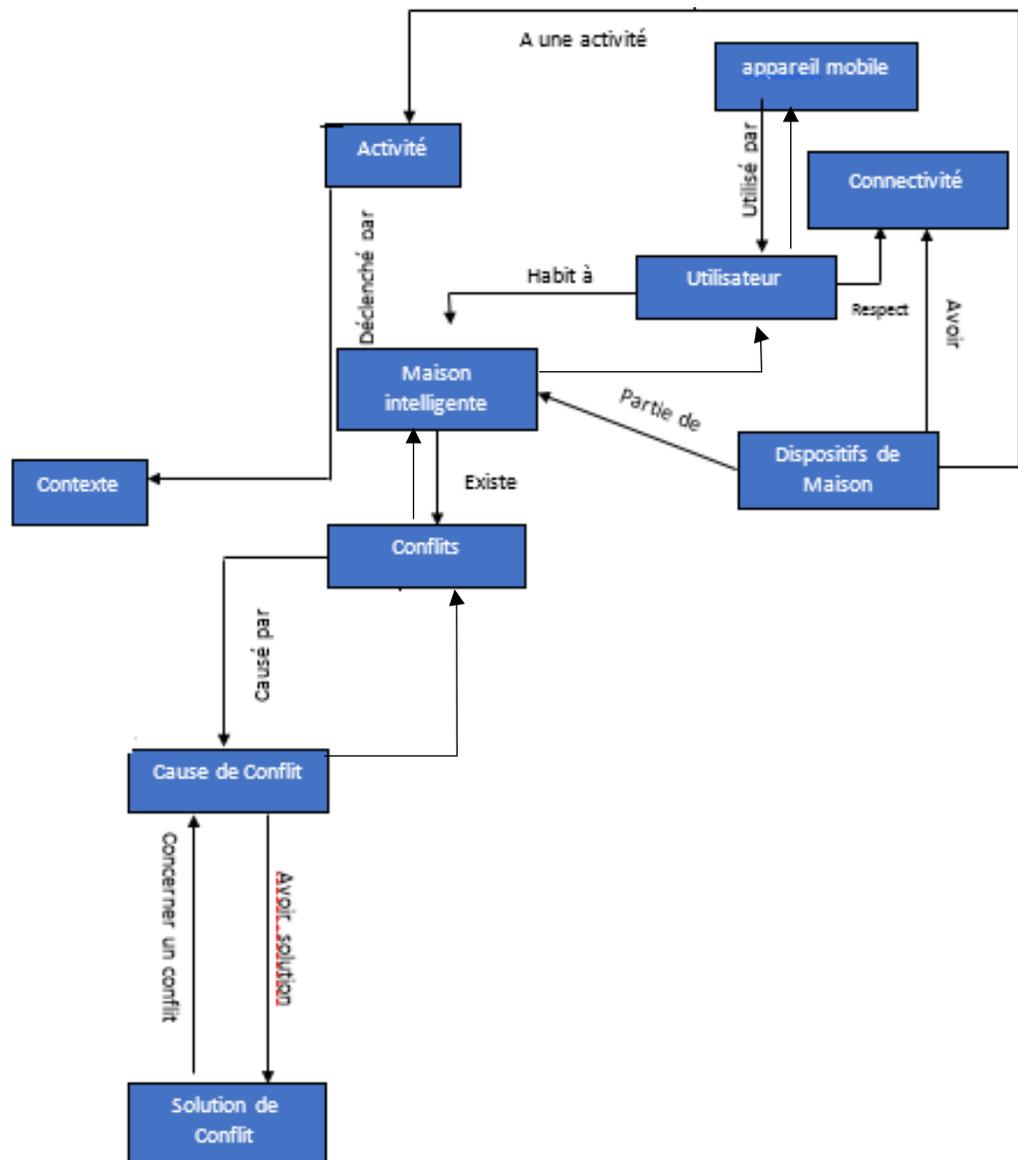


Figure 4.6 : Diagramme de relations binaires

4. Dictionnaire de concepts :

| Concept | Instance | Attributs | Relation |
|--------------------------|---|---|--------------------------------------|
| Appareille mobile | Mobile 1 Mobile2 | Modèle | Utilisé_par |
| Dispositive | - | Id_dispositive | Avoir Partie_de A_une_activité |
| Utilisateur | Utilisateur 1 Utilisateur 2 | Id_utilisateur Nom_utilisateur E_mail | Respect Habite_a |
| Contexte | - | Id_contexte | Declenché_par |
| Maison | - | Id_masion Adress_maison | Exist |
| Conflit | - | Id conflit | Concernt_un_conflit Causé_par |
| Capteur | Capteur de température Capteur de lumière Capteur de pression Capteur d'humidité Capteur de mouvement | Id_capteur Type_capteur | - |
| Activité | - | Id_activite | Declanché_par |

Tableau 4.5 : Dictionnaire de concepts

5. Tableaux des relations binaires :

| Relation | Concept source | Cardinalité max | Concept cible | Relation inverse |
|--------------------|-------------------|-----------------|---------------|------------------|
| Utilisé_par | Appareille mobile | N | Utilisateur | - |
| Avoir | Dispositive | 1 | Connectivité | - |
| Respect | Utilisateur | 1 | Connectivité | - |

| | | | | |
|----------------------------|---------------------|---|---------------------|---------------------|
| Partie_de | Dispositive | 1 | maison | - |
| Exist | Maison | N | Conflit | - |
| Concernt_un_conflit | Solution de conflit | 1 | Cause de conflit | A_une_solution |
| A_une_solution | Cause de conflit | 1 | Solution de conflit | Concernt_un_conflit |
| Causé_par | Conflit | N | Cause de conflit | - |
| Declanché_par | Activité | N | Contexte | - |
| A_une_activité | Dispositive | N | Activité | - |
| Habite_a | Utilisateur | 1 | Maison | - |

Tableau 4.6 : Tableau des relations binaires

6. Tableaux des attributs :

| Nom d'attribut instance | Concept | Type de valeur | Cardinalité |
|-------------------------|-----------------|----------------|-------------|
| Id_activite | Activité | Entier | (1,1) |
| Id_dispositive | Dispositive | Entier | (1,1) |
| Id_utilisateur | Utilisateur | Entier | (1,1) |
| Nom_utilisateur | Utilisateur | String | (1,1) |
| E_mail | Utilisateur | String | (1,1) |
| Id_conflit | Conflit | Entier | (1,1) |
| Id_masion | Maison | Entier | (1,1) |
| Adress_maison | Maison | String | (1,1) |
| Modèle | Appareil mobile | String | (1,1) |

Tableau 4.7 : Tableau des attributs

7. Tableaux des axiomes logiques :

| Description | Expression | Concept concerne |
|--|--|-------------------------|
| Deux utilisateurs avec leurs priorités demandant deux activités aux même temps, l'utilisateur avec la plus | $\text{Utilisateur} (?P1) \wedge$ $\text{Utilisateur} (?P2) \wedge$ $\text{A_une_Activité} (?A1, ?P1) \wedge$ $\text{A_une_Activité} (?A2, ?P2) \wedge$ | Utilisateur Activité |

| | | |
|---|--|---|
| grande priorité est servi le premier. | $A_la_P\acute{e}riorit\acute{e} (?Pr1, ?P1) \wedge$ $A_la_P\acute{e}riorit\acute{e} (?Pr2, ?P2) \wedge$ $Swrl :higher_Than(?P2, ?P1) \rightarrow$ $A_une_Activit\acute{e}(?A1, ?P1)$ | |
| Deux utilisateur (père et leur fils) demandant le même service , la demande de père est servi le premiere . | $Utilisateur (?U1) \wedge$ $Utilisateur(?U2) \wedge$ $A_une_Activit\acute{e}(AllumerTV , ?U1)$ $\wedge A_une_Activit\acute{e}(eteindreTV , ?U2) \rightarrow Activit\acute{e}(AllumerTV)$ | Utilisateur Activit\acute{e} Tv |
| Quand Un utilisateur est arriv\acute{e} \`a la maison, la lampe est allum\acute{e}e. | $Maison(M1) \wedge Utilisateur (?U) \wedge$ $Exist (?M1,U) \rightarrow$ $Allumer(?lamp)$ | Maison Utilisateur Lamp |
| Quand Un utilisateur a quitt\acute{e} la maison, la lampe est \acute{e}teinte. | $Maison(M1) \wedge Utilisateur (?U) \wedge$ $quit (?M1,U) \rightarrow Eteindre$ $(?lamp)$ | Maison Utilisateur Eteindre Lamp |

Tableau 4.8 : Tableau des axiomes logiques

8. Tableaux des instances :

| Nom d'instance | Nom de concept |
|--|-------------------|
| Utilisateur 1 (?) Utilisateur 2 (?) | Utilisateur |
| Alarme de temps(?) Alarme de porte(?) Alarme de (?) | Alarme |
| Porte 1 (?) Porte 2 (?) | Porte |
| Mobile 1 (?) Mobile2 (?) | Appareille mobile |
| Capteur de temp\acute{e}rature (?) Capteur de lumi\`ere (?) Capteur de pression (?) Capteur d'humidit\acute{e} (?) Capteur de mouvement (?) | Capteur |

Tableau 4.9 : Tableau des instances

4.7. CONCLUSION

Dans ce chapitre, nous avons présenté notre contribution, nous avons développé une ontologie OWL pour les conflits dans les maisons intelligentes. Nous avons utilisé un processus de construction d'une ontologie en s'inspirant des différentes phases proposées par la méthode METHONTOLOGY pour la conceptualisation de l'ontologie afin d'atteindre un ensemble de représentations intermédiaires qui facilite sa formalisation ultérieure et cela en adoptant l'approche de la logique de description.

**Implémentation d'une ontologie
pour la résolution des conflits sémantique
dans une maison intelligente**

5.1. Introduction

Ce chapitre présente notre implémentation au problème de gestion des conflits sémantiques dans les maisons intelligentes. OWL, le langage de définition d'ontologies, est choisi afin de codifier l'ontologie en utilisant l'éditeur d'ontologies Protégé OWL. Finalement, le système d'inférences RACER (*Renamed Abox and Concept Expression Reasoner*), est utilisé afin de tester la consistance de l'ontologie tout au long du processus de développement.

5.2. Implémentation :

Après la conception, nous allons implémenter notre ontologie. Notre choix porte sur OWL qui représente un langage de codification utilisé pour implémenter l'ontologie en OWL, et cela pour toutes les fonctionnalités sémantiques que permet OWL et qui sont plus riches que celles des langages RDFS & DAML+OIL

5.2.1. Présentation de l'éditeur PROTEGE OWL :

PROTEGE OWL est une interface modulaire, développée au Stanford Medical Informatics de l'Université de Stanford⁷, permettant l'édition, la visualisation, le contrôle, l'extraction à partir de sources textuelles, et la fusion semi-automatique d'ontologies [Noy00].

