

Rapport de projet Maison Intelligente



Réalisé par :

- ZAOUAM Sirageddine
- FARHI Abdellah
- HADJ-SAID Mohand-Arezki

Projet encadré par :

- Mme Amal Bouzeghoub

SOMMAIRE

1.	Introduction	3
2.	Contexte et objectif du projet	4
3.	Outils utilisés	6
4.	Conception de l'ontologie avec Protégé	7
5.	Dataset	9
6.	Les règles Sparql.....	10
7.	Élaboration des règles et vérification des inférences avec Jena et des requêtes SPARQL ...	12
8.	Détections des actions des utilisateurs ou habitants	12
9.	Conclusions et perspectives	13

1. Introduction :

Une maison intelligente est un paradigme qui se positionne comme un successeur de la domotique, bénéficiant des avancées de l'informatique omniprésente, également connue sous le nom d'informatique ambiante, y compris l'Internet des objets. Les maisons intelligentes ont la capacité d'augmenter le confort de l'habitant grâce, par exemple, à des interfaces naturelles pour contrôler la lumière, la température ou les différents types de lumière, de température, etc. Appareils électroniques. La gestion des ressources énergétiques est un autre problème des maisons intelligentes. Il est ainsi possible de mettre les appareils de chauffage en mode veille.

Lorsque les habitants sont absents ou adaptent automatiquement l'utilisation des ressources pour répondre aux besoins des habitants pour réduire le gaspillage de ressources énergétiques. En outre, un autre objectif essentiel de l'application des technologies de l'information est d'assurer la protection des personnes. Ceci est rendu possible par des systèmes capables d'anticiper des situations potentiellement dangereuses ou de réagir à des événements mettant en danger l'intégrité des personnes.

Les bénéficiaires de ces innovations peuvent être des individus autonomes, mais aussi des personnes fragiles avec une capacité de mouvement limitée. Par exemple, les personnes à autonomie limitée pourraient bénéficier d'applications domestiques intelligentes pour faciliter leur vie quotidienne ou rester en contact avec leurs proches. À cet égard, l'utilisation de la technologie représente une grande opportunité pour les personnes âgées vivant seules. Les systèmes intelligents peuvent également rappeler aux résidents quand ils doivent prendre leurs médicaments, ce qui leur permet de rester plus facilement en communication avec le monde extérieur ou même d'alerter les proches ou les services d'urgence en cas de chute accidentelle de la personne.

Chauffage, climatisation, éclairage, gestion des flux (eau, énergie, alimentation, déchets, information, etc.) et sécurité pourraient être gérés en tout ou en partie par un système informatique dédié, auto-apprentissage (centralisé ou non), interactif aux besoins des occupants, utilisant éventuellement des énergies et des ressources moins nuisibles à l'environnement. La maison s'adapterait aux habitudes et aux goûts de ses habitants et invités (éventuellement malvoyants, handicapés, âgés, malades, etc.), grâce au profilage de ces derniers, communiqué au système gérant la maison.

2. Contexte et objectif du projet :

Dans notre projet, nous allons réaliser une maison intelligente qui est une infrastructure informatique logicielle et matérielle capable de reconnaître les changements dans l'environnement ambiant tout et en traitant les données ainsi récoltées, proposer des services ou des configurations adaptées à ces changements.

La maison intelligente pourrait améliorer :

➤ Le confort :

Bien sûr, le fait d'automatiser sa maison a un véritable apport sur le confort qu'on y trouve. Plus besoin de se prendre une averse pour ouvrir le portail en rentrant à la maison, plus besoin de prendre froid en ouvrant les volets le matin, et fini les retours de week-end dans une maison toute froide. Aujourd'hui, une maison intelligente est capable de savoir quand vous rentrez à la maison (grâce à votre smart phone par exemple), et donc d'ouvrir le portail avant même que vous [Réalisation d'une maison intelligente à base d'Arduino] 8 n'arriviez. Les volets peuvent s'ouvrir et se fermer au rythme du soleil, et peuvent même aller jusqu'à s'adapter à la saison et la température pour laisser entrer la lumière et la chaleur du soleil l'hiver, ou au contraire conserver le frais l'été en fermant les volets des fenêtres exposées au soleil. De la même façon, votre maison sait quand vous êtes présent, et peut ainsi adapter elle-même votre chauffage pour que la maison soit toujours à la température idéale pour vous. Il est même possible de diffuser automatiquement votre Play List musicale préférée à votre réveil, ou quand vous rentrez à la maison. Pendant ce temps, un robot peut passer l'aspirateur dans la maison à votre place, et le système d'arrosage automatique arrosera votre jardin, tout en tenant compte des prévisions météo des prochains jours, histoire de ne pas arroser inutilement.

