

Note technique projet système expert

Julie Courgibet

Un système expert est composé de 3 éléments :

- Une base de faits ;
- Une base de règles ;
- Un moteur d'inférence.

1. La base de faits

La base de faits est un dictionnaire (objet Python), avec 7 clés :

- **mvt** : la valeur correspondante est un booléen qui traduit si il y a du mouvement dans l'appartement (si une des valeurs des 4 capteurs de mouvement : pièce de vie, salle de bain, cuisine, couloir).
- **dureeSDB** : la valeur correspondante est un entier (int). Il s'agit du temps durant lequel la personne est dans la salle de bain. Valeur remise à zéro lorsque la personne quitte la pièce.
- **dureeCuisine** : la valeur correspondante est un entier (int). Il s'agit du temps durant lequel la personne est dans la cuisine. Valeur remise à zéro lorsque la personne quitte la pièce.
- **dureeTsup25** : la valeur correspondante est un entier (int). Il s'agit du temps durant lequel la température est au-dessus de 25°C. Valeur remise à zéro lorsque la température redescend en dessous de 25°C.
- **presence** : la valeur est une chaîne de caractères (string). Il y a 4 valeurs possibles : vie, sdb, couloir, cuisine. Elle permet de connaître la position de la personne (dernière pièce dans laquelle du mouvement a été détecté).
- **alerte** : la valeur correspondante est un booléen qui traduit si une des règles a été violées.
- **messageAlerte** : la valeur correspondante est une chaîne de caractères (string). C'est le message qui est affiché pour connaître la règle non respectée.

2. La base de règles

a. Classe objet

Une classe objet mère a été créée. Elle est constituée de 3 attributs et 2 méthodes :

```
1 class Regle(object):
2     """Classe mère Règle implémentée dans système expert"""
3     def __init__(self, numero, definition, base_faits):
4         self.numero = numero
5         self.definition = definition
6         self.base_faits = base_faits
7
8     def application(self):
9         pass
10
11     def get_base_faits(self):
12         return self.base_faits
13
```

b. Règle 1 : Mise à jour de la base de faits

La classe Regle1 hérite de la classe Regle et permet la lecture d'une colonne de données contenues dans le fichier simulation. Elle traduit les données des différents capteurs et met à jour la base de faits.

c. Autres règles :

Six autres classes ont été créées et mettent en place les différentes règles à respecter.

La méthode « application » respecte la forme d'une règle :

Si (ensemble de faits) => (nouveau fait)

```
# Règle 6:
class Regle6(Regle):
    """classe fille de Regle pour remonter alerte si la personne sort avec présence dans couloir."""
    def __init__(self, base_faits):
        Regle.__init__(self, 6, "La personne sort.", base_faits)

    def application(self):
        if self.base_faits["presence"] == "couloir" and self.base_faits["porte"] :
            self.base_faits["alerte"] = True
            self.base_faits["messageAlerte"] = "La personne sort."
```

C'est un choix technique que toutes ces règles ne modifient que les clés alerte et messageAlerte. Cela permet de gérer une liste d'attente de règles sans modifier la base de faits à chaque appel/application d'une règle.

3. Moteur d'inférence

Chainage avant.

On suit le même algorithme que celui donné en cours :

Pour chaque colonne du fichier (simulation.xlsx) (si pas alerte) :

Mise à jour de la base de faits (avec Regle1)

Liste qui contient toutes les règles à appliquer (base de règles)

WHILE (not Alerte et liste de règles non vide):

Application de la première règle de la liste

Suppression de la règle de la base de règle

IF (alerte):

print(messageAlerte)

Break

4. Quelques exemples de résultats

```
$ python main.py
Temps : 0 secondes
Temps : 2 secondes
Temps : 4 secondes
Temps : 6 secondes
Temps : 8 secondes
Temps : 10 secondes
Temps : 12 secondes
Temps : 14 secondes
Temps : 16 secondes
Temps : 18 secondes
Temps : 20 secondes
Temps : 22 secondes
Il y a une intrusion.
```

```
Temps : 2 secondes
Temps : 4 secondes
Temps : 6 secondes
Temps : 8 secondes
Temps : 10 secondes
Temps : 12 secondes
Temps : 14 secondes
Temps : 16 secondes
Temps : 18 secondes
Temps : 20 secondes
Temps : 22 secondes
La personne sort.
```

```
Temps : 812 secondes
Temps : 814 secondes
Temps : 816 secondes
Temps : 818 secondes
Temps : 820 secondes
Temps : 822 secondes
Temps : 824 secondes
Temps : 826 secondes
Temps : 828 secondes
Temps : 830 secondes
Temps : 832 secondes
Temps : 834 secondes
Temps : 836 secondes
Temps : 838 secondes
Temps : 840 secondes
Présence supérieure à 10 minutes dans la cuisine
```

```
Temps : 280 secondes
Temps : 282 secondes
Temps : 284 secondes
Temps : 286 secondes
Temps : 288 secondes
Temps : 290 secondes
Temps : 292 secondes
Temps : 294 secondes
Temps : 296 secondes
Temps : 298 secondes
Temps : 300 secondes
Température supérieure à 25°C.
```