PROTEGE OWL autorise la définition de méta-classes, dont les instances sont des classes, ce qui permet de créer son propre modèle de connaissances avant de bâtir une ontologie. De nombreux plugins sont disponibles ou peuvent être ajoutés par l'utilisateur. La Figure 5.1 présente l'interface de PROTEGE OWL version 3.4.1

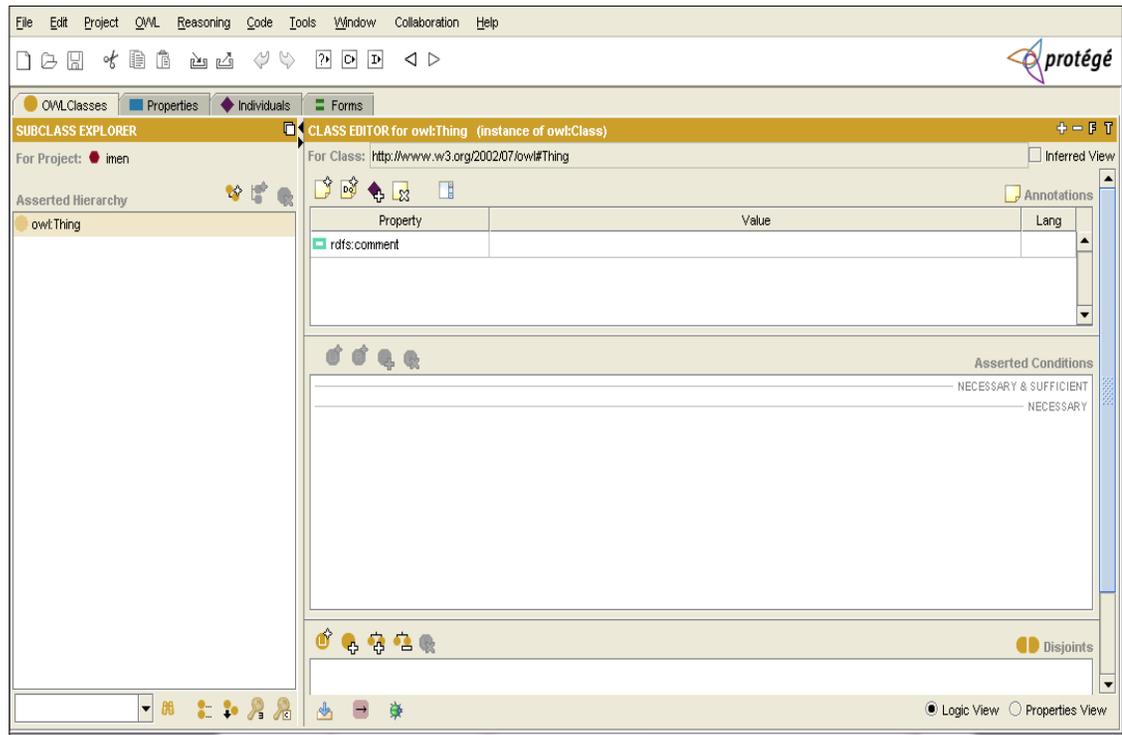


Figure 5.1 : Interface de Protégé OWL

L'interface, très bien conçue, et l'architecture logicielle permettant l'insertion de plugins pouvant apporter de nouvelles fonctionnalités (par exemple, la possibilité d'importer et d'exporter les ontologies construites dans divers langages opérationnels de représentation tels que OWL ou encore la spécification d'axiomes) ont participé au succès de PROTEGE OWL, qui regroupe une communauté d'utilisateurs très importantes et constitue une référence pour beaucoup d'autres outils [Troncy04].

5.2.2. Définition de la hiérarchie des classes :

Nous commencerons tout d'abord par la création des concepts spécifiés dans l'étape de conceptualisation. PROTEGE OWL nous offre également un moyen de construire la hiérarchie de concepts, la Figure 5.2 présente comment-on procède pour créer un concept.

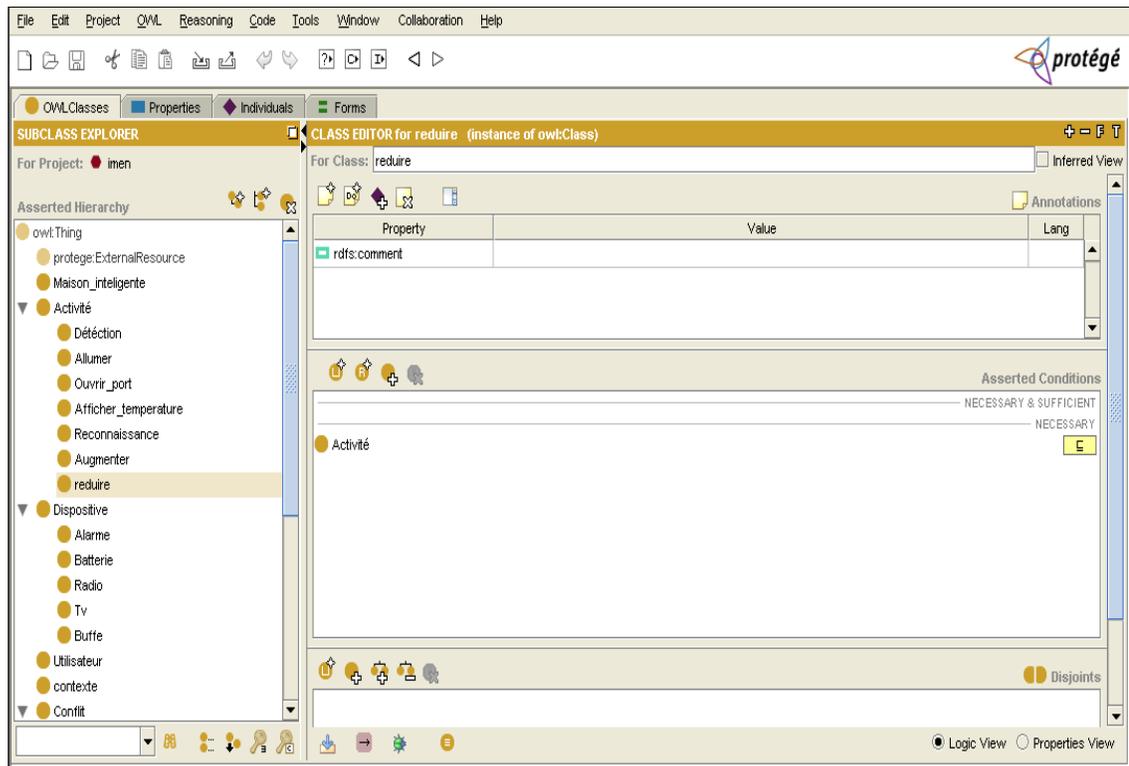


Figure 5.2 : Création des classes

5.2.3. Définition des propriétés :

Après avoir construit les classes, nous allons maintenant créer les propriétés pour chacun d'eux, les attributs vont être créés sous PROTEGE OWL par 'dataTypeProperty' et les relations par 'objectProperty'.

La Figure 5.3 montre les potentialités de PROTEGE OWL pour la création des propriétés.

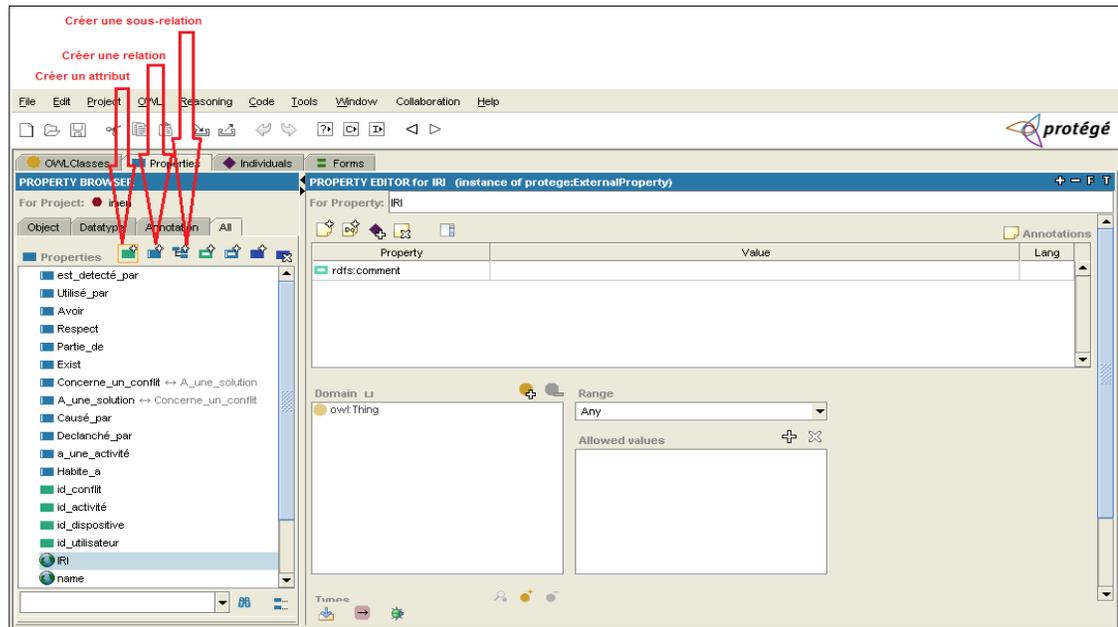


Figure 5.3: Création des propriétés pour une classe

La Figure 5.4 montre comment créer un attribut et spécifier son nom, domaine et type.

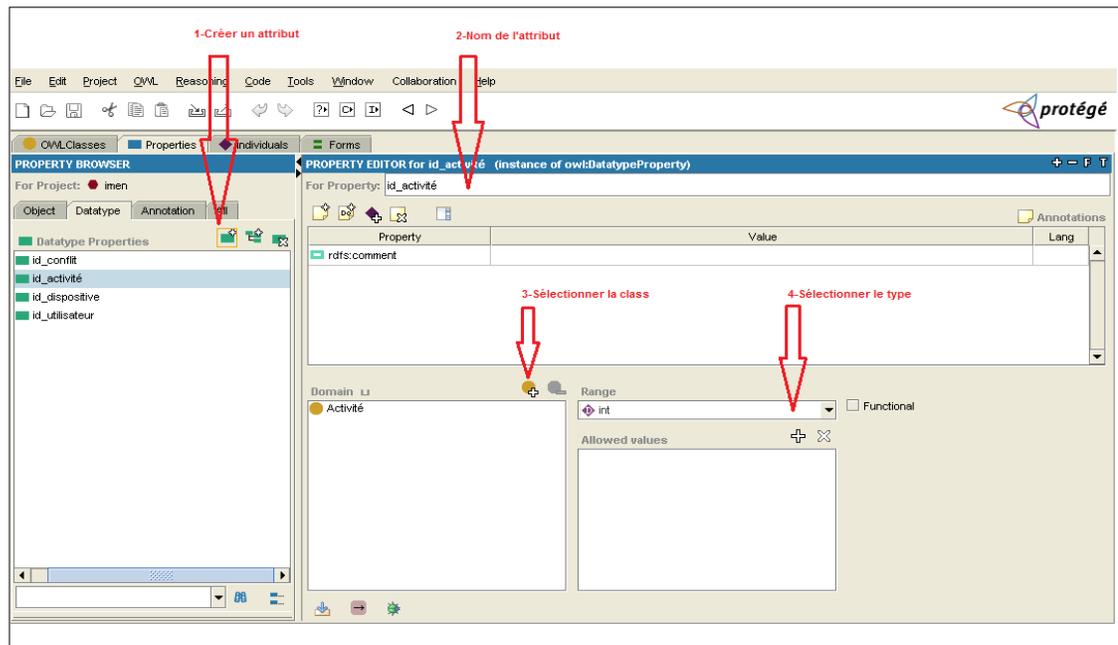


Figure 5.4 : Création d'un attribut

La Figure 5.5 montre comment créer une relation et spécifier son nom, type, domaine et codomaine.