➤ . Les économies d'énergie :

En gérant les volets selon la saison, ainsi que le chauffage, le système domotique vous permet d'économiser de l'énergie, et donc de l'argent, même si au départ on ne recherchait que le confort en plus. La consommation d'énergie peut être suivie très finement, qu'il s'agisse de votre consommation d'électricité, d'eau, ou même de gaz. Vous partez de la maison ? Le simple fait d'activer l'alarme en partant va passer le chauffage en mode éco, et éteindre toutes les lampes et les appareils restés en veille, réduisant ainsi votre consommation d'énergie en votre absence. Et ceci sans aucune action de votre part. C'est cela la maison intelligente !

➤ **Communication :**

Un système domotique permet la communication non seulement à l'intérieur de la maison, mais aussi à l'extérieur. La technologie Internet interviendra de plus en plus pour la commande à distance par La certains utilisateurs. Vous ne devez même pas être à la maison pour commander vos appareils. Un simple coup de fil ou un sms vous permettra par exemple de régler le chauffage à distance, d'activer une simulation de présence ou de lancer le lave-vaisselle ou le lave-linge. Pratique, non?

➤ **La sécurité :**

Les automatismes que nous avons vus plus haut peuvent tout à fait contribuer à la sécurité de vos biens, en réalisant ce qu'on appelle une simulation de présence : même en votre absence, les volets continuent de s'ouvrir, de la musique peut être diffusée dans la maison, et des lumières allumées aléatoirement. Ainsi, de l'extérieur, il devient très difficile de savoir si la maison est inoccupée, ce qui dissuade de nombreux cambrioleurs. Une détection de fuite d'eau peut couper automatiquement l'arrivée d'eau afin d'éviter de gros dégâts. Mais la sécurité, c'est également la sécurité des personnes : en cas de détection d'incendie, par exemple, il est tout à fait possible d'ouvrir automatiquement les volets, déverrouiller les portes, et éclairer le chemin de la sortie pour faciliter l'évacuation. Exemple d'une mise en scène d'un « simple » détecteur de fumée, couplé à un système domotique. [Réalisation d'une maison intelligente à base d'Arduino] 9 Dans le cas de personnes dépendantes, telles que des seniors ou des personnes à mobilité réduite, il est également possible par exemple de prévenir un proche ou les services de secours en cas de chute ou tout évènement anormal. Pour se faire, nous avons imaginé un système intelligent capable de piloter les équipements électriques de la maison (lampes, volets, portes, appareils électroménagers, chauffage, etc.) selon un mode automatique :

Pour réaliser ce projet nous avons suivi les étapes suivantes :

- Représenter dans une ontologie web, les concepts à utiliser: capteurs, équipements électriques, informations propres à chaque composants.
- Définir des règles applicables pour certaines situations données selon des données reçues des différents capteurs.
- Reconnaître et Recommander des actions à effectuer selon le contexte et la règle applicable.
- Utilisation d'ontologies et de règles d'inférences pour raisonner.
- Récupération du contexte courant à travers des Capteurs.

- Implémenter le cœur du système pour automatiser la gestion de l'environnement (objets + capteurs) selon les données reçues et pour gérer les commandes manuelles.
- Effectuer des Tests selon différents scénarios d'exécution automatique.

Nos objectifs se sont concentrés sur la partie logicielle de la solution, en nous appuyant sur la base du TP qu'on a fait en cours intitulé « Internet des objets ».

3. Outils utilisés :

Dans cette section, nous définirons les concepts et technologie/outils utilisés afin de mettre en place notre maison intelligente.

Protégé c'est un éditeur d'ontologie permettant d'une part de déclarer :

- des classes OWL ;
- des propriétés OWL (owl:DataProperty et owl:ObjectProperty) ;
- des individus ;

Jena, ensemble d'outils pour la mise en place d'applications orientées Web sémantique. Parmi elle, on trouve une API Java avec laquelle on peut manipuler de nombreux langages tels que OWL, SPARQL, RDF/RDFS et de faire un raisonnement sur des modèles ontologiques à l'aide de moteurs d'inférences inclus dans Jena ou externes.

Les ontologies c'est un ensemble des objets reconnus comme existants dans un domaine. Ils permettent de construire des systèmes à base de connaissances.

SparkQL est un langage d'interrogation des données du Web Sémantique et ontologie.



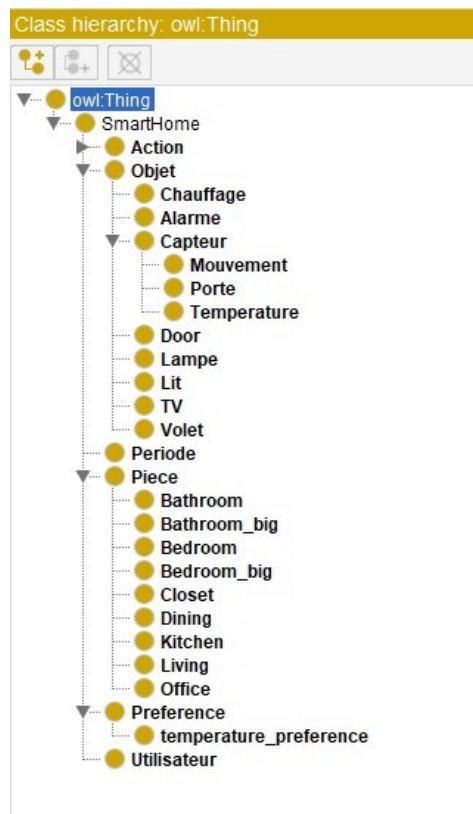
4. Conception de l'ontologie avec Protégé :

D'un point de vue conceptuel, nous avons compris que les équipements domotiques devaient faire partie du modèle de données que représente l'ontologie ainsi que les pièces dans lesquelles se trouvent ces équipements. Nous avons aussi ajouté une classe personnage pour modéliser la présence humaine. Ces classes ont pour but de nous aider à localiser les ressources et le propriétaire via les règles .

Puis nous avons décidé d'ajouter une classe «Preference» qui contient une sous classe «temperature_preference» pour modéliser la température actuelle.

Nous avons affecté à l'utilisateur deux « data properties » l'heure de réveil désirée et la température désirée « température_préférence ».

Ontologie finale :



Comme on peut le voir on a créé une classe **Objet** qui contient les équipements de la maison comme (Chauffage, Alarme, Lampe, etc). Ici on considère les capteurs (Mouvement, Porte, et Température) comme des sous classes de **Capteur** qui est elle-même une sous classe de **Objet** (donc un objet).

Aussi, nous avons donné aux capteurs (Mouvement et Porte), deux data properties représentant l'état du capteur (ON, OFF) de type xsd:string et la date de la mesure de type xsd:date.

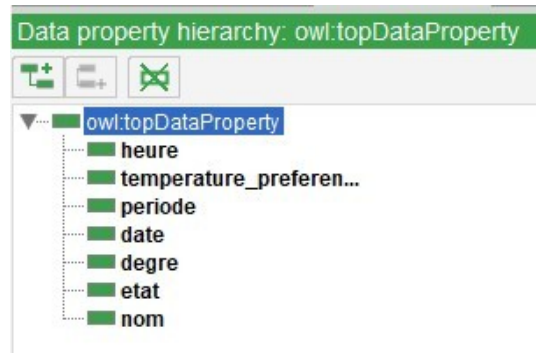
Pour le capteur température on lui a donné aussi deux data properties (degré) qui enregistrent la mesure de température et la date de cette mesure.

Pour tous les autres équipements de la maison comme (Volet, Chauffage, ect) on a utilisé le meme data property « état » pour décrire l'état de l'objet tout dépend de son type et de comment il fonctionne.