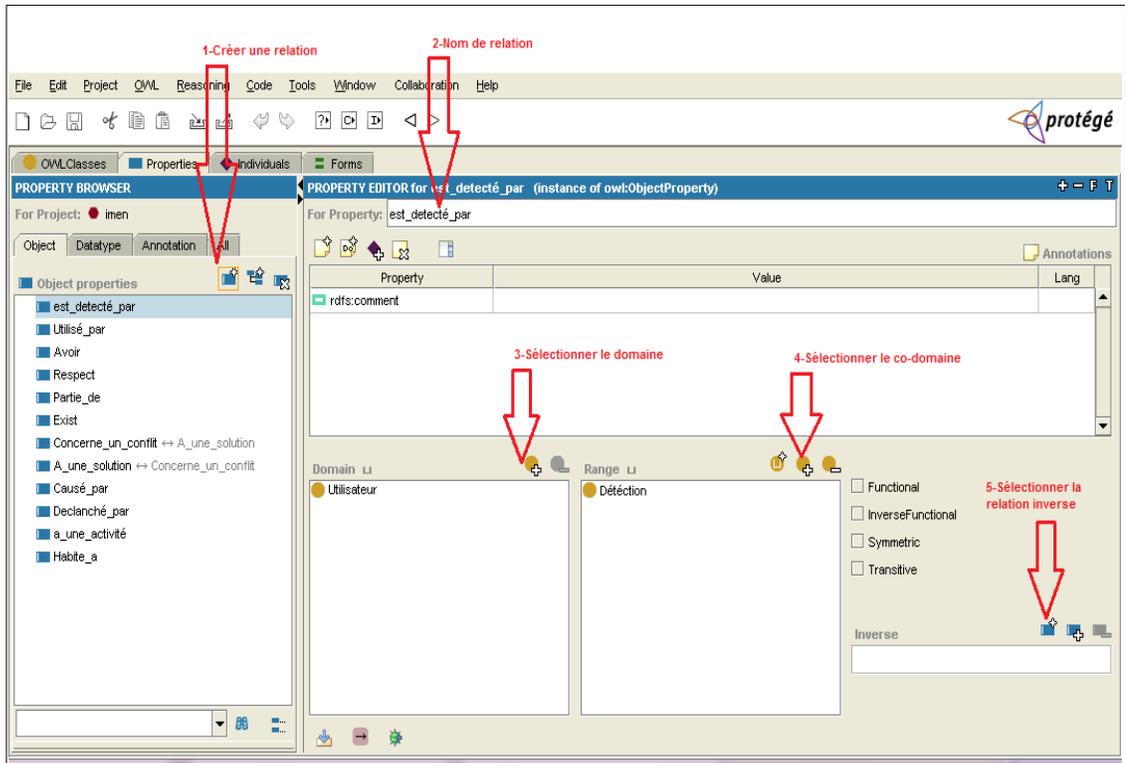


Figure 5.5 : Création d'une relation

5.2.4. Définitions des restrictions :

PROTEGE OWL nous offre un moyen pour créer des restrictions sur les classes et les propriétés. La Figure 5.6 montre comment créer une restriction sur une classe.

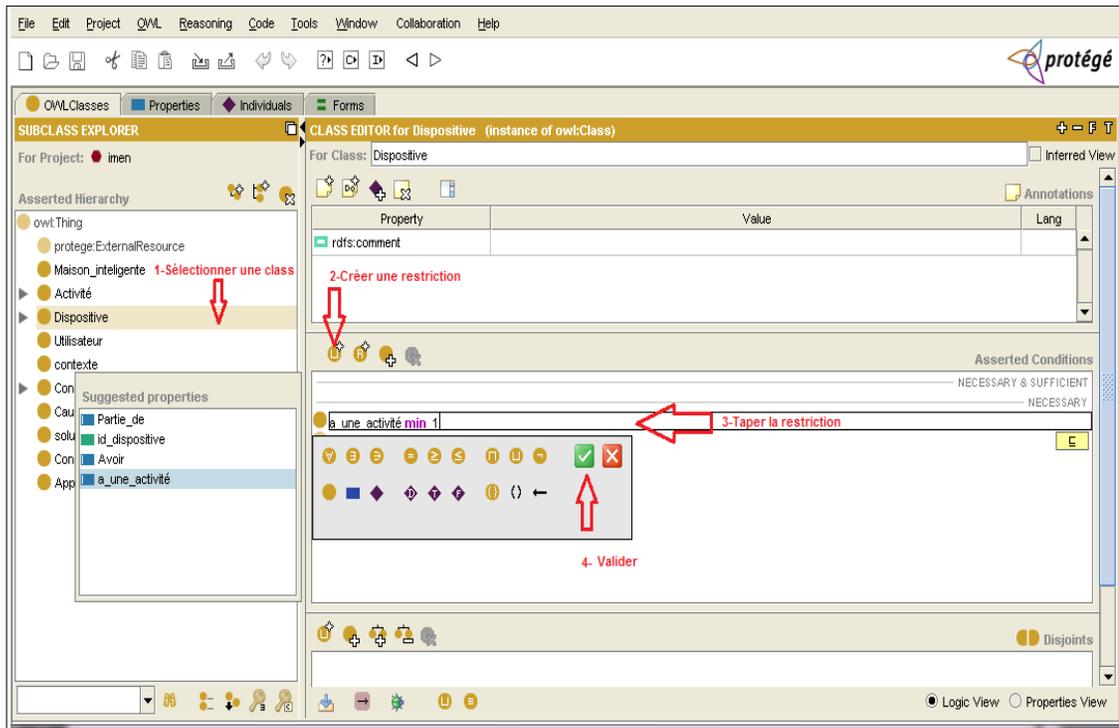


Figure 5.6 : Création d'une restriction sur une classe

5.2.5. Définition des instances :

Cette étape consiste à créer les instances des classes dans la hiérarchie. Définir une instance individuelle d'une classe exige

1. choisir une classe
2. créer une instance individuelle de cette classe
3. la renseigner avec les valeurs des attributs.

Par exemple, la Figure 5.7 montre la création d'un individu de la classe Propriétaire.

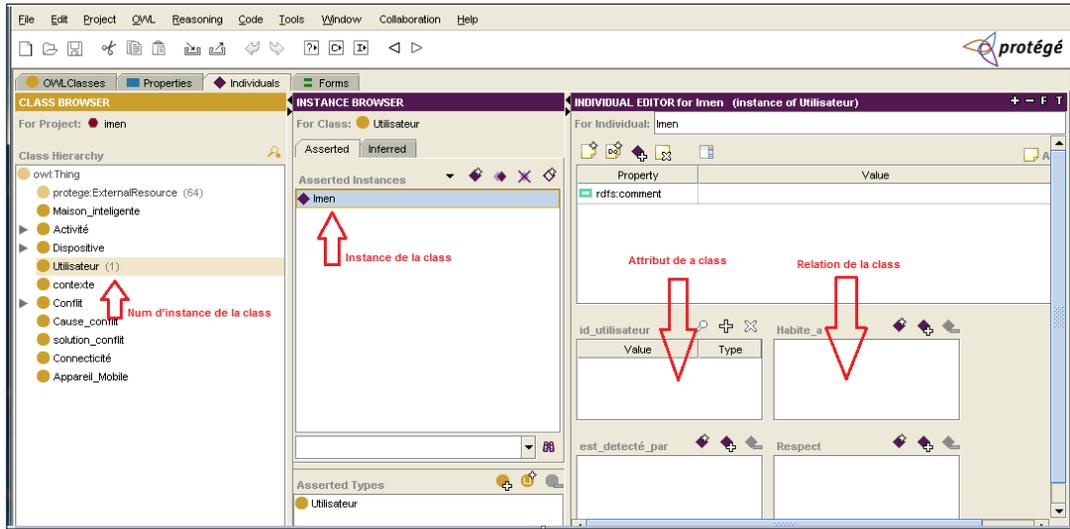


Figure 5.7 : Création des instances

5.2.6. Définition des règles SWRL :

La dernière étape consiste à créer les règles SWRL :

1. cliquer sur créer une nouvelle règle de l'interface de « SWRL Rules »
2. définir la règle dans la nouvelle interface affiche

Par exemple, la Figure 5.8 montre la création d'un individu de la classe Propriétaire.

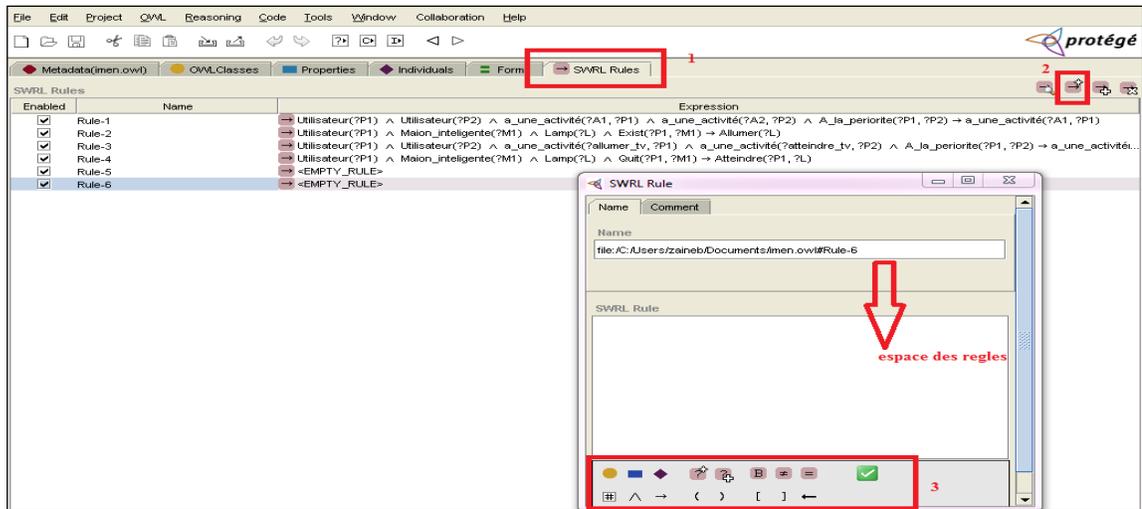


Figure 5.8 : Création des règles SWRL

5.2.7. Test de l'ontologie :

Nous avons utilisé le système Racer pour tester notre ontologie, nous distinguons deux types de test: test de consistance et test de satisfiabilité; le premier consiste à enlever l'inconsistance entre les concepts et cela en utilisant le test de subsumption incorporé au système Racer, par contre le deuxième permet de vérifier pour chaque concept l'existence des instances; un concept C est satisfiable si et seulement si, il existe au moins une interprétation I (instance) pour le concept C.

Racer se présente sous la forme d'un serveur qui peut être accédé par le protocole TCP ou HTTP. Donc, nous devons d'abord configurer la connexion au serveur hébergeant le système Racer. Mais ce que nous souhaitons, est de tester l'ontologie localement. Pour cela nous allons procéder comme suit :

- 1. Activer le menu OWL de Protégé.**
- 2. Choisir l'option Préférences...**
- 3. Dans la fenêtre qui s'affiche, vérifier que l'URL est « <http://localhost:8080> », sinon saisir puis valider en cliquant sur le bouton 'Close'.**

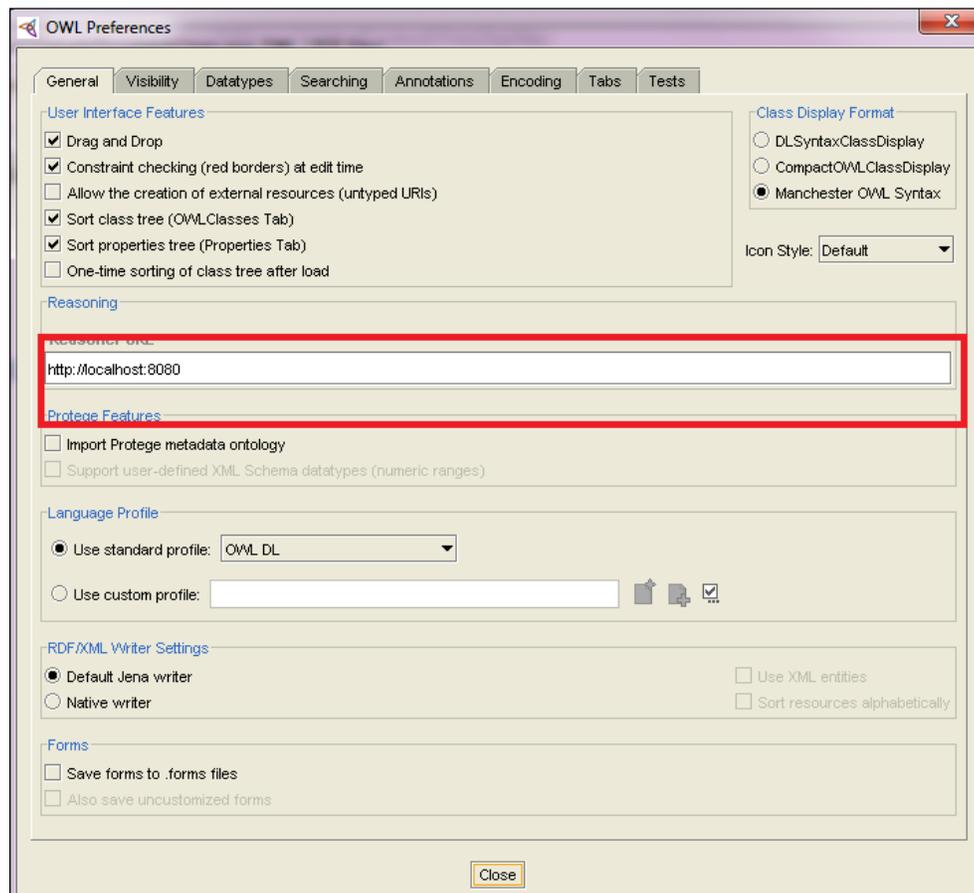


Figure 5.9 : Vérification de l'URL

4. Télécharger et exécuter Racer localement, le service HTTP va être activé sur le port 8080 de la machine local (localhost)

D'après les tests que nous avons appliqués à l'ontologie, aucune erreur ne s'est produite lors du test.



Figure 5.10 : Test de consistance

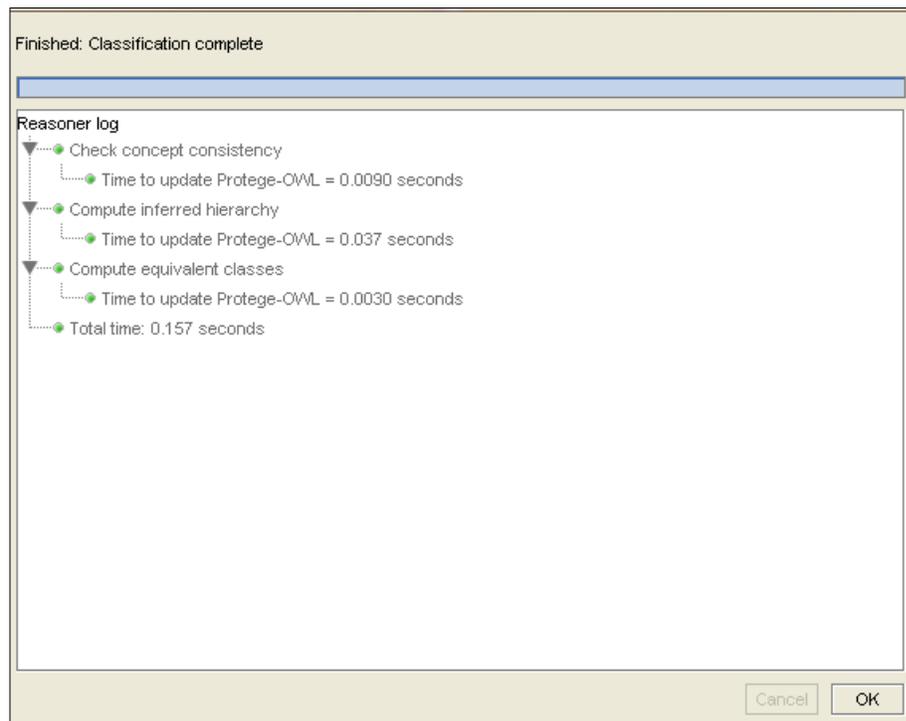


Figure 5.11: Test de classification

Après avoir développé notre ontologie, il nous reste à suivre son évolution, c'est-à-dire les nouveaux concepts à ajouter dans la partie terminologique (TBOX). Le résultat de cette étape est une nouvelle ontologie avec une nouvelle hiérarchie de concepts. Pour cela, nous proposons l'utilisation du raisonnement par classification qui est l'un des mécanismes de raisonnements de base pour les logiques de descriptions.

5.3. CONCLUSION

Dans ce chapitre, nous avons présenté notre implémentation, nous nous sommes basés sur la formalisation avec les logiques de description, et nous avons choisi le langage OWL pour codifier l'ontologie formelle. Ensuite, nous avons utilisé l'éditeur graphique PROTEGE OWL pour guider l'implémentation et produire un document OWL.



Conclusion Générale

1. Conclusion

Dans ce mémoire, nous avons proposé une ontologie de gestion des conflits sémantique pour une maison intelligente, dans laquelle plusieurs résidents interagissent avec différents types d'applications ensemble et simultanément. Nous avons introduit une ontologie qui décrit les services souhaités par les utilisateurs implémentés dans la détection de conflit ainsi qu'une approche qui attribue une méthode de résolution appropriée au conflit détecté. Sur la base de la méthode choisie, le conflit a été résolu soit par décision automatique c.-à-d. par priorité ou par préférence, soit par discussion. Grâce à cette proposition et leur implémentation, nous avons constaté que l'ontologie proposée peut gérer de manière flexible les conflits de plusieurs utilisateurs dans une maison intelligente.

Nous avons commencé par la représentation du domaine de travail « internet des Object & les smart cities » dans laquelle nous avons présenté la notion de iot, ses caractéristiques et leurs domaines. Nous avons découvert les principaux termes et notions correspondants au domaine d'ontologie en découvrant quelques méthodologies de construction des ontologies, avec les outils nécessaires à savoir les langages de spécification, les moteurs d'inférences, les éditeurs d'ontologies, ...etc et smart home à savoir les domaines d'applications et les caractéristiques d'un environnement ubiquitaire...etc. Nous avons, ensuite, présenté les différents travaux connexes sur la gestion des conflits sémantiques dans une maison intelligente où nous avons présenté les solutions existantes, critiqué ses solutions et construire une synthèse. Enfin, notre travail est présenté dans la deuxième partie sur le titre de conception d'une ontologie pour la résolution des conflits sémantique dans une maison intelligente où nous avons agrégé un ensemble des conflits leur causes, leurs solutions, et l'implémentation de l'ontologie.

2. Les perspectives

Ce travail est la première étape vers le développement d'une ontologie générale de gestion des conflits sémantiques dans une maison intelligente et nous prévoyons de poursuivre cette recherche.

Conclusion Générale

Le premier domaine d'étude future se concentrera sur la résolution d'autres types des conflits. Nous s'intéressons aussi d'introduire la technologie d'apprentissage automatique pour résoudre des nouveaux conflits non traités précédemment par notre architecture.

Malgré ses limites, nous prévoyons que cette étude jouera un rôle vital dans les maisons intelligentes et aussi dans les environnements intelligents, en utilisant à la fois la capacité des applications et des utilisateurs à résoudre les conflits.