Exemple :

- ➔ Volet : état = « Ouvert » ou « Fermée »
- ➔ Chauffage : état = « ON » ou « OFF »
- ➔ Lampe : état = « allumée » ou « éteinte »

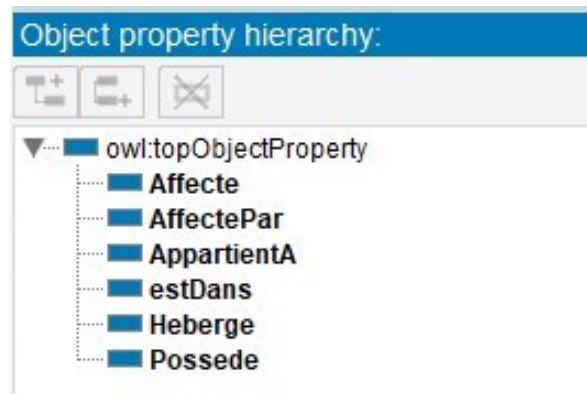
Comme on peut le voir dans la figure ci dessous les data properties qu'on a utilisé dans le projet :



Ensuite pour mettre en place les équipements de la maison ou les objets dans des pièces convenables (selon le dataset) on sera amené à utiliser des relations entre les classes comme :

- **estDans** : pour mettre un objet dans un pièce.
Exemple :
TV estDans Living, réfrigérateur estDans Kitchen, ect
- **Affecte** : Pour affecter une action à un objet
- **AffectePar** : l'inverse d'Affecte, une action est affectée par un objet
- **AppartientA** : une pièce appartient à un utilisateur
Exemple :
« Bedroom » appartientA « Sirageddine »
- **Heberge** : une pièce héberge un objet.
Exemple :
« Living » héberge « Table à manger »
- **Possede** : un utilisateur possède une pièce « Bedroom » par exemple.
Exemple :
« Abdellah » possède « Office »
« Mohand » possède «Kitchen »

Comme on peut le voir dans la figure ci dessous les object properties qu'on a utilisé dans notre ontologie :



5. Dataset :

C'est ensemble de données contient des données de capteur qui ont été collectées au domicile d'un adulte volontaire. Le résident de la maison était une femme. Les enfants et petits-enfants de la femme enfants et petits-enfants de la femme lui rendaient régulièrement visite.

Les activités suivantes sont annotées dans l'ensemble de données.

Le nombre entre parenthèses est le nombre de fois où l'activité apparaît dans les données.

Préparation du repas (1606)

Se détendre (2910)

Manger (257)

Travailler (171)

Dormir (401)

Laver la vaisselle (65)

Du lit à la toilette (157)

Entrer à la maison (431)

Sortir de chez soi (431)

Faire le ménage (33)

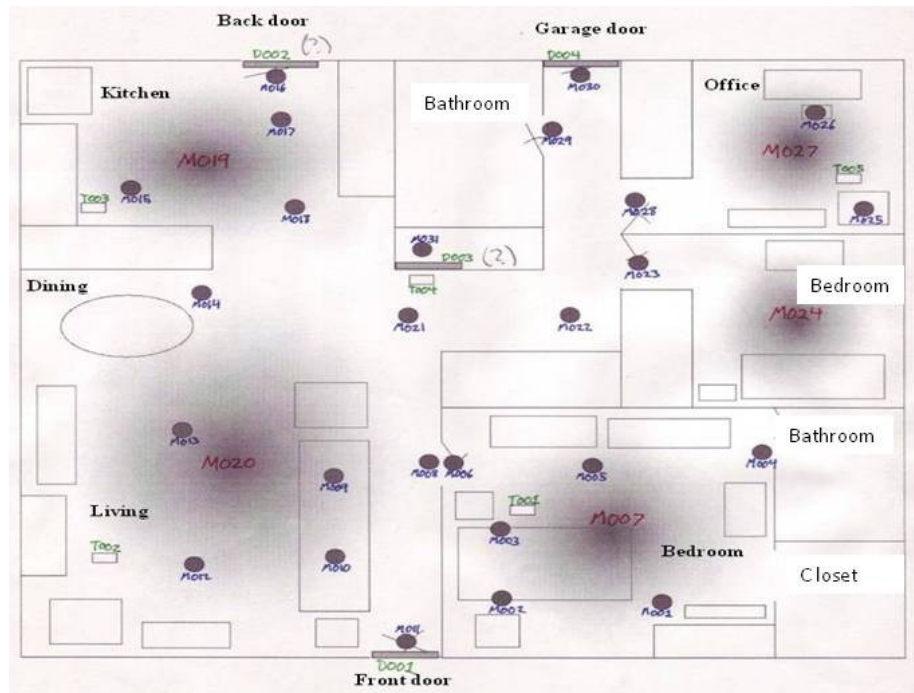
Resperate (6)

Les événements de capteur sont générés par des capteurs de mouvement (ces ID de capteur commencent M"), des capteurs de fermeture de porte (ces ID de capteurs commencent par "D") et des des capteurs de température (les ID de ces capteurs commencent par "T").