Annexe :

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:xsp="http://www.owl-ontologies.com/2005/08/07/xsp.owl#"
  xmlns:swrlb="http://www.w3.org/2003/11/swrlb#"
  xmlns:swrla="http://swrl.stanford.edu/ontologies/3.3/swrla.owl#"
  xmlns="file:/C:/Users/zaine/Document/imen.owl#"
  xmlns:swrl="http://www.w3.org/2003/11/swrl#"
  xmlns:protege="http://protege.stanford.edu/plugins/owl/protege#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:sqwrl="http://sqwrl.stanford.edu/ontologies/built-ins/3.4/sqwrl.owl#"
  xml:base="file:/C:/Users/zaine/Document/imen.owl">
  <owl:Ontology rdf:about="file:/C:/Users/zaine/Document/imen.owl">
    <owl:imports rdf:resource="http://sqwrl.stanford.edu/ontologies/built-ins/3.4/sqwrl.owl"/>
    <owl:imports rdf:resource="http://swrl.stanford.edu/ontologies/3.3/swrla.owl"/>
  </owl:Ontology>
  <owl:Class rdf:ID="Solution_conflit">
    <rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:maxCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int">
          >1</owl:maxCardinality>
        <owl:onProperty>
          <owl:ObjectProperty rdf:ID="Concerne_un_conflit"/>
        </owl:onProperty>
      </owl:Restriction>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="Connection">
    <rdfs:subClassOf>
      <owl:Class rdf:ID="Contexte"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="Ouvrir_port">
    <rdfs:subClassOf>
      <owl:Class rdf:ID="Activité"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="Réduire">
    <rdfs:subClassOf>
      <owl:Class rdf:about="file:/C:/Users/zaine/Document/imen.owl#Activité"/>
    </rdfs:subClassOf>
```

```

</owl:Class>
<owl:Class rdf:ID="Thermometre">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="Dispositive"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Lamp">
  <rdfs:subClassOf>
    <owl:Class rdf:about="file:/C:/Users/zaineb/Documents/imen.owl#Dispositive"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Conflit_indirect">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="Conflit"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Conflit_entre_service">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="Conflit_direct"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Conflit_hybrid">
  <rdfs:subClassOf>
    <owl:Class rdf:about="file:/C:/Users/zaineb/Documents/imen.owl#Conflit"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Television">
  <rdfs:subClassOf>
    <owl:Class rdf:about="file:/C:/Users/zaineb/Documents/imen.owl#Dispositive"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Cause_conflit">
  <rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:ObjectProperty rdf:ID="A_une_solution"/>
      </owl:onProperty>
      <owl:maxCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
        >1</owl:maxCardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Reconnaissance">
  <rdfs:subClassOf>
    <owl:Class rdf:about="file:/C:/Users/zaineb/Documents/imen.owl#Activité"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Conflit_de_connexion">

```

```

    <rdfs:subClassOf rdf:resource="file:/C:/Users/zaineb/Documents/imen.owl#Conflit_entre_service"/>
</owl:Class>
<owl:Class rdf:ID="Envirenement">
    <rdfs:subClassOf rdf:resource="file:/C:/Users/zaineb/Documents/imen.owl#Contexte"/>
</owl:Class>
<owl:Class rdf:ID="Demande_Buffe">
    <rdfs:subClassOf>
        <owl:Class rdf:about="file:/C:/Users/zaineb/Documents/imen.owl#Activité"/>
    </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Conflit_entre_application">
    <rdfs:subClassOf>
        <owl:Class rdf:about="file:/C:/Users/zaineb/Documents/imen.owl#Conflit_direct"/>
    </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Radio">
    <rdfs:subClassOf>
        <owl:Class rdf:about="file:/C:/Users/zaineb/Documents/imen.owl#Dispositive"/>
    </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Détection">
    <rdfs:subClassOf>
        <owl:Class rdf:about="file:/C:/Users/zaineb/Documents/imen.owl#Activité"/>
    </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Conflit_de__periorite">
    <rdfs:subClassOf rdf:resource="file:/C:/Users/zaineb/Documents/imen.owl#Conflit_entre_application"/>
</owl:Class>
<owl:Class rdf:ID="Alarme">
    <rdfs:subClassOf>
        <owl:Class rdf:about="file:/C:/Users/zaineb/Documents/imen.owl#Dispositive"/>
    </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Maion_intelligente">
    <rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty>
                <owl:ObjectProperty rdf:ID="Exist"/>
            </owl:onProperty>
            <owl:minCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
                >1</owl:minCardinality>
        </owl:Restriction>
    </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="file:/C:/Users/zaineb/Documents/imen.owl#Conflit_direct">
    <rdfs:subClassOf>
        <owl:Class rdf:about="file:/C:/Users/zaineb/Documents/imen.owl#Conflit"/>
    </rdfs:subClassOf>

```

```

</owl:Class>
<owl:Class rdf:about="file:/C:/Users/zaineb/Documents/imen.owl#Dispositive">
  <rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:ObjectProperty rdf:ID="a_une_activité"/>
      </owl:onProperty>
      <owl:minCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
        >1</owl:minCardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:ObjectProperty rdf:ID="Avoir"/>
      </owl:onProperty>
      <owl:maxCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
        >1</owl:maxCardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Connectivite"/>
<owl:Class rdf:ID="Batterie">
  <rdfs:subClassOf rdf:resource="file:/C:/Users/zaineb/Documents/imen.owl#Dispositive"/>
</owl:Class>
<owl:Class rdf:ID="Appareil_mobile">
  <rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:ObjectProperty rdf:ID="Utilisé_par"/>
      </owl:onProperty>
      <owl:minCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
        >1</owl:minCardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Porte">
  <rdfs:subClassOf rdf:resource="file:/C:/Users/zaineb/Documents/imen.owl#Dispositive"/>
</owl:Class>
<owl:Class rdf:ID="Allumer">
  <rdfs:subClassOf>
    <owl:Class rdf:about="file:/C:/Users/zaineb/Documents/imen.owl#Activité"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Utilisateur">
  <rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
  <rdfs:subClassOf>

```

```

<owl:Restriction>
  <owl:maxCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
  >1</owl:maxCardinality>
  <owl:onProperty>
    <owl:ObjectProperty rdf:ID="habit_a"/>
  </owl:onProperty>
</owl:Restriction>
</rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Conflit_de_reconnaissance">
  <rdfs:subClassOf rdf:resource="file:/C:/Users/zaineB/Documents/imen.owl#Conflit_hybrid"/>
</owl:Class>
<owl:Class rdf:ID="Afficher">
  <rdfs:subClassOf>
    <owl:Class rdf:about="file:/C:/Users/zaineB/Documents/imen.owl#Activité"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="file:/C:/Users/zaineB/Documents/imen.owl#Activité">
  <rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:minCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
      >1</owl:minCardinality>
      <owl:onProperty>
        <owl:ObjectProperty rdf:ID="declanché_par"/>
      </owl:onProperty>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Fondation">
  <rdfs:subClassOf rdf:resource="file:/C:/Users/zaineB/Documents/imen.owl#Contexte"/>
</owl:Class>
<owl:Class rdf:ID="conflit_de_service">
  <rdfs:subClassOf rdf:resource="file:/C:/Users/zaineB/Documents/imen.owl#Conflit_hybrid"/>
</owl:Class>
<owl:Class rdf:ID="Augmenter">
  <rdfs:subClassOf rdf:resource="file:/C:/Users/zaineB/Documents/imen.owl#Activité"/>
</owl:Class>
<owl:Class rdf:ID="Localisation">
  <rdfs:subClassOf rdf:resource="file:/C:/Users/zaineB/Documents/imen.owl#Contexte"/>
</owl:Class>
<owl:Class rdf:about="file:/C:/Users/zaineB/Documents/imen.owl#Conflit">
  <rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:ObjectProperty rdf:ID="causé_par"/>
      </owl:onProperty>
    </owl:Restriction>
  <owl:minCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
  >1</owl:minCardinality>
  </rdfs:subClassOf>

```

La gestion de conflit sémantique dans une maison intelligente « Enjeux & Défi

```

    >1</owl:minCardinality>
  </owl:Restriction>
</rdfs:subClassOf>
</owl:Class>
<owl:ObjectProperty rdf:about="file:/C:/Users/zaineb/Documents/imen.owl#causé_par">
  <rdfs:domain rdf:resource="file:/C:/Users/zaineb/Documents/imen.owl#Conflit"/>
  <rdfs:range rdf:resource="file:/C:/Users/zaineb/Documents/imen.owl#Cause_conflit"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="Quit">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#InverseFunctionalProperty"/>
  <rdfs:range rdf:resource="file:/C:/Users/zaineb/Documents/imen.owl#Maion_intelligente"/>
  <owl:inverseOf>
    <owl:ObjectProperty rdf:about="file:/C:/Users/zaineb/Documents/imen.owl#Exist"/>
  </owl:inverseOf>
  <rdfs:domain rdf:resource="file:/C:/Users/zaineb/Documents/imen.owl#Utilisateur"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="file:/C:/Users/zaineb/Documents/imen.owl#Utilisé_par">
  <rdfs:range rdf:resource="file:/C:/Users/zaineb/Documents/imen.owl#Utilisateur"/>
  <rdfs:domain rdf:resource="file:/C:/Users/zaineb/Documents/imen.owl#Appareil_mobile"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="Respect">
  <rdfs:range rdf:resource="file:/C:/Users/zaineb/Documents/imen.owl#Connectivite"/>
  <rdfs:domain rdf:resource="file:/C:/Users/zaineb/Documents/imen.owl#Utilisateur"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="file:/C:/Users/zaineb/Documents/imen.owl#habit_a">
  <rdfs:range rdf:resource="file:/C:/Users/zaineb/Documents/imen.owl#Maion_intelligente"/>
  <rdfs:domain rdf:resource="file:/C:/Users/zaineb/Documents/imen.owl#Utilisateur"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="Allumer"/>
<owl:ObjectProperty rdf:about="file:/C:/Users/zaineb/Documents/imen.owl#declanché_par">
  <rdfs:range rdf:resource="file:/C:/Users/zaineb/Documents/imen.owl#Contexte"/>
  <rdfs:domain rdf:resource="file:/C:/Users/zaineb/Documents/imen.owl#Activité"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="file:/C:/Users/zaineb/Documents/imen.owl#Exist">
  <rdfs:range rdf:resource="file:/C:/Users/zaineb/Documents/imen.owl#Conflit"/>
  <rdfs:domain rdf:resource="file:/C:/Users/zaineb/Documents/imen.owl#Maion_intelligente"/>
  <owl:inverseOf rdf:resource="file:/C:/Users/zaineb/Documents/imen.owl#Quit"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="file:/C:/Users/zaineb/Documents/imen.owl#a_une_activité">
  <rdfs:domain rdf:resource="file:/C:/Users/zaineb/Documents/imen.owl#Dispositive"/>
  <rdfs:range rdf:resource="file:/C:/Users/zaineb/Documents/imen.owl#Activité"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="file:/C:/Users/zaineb/Documents/imen.owl#Avoir">
  <rdfs:domain rdf:resource="file:/C:/Users/zaineb/Documents/imen.owl#Dispositive"/>
  <rdfs:range rdf:resource="file:/C:/Users/zaineb/Documents/imen.owl#Connectivite"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="est_detecter_par">
  <rdfs:domain rdf:resource="file:/C:/Users/zaineb/Documents/imen.owl#Utilisateur"/>
  <rdfs:range rdf:resource="file:/C:/Users/zaineb/Documents/imen.owl#Détection"/>

```

```

</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="file:/C:/Users/zaineb/Documents/imen.owl#Concerne_un_conflit">
  <owl:inverseOf>
    <owl:ObjectProperty rdf:about="file:/C:/Users/zaineb/Documents/imen.owl#A_une_solution"/>
  </owl:inverseOf>
  <rdfs:range rdf:resource="file:/C:/Users/zaineb/Documents/imen.owl#Cause_conflit"/>
  <rdfs:domain rdf:resource="file:/C:/Users/zaineb/Documents/imen.owl#Solution_conflit"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="file:/C:/Users/zaineb/Documents/imen.owl#A_une_solution">
  <rdfs:range rdf:resource="file:/C:/Users/zaineb/Documents/imen.owl#Solution_conflit"/>
  <owl:inverseOf rdf:resource="file:/C:/Users/zaineb/Documents/imen.owl#Concerne_un_conflit"/>
  <rdfs:type rdf:resource="http://www.w3.org/2002/07/owl#InverseFunctionalProperty"/>
  <rdfs:domain rdf:resource="file:/C:/Users/zaineb/Documents/imen.owl#Cause_conflit"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="partie_de">
  <rdfs:range rdf:resource="file:/C:/Users/zaineb/Documents/imen.owl#Maion_intelligente"/>
  <rdfs:domain rdf:resource="file:/C:/Users/zaineb/Documents/imen.owl#Dispositive"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="Atteindre"/>
<owl:ObjectProperty rdf:ID="Allumer_tv"/>
<owl:ObjectProperty rdf:ID="A_la_periorite"/>
<rdf:Property rdf:ID="id_dispositive">
  <rdfs:domain>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <rdf:Description rdf:about="http://swrl.stanford.edu/ontologies/3.3/swrla.owl#RuleGroup"/>
        <owl:Class rdf:about="file:/C:/Users/zaineb/Documents/imen.owl#Dispositive"/>
      </owl:unionOf>
    </owl:Class>
  </rdfs:domain>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
</rdf:Property>
<owl:DatatypeProperty rdf:ID="id_maison">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
  <rdfs:domain rdf:resource="file:/C:/Users/zaineb/Documents/imen.owl#Maion_intelligente"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="id_conflit">
  <rdfs:domain rdf:resource="file:/C:/Users/zaineb/Documents/imen.owl#Conflit"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="adress_maison">
  <rdfs:domain rdf:resource="file:/C:/Users/zaineb/Documents/imen.owl#Maion_intelligente"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="modèle">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:domain rdf:resource="file:/C:/Users/zaineb/Documents/imen.owl#Appareil_mobile"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="E_mail">

```

```

<rdfs:domain rdf:resource="file:/C:/Users/zaineb/Documents/imen.owl#Utilisateur"/>
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="nom_utilisateur">
  <rdfs:domain rdf:resource="file:/C:/Users/zaineb/Documents/imen.owl#Utilisateur"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="id_utilisateur">
  <rdfs:domain rdf:resource="file:/C:/Users/zaineb/Documents/imen.owl#Utilisateur"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="id_activité">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
  <rdfs:domain rdf:resource="file:/C:/Users/zaineb/Documents/imen.owl#Activité"/>
</owl:DatatypeProperty>
<swrl:Imp rdf:ID="Rule-1">
  <swrl:head>
    <swrl:AtomList>
      <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
      <rdf:first>
        <swrl:IndividualPropertyAtom>
          <swrl:propertyPredicate
rdf:resource="file:/C:/Users/zaineb/Documents/imen.owl#a_une_activité"/>
          <swrl:argument2>
            <swrl:Variable rdf:ID="P1"/>
          </swrl:argument2>
          <swrl:argument1>
            <swrl:Variable rdf:ID="A1"/>
          </swrl:argument1>
        </swrl:IndividualPropertyAtom>
      </rdf:first>
    </swrl:AtomList>
  </swrl:head>
  <swrl:body>
    <swrl:AtomList>
      <rdf:first>
        <swrl:ClassAtom>
          <swrl:argument1 rdf:resource="file:/C:/Users/zaineb/Documents/imen.owl#P1"/>
          <swrl:classPredicate rdf:resource="file:/C:/Users/zaineb/Documents/imen.owl#Utilisateur"/>
        </swrl:ClassAtom>
      </rdf:first>
      <rdf:rest>
        <swrl:AtomList>
          <rdf:rest>
            <swrl:AtomList>
              <rdf:first>
                <swrl:IndividualPropertyAtom>
                  <swrl:argument2 rdf:resource="file:/C:/Users/zaineb/Documents/imen.owl#P1"/>
                  <swrl:propertyPredicate

```

```

rdf:resource="file:/C:/Users/zaineb/Documents/imen.owl#a_une_activité"/>
  <swrl:argument1 rdf:resource="file:/C:/Users/zaineb/Documents/imen.owl#A1"/>
  </swrl:IndividualPropertyAtom>
</rdf:first>
<rdf:rest>
  <swrl:AtomList>
    <rdf:first>
      <swrl:IndividualPropertyAtom>
        <swrl:propertyPredicate
rdf:resource="file:/C:/Users/zaineb/Documents/imen.owl#a_une_activité"/>
          <swrl:argument1>
            <swrl:Variable rdf:ID="A2"/>
          </swrl:argument1>
          <swrl:argument2>
            <swrl:Variable rdf:ID="P2"/>
          </swrl:argument2>
        </swrl:IndividualPropertyAtom>
      </rdf:first>
      <rdf:rest>
        <swrl:AtomList>
          <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
          <rdf:first>
            <swrl:IndividualPropertyAtom>
              <swrl:propertyPredicate
rdf:resource="file:/C:/Users/zaineb/Documents/imen.owl#A_la_periorite"/>
                <swrl:argument2 rdf:resource="file:/C:/Users/zaineb/Documents/imen.owl#P2"/>
                <swrl:argument1 rdf:resource="file:/C:/Users/zaineb/Documents/imen.owl#P1"/>
              </swrl:IndividualPropertyAtom>
            </rdf:first>
          </swrl:AtomList>
        </rdf:rest>
      </swrl:AtomList>
    </rdf:rest>
  </swrl:AtomList>
</rdf:rest>
</swrl:AtomList>
</rdf:rest>
</swrl:AtomList>
</rdf:rest>
<rdf:first>
  <swrl:ClassAtom>
    <swrl:classPredicate rdf:resource="file:/C:/Users/zaineb/Documents/imen.owl#Utilisateur"/>
    <swrl:argument1 rdf:resource="file:/C:/Users/zaineb/Documents/imen.owl#P2"/>
  </swrl:ClassAtom>
</rdf:first>
</swrl:AtomList>
</rdf:rest>
</swrl:AtomList>
</swrl:body>
</swrl:Imp>
<Porte rdf:ID="porte2"/>
<Appareil_mobile rdf:ID="mobile1">
  <Utilisé_par>

```

```

<Utilisateur rdf:ID="utilisateur_1">
  <id_utilisateur rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
  >1</id_utilisateur>
  <nom_utilisateur rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >père</nom_utilisateur>
  <E_mail rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >user1@gmail.com</E_mail>
</Utilisateur>
</Utilisé_par>
<modèle rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>sony</modèle>
</Appareil_mobile>
<Appareil_mobile rdf:ID="mobile2">
  <Utilisé_par>
    <Utilisateur rdf:ID="utilisateur_2">
      <nom_utilisateur rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      ></nom_utilisateur>
      <id_utilisateur rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
      >0</id_utilisateur>
      <E_mail rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      ></E_mail>
    </Utilisateur>
  </Utilisé_par>
  <modèle rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  ></modèle>
</Appareil_mobile>
<swrl:AtomList/>
<swrl:AtomList/>
<swrl:Imp rdf:ID="Rule-3">
  <swrl:head>
    <swrl:AtomList>
      <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
      <rdf:first>
        <swrl:IndividualPropertyAtom>
          <swrl:argument2 rdf:resource="file:/C:/Users/zaine/ Documents/ imen.owl#P1"/>
          <swrl:propertyPredicate
rdf:resource="file:/C:/Users/zaine/ Documents/ imen.owl#a_une_activité"/>
          <swrl:argument1>
            <swrl:Variable rdf:ID="allumer_tv"/>
          </swrl:argument1>
        </swrl:IndividualPropertyAtom>
      </rdf:first>
    </swrl:AtomList>
  </swrl:head>
  <swrl:body>
    <swrl:AtomList>
      <rdf:rest>
        <swrl:AtomList>
          <rdf:first>

```

```

<swrl:ClassAtom>
  <swrl:classPredicate rdf:resource="file:/C:/Users/zaineb/Documents/imen.owl#Utilisateur"/>
  <swrl:argument1 rdf:resource="file:/C:/Users/zaineb/Documents/imen.owl#P2"/>
</swrl:ClassAtom>
</rdf:first>
<rdf:rest>
  <swrl:AtomList>
    <rdf:first>
      <swrl:IndividualPropertyAtom>
        <swrl:propertyPredicate
rdf:resource="file:/C:/Users/zaineb/Documents/imen.owl#a_une_activité"/>
        <swrl:argument1 rdf:resource="file:/C:/Users/zaineb/Documents/imen.owl#allumer_tv"/>
        <swrl:argument2 rdf:resource="file:/C:/Users/zaineb/Documents/imen.owl#P1"/>
      </swrl:IndividualPropertyAtom>
    </rdf:first>
    <rdf:rest>
      <swrl:AtomList>
        <rdf:first>
          <swrl:IndividualPropertyAtom>
            <swrl:propertyPredicate
rdf:resource="file:/C:/Users/zaineb/Documents/imen.owl#a_une_activité"/>
            <swrl:argument1>
              <swrl:Variable rdf:ID="atteindre_tv"/>
            </swrl:argument1>
            <swrl:argument2 rdf:resource="file:/C:/Users/zaineb/Documents/imen.owl#P2"/>
          </swrl:IndividualPropertyAtom>
        </rdf:first>
        <rdf:rest>
          <swrl:AtomList>
            <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
          </rdf:rest>
          <rdf:first>
            <swrl:IndividualPropertyAtom>
              <swrl:argument1 rdf:resource="file:/C:/Users/zaineb/Documents/imen.owl#P1"/>
              <swrl:argument2 rdf:resource="file:/C:/Users/zaineb/Documents/imen.owl#P2"/>
              <swrl:propertyPredicate
rdf:resource="file:/C:/Users/zaineb/Documents/imen.owl#A_la_periorite"/>
            </swrl:IndividualPropertyAtom>
          </rdf:first>
          </swrl:AtomList>
        </rdf:rest>
      </swrl:AtomList>
    </rdf:rest>
  </swrl:AtomList>
</rdf:rest>
<rdf:first>
  <swrl:ClassAtom>
    <swrl:classPredicate rdf:resource="file:/C:/Users/zaineb/Documents/imen.owl#Utilisateur"/>
    La gestion de conflit sémantique dans une maison intelligente « Enjeux & Défi

```