Exemple d'une portion de Dataset qui s'appelle Aruba :

```
2010-11-04 00:03:50.209589 M003 ON Sleeping begin
2010-11-04 00:03:57.399391 M003 OFF
2010-11-04 00:15:08.984841 T002 21.5
2010-11-04 00:30:19.185547 T003 21
2010-11-04 00:30:19.385336 T004 21
2010-11-04 00:35:22.24587 T005 20.5
2010-11-04 00:40:25.428962 T005 21
2010-11-04 00:45:28.658171 T005 20.5
```

La disposition des capteurs dans la maison est illustrée dans l'image ci dessous :



Source : WSU CASAS smart home project.

D. Cook. Learning setting-generalized activity models for smart spaces. IEEE Intelligent Systems, 2011.

6. Les règles SPARQL :

Notre application utilise les bibliothèques et l'API de Jena afin de manipuler notre modèle owl, d'effectuer de l'inférence dessus (pour pouvoir par exemple gérer certaines actions automatiques) et surtout d'exécuter des règles SPARQL sur cette structure de données.

Dans notre application, elles sont utilisées pour assurer les actions suivantes :

- ◆ éteindre la télé et la lampe s'il n'y a pas de mouvement dans la chambre

```
[rule1:    (?capteur_mv rdf:type ns:Mouvement) (?capteur_mv ns:estDans ?living)
          (?capteur_mv ns:etat "OFF")
          (?lampe rdf:type ns:Lampe) (?lampe ns:estDans ?living)
          (?tv rdf:type ns:TV) (?tv ns:estDans ?living)
->    (?tv ns:etat "eteinte"^^xsd:string)    (?lampe ns:etat "eteinte"^^xsd:string)
      (?capteur_mv ns:etat "OFF"^^xsd:string)
]
```

- ◆ S'il y a un objet Alarme dont l'etat est (ON) on regarde si les volets sont fermés, si c'est le cas on va les ouvrir

```
[rule2:      (?alarme rdf:type ns:Alarme) (?alarme ns:etat "ON")
              (?volet rdf:type ns:Volet) (?volet ns:etat "fermee")
              -> (?volet ns:etat "ouvert"^^xsd:string)
              ]
```

- ◆ On ouvre les volets de la pièce où la température est supérieure à 19°

```
[rule3:      (?capteur rdf:type ns:Temperature)
              (?capteur ns:estDans ?bedroom) (?capteur ns:date ?date)
              (?volet rdf:type ns:Volet) (?volet ns:etat "fermee")
              (?volet ns:estDans ?bedroom) (?capteur ns:degre ?value)
              greaterThan(?value, "19"^^xsd:double)
              -> (?volet ns:etat "Ouvert"^^xsd:string) (?volet ns:date ?date)
              ]
```

- ◆ On ferme les volets de la pièce où la température est inférieure à 19

```
[rule5:      (?capteur rdf:type ns:Temperature)
              (?capteur ns:estDans ?bedroom) (?capteur ns:date ?date)
              (?volet rdf:type ns:Volet)
              (?volet ns:estDans ?bedroom) (?capteur ns:degre ?value)
              lessThan(?value, "20"^^xsd:double)
              -> (?volet ns:etat "fermee"^^xsd:string) (?volet ns:date ?date)
              ]
```

- ◆ Si la température est inférieure à 10, on allume le chauffage de la pièce ou il se trouve le capteur de température

```
[rule4:      (?capteur rdf:type ns:Temperature)
              (?capteur ns:estDans ?bedroom) (?capteur ns:date ?date)
              (?chauffage rdf:type ns:Chauffage) (?chauffage ns:etat "OFF")
              (?chauffage ns:estDans ?bedroom) (?chauffage ns:degre ?value)
              greaterThan(?value, "10"^^xsd:double)
              -> (?chauffage ns:etat "Ouvert"^^xsd:string) (?chauffage ns:date ?date)
              ]
```