```

        <swrl:argument1 rdf:resource="file:/C:/Users/zaineب/Documents/imen.owl#P1"/>
    </swrl:ClassAtom>
</rdf:first>
</swrl:AtomList>
</swrl:body>
</swrl:Imp>
<swrl:AtomList/>
<swrl:Imp rdf:ID="Rule-4">
    <swrl:head>
        <swrl:AtomList>
            <rdf:first>
                <swrl:IndividualPropertyAtom>
                    <swrl:argument1 rdf:resource="file:/C:/Users/zaineب/Documents/imen.owl#P1"/>
                    <swrl:argument2>
                        <swrl:Variable rdf:ID="L"/>
                    </swrl:argument2>
                    <swrl:propertyPredicate rdf:resource="file:/C:/Users/zaineب/Documents/imen.owl#Atteindre"/>
                </swrl:IndividualPropertyAtom>
            </rdf:first>
            <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
        </swrl:AtomList>
    </swrl:head>
    <swrl:body>
        <swrl:AtomList>
            <rdf:first>
                <swrl:ClassAtom>
                    <swrl:classPredicate rdf:resource="file:/C:/Users/zaineب/Documents/imen.owl#Utilisateur"/>
                    <swrl:argument1 rdf:resource="file:/C:/Users/zaineب/Documents/imen.owl#P1"/>
                </swrl:ClassAtom>
            </rdf:first>
            <rdf:rest>
                <swrl:AtomList>
                    <rdf:first>
                        <swrl:ClassAtom>
                            <swrl:classPredicate
rdf:resource="file:/C:/Users/zaineب/Documents/imen.owl#Maion_intelligente"/>
                            <swrl:argument1>
                                <swrl:Variable rdf:ID="M1"/>
                            </swrl:argument1>
                        </swrl:ClassAtom>
                    </rdf:first>
                    <rdf:rest>
                        <swrl:AtomList>
                            <rdf:first>
                                <swrl:ClassAtom>
                                    <swrl:classPredicate rdf:resource="file:/C:/Users/zaineب/Documents/imen.owl#Lamp"/>
                                    <swrl:argument1 rdf:resource="file:/C:/Users/zaineب/Documents/imen.owl#L"/>
                                </swrl:ClassAtom>
                            </rdf:first>

```

```

<rdf:rest>
  <swrl:AtomList>
    <rdf:first>
      <swrl:IndividualPropertyAtom>
        <swrl:argument1 rdf:resource="file:/C:/Users/zaineب/Documents/imen.owl#P1"/>
        <swrl:argument2 rdf:resource="file:/C:/Users/zaineب/Documents/imen.owl#M1"/>
        <swrl:propertyPredicate rdf:resource="file:/C:/Users/zaineب/Documents/imen.owl#Quit"/>
      </swrl:IndividualPropertyAtom>
    </rdf:first>
    <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
  </swrl:AtomList>
</rdf:rest>
</swrl:AtomList>
</rdf:rest>
</swrl:AtomList>
</rdf:rest>
</swrl:AtomList>
</swrl:body>
</swrl:Imp>
<swrl:Imp rdf:ID="Rule-2">
  <swrl:body>
    <swrl:AtomList>
      <rdf:first>
        <swrl:ClassAtom>
          <swrl:argument1 rdf:resource="file:/C:/Users/zaineب/Documents/imen.owl#P1"/>
          <swrl:classPredicate rdf:resource="file:/C:/Users/zaineب/Documents/imen.owl#Utilisateur"/>
        </swrl:ClassAtom>
      </rdf:first>
      <rdf:rest>
        <swrl:AtomList>
          <rdf:rest>
            <swrl:AtomList>
              <rdf:first>
                <swrl:ClassAtom>
                  <swrl:classPredicate rdf:resource="file:/C:/Users/zaineب/Documents/imen.owl#Lamp"/>
                  <swrl:argument1 rdf:resource="file:/C:/Users/zaineب/Documents/imen.owl#L"/>
                </swrl:ClassAtom>
              </rdf:first>
            </rdf:rest>
          </swrl:AtomList>
          <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
        </rdf:rest>
        <swrl:IndividualPropertyAtom>
          <swrl:argument1 rdf:resource="file:/C:/Users/zaineب/Documents/imen.owl#P1"/>
          <swrl:propertyPredicate rdf:resource="file:/C:/Users/zaineب/Documents/imen.owl#Exist"/>
          <swrl:argument2 rdf:resource="file:/C:/Users/zaineب/Documents/imen.owl#M1"/>
        </swrl:IndividualPropertyAtom>
      </rdf:first>
    </swrl:AtomList>
  </swrl:body>
</swrl:Imp>

```

```

    </rdf:rest>
  </swrl:AtomList>
</rdf:rest>
<rdf:first>
  <swrl:ClassAtom>
    <swrl:classPredicate
rdf:resource="file:/C:/Users/zaineb/Documents/imen.owl#Maion_intelligente"/>
    <swrl:argument1 rdf:resource="file:/C:/Users/zaineb/Documents/imen.owl#M1"/>
    </swrl:ClassAtom>
  </rdf:first>
</swrl:AtomList>
</rdf:rest>
</swrl:AtomList>
</swrl:body>
<swrl:head>
  <swrl:AtomList>
    <rdf:first>
      <swrl:ClassAtom>
        <swrl:argument1 rdf:resource="file:/C:/Users/zaineb/Documents/imen.owl#L"/>
        <swrl:classPredicate rdf:resource="file:/C:/Users/zaineb/Documents/imen.owl#Allumer"/>
      </swrl:ClassAtom>
    </rdf:first>
    <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
  </swrl:AtomList>
</swrl:head>
</swrl:Imp>
<Porte rdf:ID="porte1"/>
<swrl:AtomList/>
</rdf:RDF>