7. Élaboration des règles et vérification des inférences avec Jena et des requêtes SPARQL sous Eclipse (Java) :

Pour l'élaboration de règles suivantes, nous avons utilisé Jena et avec des requêtes SparQL, nous avons constaté le résultat des inférences.

les objets de kitchen

```
-----  
| obj |  
=====|  
| <http://www.semanticweb.org/win10/ontologies/2021/0/untitled-ontology-15#volet_kitchen> |  
| <http://www.semanticweb.org/win10/ontologies/2021/0/untitled-ontology-15#Chaise> |  
| <http://www.semanticweb.org/win10/ontologies/2021/0/untitled-ontology-15#Cuisiniere> |  
| <http://www.semanticweb.org/win10/ontologies/2021/0/untitled-ontology-15#Micro-ondes> |  
| <http://www.semanticweb.org/win10/ontologies/2021/0/untitled-ontology-15#refrigerateur> |  
| <http://www.semanticweb.org/win10/ontologies/2021/0/untitled-ontology-15#table_dining> |  
| <http://www.semanticweb.org/win10/ontologies/2021/0/untitled-ontology-15#Table%20a%20manger> |  
-----
```

Puis on teste si Jena a bien interprété le raisonnement qu'on a fait (les règles).

L'état du volet kitchen quand T=22

```
-----  
| etat | date |  
=====|=====|  
| "Ouvert"^^<http://www.w3.org/2001/XMLSchema#string> | "23:25:16.840313"^^<http://www.w3.org/2001/XMLSchema#string> |  
-----
```

L'état du volet de living T<20 (T=19)

```
-----  
| etat | date |  
=====|=====|  
| "fermee"^^<http://www.w3.org/2001/XMLSchema#string> | "23:24:42.216445"^^<http://www.w3.org/2001/XMLSchema#string> |  
-----
```

C'est bien le cas, comme indiqué dans la règle on ouvre les volets si la température est > 22° si non on les ferme.

8. Détection des actions de l'utilisateur :

Comme on a pu aussi détecter les actions établies par l'habitant en analysant à chaque itération de la fonction qui lit le dataset le type de mouvement et dans quel pièce.

Exemple : L'utilisateur mange dans la salle

```
Update.....>M014  
  
=====> Quelqu'un mange dans la salle  
Update.....>M020  
  
=====> Quelqu'un mange dans la salle  
Update.....>M020  
  
=====> Quelqu'un mange dans la salle  
Update.....>M014  
Update.....>M020  
Update.....>M020  
Update.....>M008
```

ou bien l'utilisateur est dans la douche en ce moment là

```
Update.....>M020  
Update.....>M023  
Update.....>M022  
Create.....>M031  
  
=====> Quelqu'un dans la douche  
Update.....>M021
```

9. Conclusion et perspectives :

La domotique est une solution pour le confort mais aussi pour les personnes avec des difficultés mobiles. Au cours de ce projet nous avons pu apprécier la force de l'ontologie qui est la base pour la modélisation de notre maison intelligente, qui est capable de gérer l'éclairage des chambres et les portes, surveillé l'état de la température et l'humidité.

Ainsi on en conclut qu'une bonne modélisation de la maison facilite également la mise en œuvre des règles d'inférence et le contrôle de la maison intelligente. Comme on a pu aussi découvrir de meilleures connaissances des applications de la domotique, ce qui pourrait nous être fortement utile pour notre vie professionnelle future. On aurait aimé que ce travail soit déroulé dans les meilleures conditions possible pour qu'on puisse faire le test de notre application avec des capteurs réels ce qui est l'objectif global du projet.

Enfin, le seul point « négatif », serait sûrement le manque de temps pour pouvoir encore approfondir ce travail, car ce dernier ne s'arrête pas ici il a encore plusieurs tâches qui peuvent être améliorées. En effet, beaucoup de possibilités s'offrent aux passionnés de domotique, tant sur le matériel disponible que sur les actions à réaliser. Nous pensons avoir entraperçu une partie de notre future vie active.