<!-- Created with Protege (with OWL Plugin 3.4.1, Build 536) http://protege.stanford.edu -->

```

REFERENCES

- [Ashton09]: Ashton K., Internet of Things, *RFiD J.*, 22, 97–114, 2009.
- [Sagar15]: Sagar Sukode, Shilpa Gite, Himanshu Agrawal, "CONTEXT AWARE FRAMEWORK IN IOT: A SURVEY", *International Journal of Advanced Trends in Computer Science and Engineering*, Volume 4, No.1, January – February 2015
- [V10] : Chui, M., Löffler, M. and Robets, R. (2010). *The Internet of Things*. McKinsey and Company: Chicago.
- [Atzori 08] : L. Atzori, A. Iera, and G. Morabito, "The internet of things : A survey," *Computer networks*
- [CITC13] : Analyse et perspectives de l'Internet des Objets : Horizons 2013-2020 CITCEura RFID
- [Friess11] : O. Vermesan, P. Friess, P. Guillemin, S. Gusmeroli, et al., "Internet of Things Strategic Research Agenda", Chapter 2 in *Internet of Things -Global Technological and Societal Trends*, River Publishers, 2011.
- [AC10] : Alain Coulon *L'Internet des Objets Un gisement à exploiter*, Club d'internet.
- [AC92] : Anne Cancellieri. *L'habitat du futur : défis et prospective pour le prochain quart de siècle*. Documentation française, 1992.
- [Gallissot 12] : Mathieu Gallissot. *_Modéliser le concept de confort dans un habitat intelligent : du multisensoriel au comportement. _ PhD thésis*. Grenoble, 2012.
- [Aldrich03] : Frances K. Aldrich. *_Smart Homes : Past, Present and Future. _ In: Inside the Smart Home*. Ed. by Richard Harper. London: Springer London, 2003, pp. 17_39. isbn: 978-1-85233-854-1. doi: 10.1007/1-85233-854-7_2. url: https://doi.org/10.1007/1-85233-854-7_2.
- [François 05] : François-Xavier Jeuland. *La maison communicante*. Eyrolles, 2005.

- [Edwards 01]: W Edwards and Rebecca Grinter. _At home with ubiquitous computing: Seven challenges._ In : Ubicomp 2001 : Ubiquitous Computing. Springer. 2001, pp. 256_272.
- [Paolo 08] : Silvestro Micera, Paolo Bonato, and Toshiyo Tamura. _Gerontechnology._ In: IEEE Engineering in Medicine and Biology Magazine 27.4 (2008), pp. 10_14.
- [Baldauf 07] : Matthias Baldauf, Schahram Dustdar, and Florian Rosenberg. _A survey on context-aware systems._ In: IJAHUC 2.4 (2007), pp. 263_277. doi: 10.1504/IJAHUC.2007.014070. url: <https://doi.org/10.1504/IJAHUC.2007.014070>.
- [Roy92] : Roy Want et al. _The Active Badge Location System._ In: ACM Trans.Inf. Syst. 10.1 (1992), pp. 91_102. doi: 10.1145/128756.128759. url: <http://doi.acm.org/10.1145/128756.128759>.
- [Bettini10] : Claudio Bettini et al. _A survey of context modelling and reasoning techniques._ In: Pervasive and Mobile Computing 6.2 (2010), pp. 161_180. doi: 10.1016/j.pmcj.2009.06.002. url: <https://doi.org/10.1016/j.pmcj.2009.06.002>.
- [Chen00] : Guanling Chen and David Kotz. A Survey of Context-Aware Mobile Computing Research. Tech. rep. Hanover, NH, USA, 2000.
- [Bachimont 03]:B. Bachimont, J. Charlet & R. Troncy, “Ontologies pour le Web Sémantique”. Action spécifique 32 CNRS / STIC Web sémantique Rapport final. 2003.
- [Gruber93] :T. Gruber, “A translation approach to portable ontology specification”, 1993.
- [Borst97] :W. N. Borst, “Construction of engineering ontologies”. University of Twente, Enschede, Centre for Telematica and Information Technology, 1997.

- [Grüninger95] : M. Gruninger and M.S. Fox, “Methodology for the Design and Evaluation of Ontologies”. In: Proceedings of the Workshop on Basic Ontological Issues in Knowledge Sharing, IJCAI-95, Montreal, 1995.
- [Guarino97] :N. Guarino, “Understanding, building, and using ontologies”. International Journal of Human- Computer Studies, 46: 293-310. 1997.
- [Gomez99] :Gomez Pérez A., Benjamins V.R. “Overview of Knowledge Sharing and Reuse Components : Ontologies and problem-Solving Methods”. Proceeding of the IJCAI-99 workshop on Ontologies and problem-Solving Methods (KRR5), Stockholm (Suède), pp. 1.1-1.15, 1999.
- [Staab00] :S. Staab, A. Maedche, “Axioms are objects too: Ontology engineering beyond the modeling of concepts and relations”, Research report 399, Institute AIFB, Karlsruhe, 2000.
- [Guarino98]:N. Guarino, “Formal Ontology and Information Systems”. Formal Ontology in Information Systems. IOS Press, 1998.
- [Uschold96] :M. Uschold and M. Grüninger, “ONTOLOGIES: Principles, Methods and Applications” Knowledge Engineering Review, 1996.
- [Lee02] :T.B. Lee et al., “The semantic Web”. in Scientific American, May 2002.
- [Uschold02] :M. Uschold and M.Gruninger, “Creating semantically integrated communities on the World Wide Web”. Honolulu: Semantic Web Workshop, 2002
- [Oberle04] :D. Oberle, R. Volz, B. Motik et S. Staab, “An extensible ontology software environment”. In S. Staab et R. Studer (Eds.), Handbook on Ontologies (pp. 299-320): Springer Verlag. 2004.
- [Gruber95] :T.R. Gruber, “Toward principles for the design of ontologies used for knowledge sharing”. International Journal of Human Computer Studies. 1995.
- [Sowa84] :J. Sowa, “Conceptual Structures: information processing in mind and machine”. Addison-Wesley, 1984.
- [Nardi03] :D. Nardi et R. J. Brachman. “An introduction to description

logics”. Dans Baader, F., Calvanese, D., McGuinness, D., Nardi, D. et Patel-Schneider, P. (éditeurs), “The Description Logic Handbook : Theory , Implementation and Applications”. Cambridge University Press, pp. 544. 2003.

- [Baader03a] :F. Baader, et W. Nutt, “Basic description logics”. Dans Baader, F., Calvanese, D., McGuinness, D., Nardi, D. et Patel-Schneider, P. (éditeurs), “The Description Logic Handbook : Theory, Implementation and Applications”. Cambridge University Press, pp. 47100. 2003.
- [Gomez02] : A. Gómez-Pérez, M. Fernandez-Lopez, O. Corcho, “OntoWeb: Ontology -based information exchange for knowledge management and electronic commerce”, Deliverable 1.3: A survey on ontology tools, IST-2000-29243, 31 st May, 2002
- [W3C04a] :Recommendation W3C RDF, 2004. <http://www.w3.org/TR/rdf-syntax-grammar/>.
- [Sirin06]:E. Sirin, B. Parsia, B. C. Grau, A. Kalyanpur, and Y. Katz, “Pellet : A practical owl-dl reasoned”. Submitted for publication to Journal of Web Semantics, 2006.
- [W3C06] :Candidate recommandation W3C SPARQL, 2006. <http://www.w3.org/TR/rdf-sparqlquery/>.
- [Haarslev01] : V. Haarslev and R. Möller, “Racer user’s guide and reference manual version 1.6”. Technica , report, University of Hamburg, Computer Science Department, 2001.
- [Sure02] :Y. Sure, M. Erdmann, J. Angele, S. Staab, R. Studer and Wenke, D., “OntoEdit: Collaborative Ontology Engineering for the Semantic Web”. In Proceedings of the International Semantic Web Conference 2002 (ISWC 2002), Sardinia, Italia, June 2002.
- [Noy01] :Natalya F. Noy and Deborah L. McGuinness, “Ontology Development 101: A Guide to Creating Your First Ontology”. Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880, March 2001.
- [Noy00] N. Noy, R. W. Ferguson and M. A.Musen. “The knowledge model of Protégé2000: combining interoperability and flexibility”. In Proceedings

of the International Conference on Knowledge Engineering and Knowledge Management (EKAW'00), 2000.

- [Troncy04] Raphaël Troncy, “Formalisation des connaissances documentaires et des connaissances conceptuelles à l’aide d’ontologies : application à la description de documents audiovisuels”. Thèse pour l’obtention du Doctorat de l’université Joseph Fourier – Grenoble, 2004.
- [Kimble95] K Kimbler and H Velthuijsen. Feature interaction benchmark. In: Third Feature Interaction Workshop (FIW'95). 1995.
- [Muffy00] Muffy Calder and Evan Magill. Feature Interactions in Telecommunications and software systems VI. IOS Press, 2000.
- [Plath98] Malte Plath and Mark Dermot Ryan. Plug-and-play Features. In: FIW. 1998, pp. 150-164.
- [Verena07] Verena Tuttlies, Gregor Schiele, and Christian Becker. Comity-conflict avoidance in pervasive computing environments. In: On the move to meaningful internet systems 2007: OTM 2007 workshops. Springer. 2007, pp. 763-772.
- [Henner11] Henner Jakob, Charles Consel, and Nicolas Lorient. Architecturing Conflict Handling of Pervasive Computing Resources. In: DAIS. Vol. 11. Springer. 2011, pp. 92-105.
- [Jane93] E Jane Cameron and Hugo Velthuijsen. Feature interactions in telecommunications systems. In: IEEE Communications Magazine 31.8 (1993), pp. 18-23.
- [Bouma94] LG Bouma and Hugo Velthuijsen. Feature interactions in telecommunications systems. IOS press, 1994.
- [Alma12] Alma L Juarez Dominguez. Detection of feature interactions in automotive active safety features. In: (2012).
- [Miki15] Miki Yagita, Fuyuki Ishikawa, and Shinichi Honiden. An application conflict detection and resolution system for smart homes. In: Proceedings of the First International Workshop on Software Engineering

for Smart Cyber-Physical Systems. IEEE Press. 2015, pp. 33-39.

- [Shin05] : Shin, C., Woo, W.: Conflict Resolution Method utilizing Context History for Context-Aware Applications. In: Proceedings of ECHISE 2005 - 1st International Workshop on Exploiting Context Histories in Smart Environments, Munich, Germany (2005)
- [Van10] : Van Nguyen T, Lim W, Nguyen H, Choi D, Lee C. Context ontology implementation for smart home. arXiv preprint arXiv:1007.1273. 2010 Jul 8.
- [Wang19] : Gu T, Wang XH, Pung HK, Zhang DQ. An ontology-based context model in intelligent environments. arXiv preprint arXiv:2003.05055. 2020 Mar 6.
- [Chen12] : Chen L, Nugent CD, Wang H. A knowledge-driven approach to activity recognition in smart homes. IEEE Transactions on Knowledge and Data Engineering. 2011 Feb 10;24(6):961-74.
- [Zhang 05] : Zhang D, Gu T, Wang X. Enabling context-aware smart home with semantic web technologies. International Journal of Human-friendly Welfare Robotic Systems. 2005 Jun;6(4):12-20.
- [Sezer 15] : Sezer OB, Can SZ, Dogdu E. Development of a smart home ontology and the implementation of a semantic sensor network simulator: An Internet of Things approach. In 2015 International Conference on Collaboration Technologies and Systems (CTS) 2015 Jun 1 (pp. 12-18). IEEE.
- [DIRKT17] : B.DIRKT, B. J. O. R. N. "A survey on information modeling and ontologies in building automation." (2017): 8615-8621.
- [Verstaevel18] : Verstaevel N, Garzone G, Monteil T, Guermouche N, Barthelemy J, Perez P. An ontology based context-aware architecture for smart campus applications. In 2018 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Ubiquitous Computing & Communications, Big Data & Cloud Computing, Social Computing & Networking, Sustainable Computing & Communications (ISPA/IUCC/BDCloud/SocialCom/SustainCom) 2018 Dec 11 (pp. 1056- 1063). IEEE.
- [Szilagyi16] : Szilagyi I, Wira P. Ontologies and Semantic Web for the Internet of Things-a survey. In IECON 2016-42nd Annual Conference of the

IEEE Industrial Electronics Society 2016 Oct 23 (pp. 6949-6954). IEEE.

- [Hajaj18] : Mémoire de fin d'étude « étude de cas sur un système medical domotique contrôlé par un SMA », Université Larbi Ben M'hidi Om EL Bouaghi , présenté par Hajaj Walid 2017-2018
- [Alok14]: Alok Kulkarni, Sampada Sathe (2014). Healthcare applications of the Internet of Things: A Review. (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (5) , 2014, 6229-6232
- [Thomas 19] :Johnson Thomas , Latest Trends In Smart Grid Technology In The Utilities Industry , 2019 , <https://medium.com/@mikethomsan/latest-trends-in-smart-grid-technology-in-the-utilities-industry-9e2f295d3a4f>, Consultee le 20/06/2020 a 23:10.
- [Bourougaa19] : Bourougaa-Tria , Salima « Ontologie et Web Sémantique » (2019) support de cours
- [Bourougaa16] Bourougaa-Tria, Salima, Hassina Seridi-Bouchelaghem, and Farid Mokhati. "An Ontology-Based Context Model to Manage Users Preferences And Conflicts." *Informatica* 40.1 (2016).
- [McCathy98] McCarthy, J.F. and Anagnost, T.D.: "Music FX: An arbiter of group preferences for computer supported collaborative workouts"; Proc. of CSCW, ACM Press, Seattle, Washington, USA (1998), 363-372.
- [Das06] Das, S.K., Roy, N., Roy, A.: "Context-Aware Resource Management in Multi-Inhabitant Smart Homes: A Nash H-Learning based Approach"; *Pervasive and Mobile Computing*, ELSEVIER, 2, 4(2006), 372-404
- [Shin08] Shin, C., Dey, A.K., Woo, W.: "Mixed-Initiative Conflict Resolution for Context-aware applications"; Proc. of ubiComp2008, ACM Press, Seoul, Korea (2008), 262- 271.
- [Kulkarni02] Kulkarni, A.: "A Reactive Behavioral System for the Intelligent Room"; M. Eng. Thesis for MIT, Cambridge, Massachusetts, USA (2002)
- [Haya06] Haya, P.A., Montoro, G., Esquivel, A., Garcia-Herranz, M., Alaman, X.: "A Mechanism for Solving Conflicts in Ambient Intelligent Environments"; *Journal of Universal Computer Science*, 12, 3(2006), 284- 296.

- [Park05] Park, I., Lee, K., Hyun, S., Yoon, H.: “A dynamic Context Conflict Resolution scheme for Group-aware ubiquitous computing”; Proc. of ubiPCMM at ubiComp2005, CEUR, Tokyo, Japan (2005), 42-47.
- [Shin07] Shin, C., Yoon, H., Woo, W.: “User-Centric Conflict Management for Media Services Using Personal Companions”; ETRI Journal, 29, 3(2007), 310-321.
- [Jameson04] Jameson, A., “More than the sum of its members: challenges for group recommender systems, Proceedings of the working conference on Advanced visual interfaces, Gallipoli, Italy (2004), pp. 8-54
- [Yu06] Yu, Z., Zhou, X. Hao, Y., Gu,J.: “TV program recommendation for multiple viewers based on user profile merging”; User Modeling and User Adapted Interaction, 16, 2(2006), pp. 62-82.
- [Wishart05] Wishart, R. Henricksen, K., Indulska1, J.: “Context Obfuscation for Privacy via Ontological Descriptions”; The Proceedings of the First International Workshop on Location and Context-awareness, Oberpfaffenhofen, Germany (2005), pp. 276~288.
- [Niu08] Niu, W., Kay J.: “Location Conflict Resolution with an Ontology”; The proceeding of Pervasive 2008, Sydney, Australia (2008), pp. 162–179.
- [futura20] site web : <https://www.futura-sciences.com/tech/definitions/internet-internet-objets-15158/> consulter le 20/avril/2020
- [Alain10] COULON, Alain. "L'internet des objets un gisement à exploiter." *La Lettre d'ADELI* n (2010).
- [Gattal17] », Gattal Elhachmi , mémoire master en informatique option SI : « L'Informatique Pervasive pour l'Internet of Things », Université Larbi tebessi 2017