**IA01-TP03**

**Conduite d’expertise d’un système expert d’ordre 0+**

Maxime DILY, Simon LAURENT

Université

de

Technologie

de

Compiègne

Ce document fait office de compte rendu du TP03.

Il a pour but d’expliquer les différentes étapes du développement de notre système expert

De la réflexion du sujet jusqu’à la programmation de ce dernier en détaillant les difficultés rencontrées

Bonne dégustation oenologique

Automne 2015

Responsable UV :

Marie-Helene ABEL

Responsable TD/TP :

Dominique LENNE

Table des matières

I. Énoncé du sujet 3

II. Formalisation de la problématique 4

1. Définition des termes 4

2. Choix du sujet 4

3. Introduction 5

III. Représentation des connaissances 5

1. Base de connaissance 5

2. Base de fait 6

3. Base de règles 6

4. Jeux d’essais 6

*5.* Arbre de déduction 6

IV. Programmation du Système expert 7

1. Représentation Lisp 7

2. Moteur d’inférences 7

3. Fonctions de services 7

4. Résultats 7

V. Conclusion 8

# Énoncé du sujet

Conduite d’expertise d’un SE d’ordre 0+

Dates de remise :

* -  Lundi 30 novembre 2015 à 18H pour la réponse à la question 1.
* -  Lundi 4 janvier 2016 à 18H pour la réponse aux questions 2 et 3.
* -  Démonstration et présentation orale lors du dernier TD.
* L’objet du TP03 est de réaliser le développement d’un SE de sa phase d’expertise à sa phase d’utilisation. A cette fin, vous devez :
  1. Formalisez une problématique d’un domaine au choix (un qui vous passionne) qui puisse être traitée par un SE d’ordre 0+. Justifiez votre choix et faites-le valider par votre chargé de TD.
  2. Déterminez les connaissances nécessaires au SE : explicitez votre base de règles (donnez vos sources). Présentez l’arbre de déduction associé et donnez des jeux d’essais.
  3. Programmez votre SE
     1. Justifiez la représentation Lisp choisie pour exploiter les faits et la base de  règles.
     2. Développez, justifiez et commentez le moteur d’inférences choisi: chaînage  avant (ou arrière) en profondeur d’abord, chaînage avant (ou arrière) en largeur  d’abord.
     3. Testez votre moteur et commentez les résultats. Une comparaison avec un  deuxième moteur développé serait un plus.
* Documents à produire :
*   Un rapport écrit comportant les réponses aux points précédents et présentant des  scénarios d’utilisation.
*   Un fichier comportant le code lisp de votre SE avec les scénarios d’utilisation (à  envoyer par courriel).
*   Une courte présentation orale s’appuyant sur des transparents et une démonstration  sont attendues au cours du dernier TD.

# Formalisation de la problématique

## Définition des termes

Le sujet de ce dernier TP est de réaliser un système expert d’ordre 0+. Nous allons tout d’abord expliciter ce que l’on nous demande, c'est-à-dire ce qu’est un système expert, puis l’ « ordre 0+ ».

* Système expert : système permettant la résolution de problèmes difficiles, en particulier remplacer un expert humain, ou au moins l’aider dans sa démarche d’expertise. Un système expert repose sur trois composantes : la base de faits, la base de règles et le moteur d’inférence. Il existe trois types de systèmes experts définis par leur ordre : 0, 0+ ou 1. Le choix de l'ordre détermine le formalisme des règles.
* Ordre 0+ : Les règle seront sous la forme d’un couple attribut valeur.

La première partie de ce TP, consistait à choisir un sujet qui pouvait correspondre à un système expert d’ordre 0+.

## Choix du sujet

Notre première idée fut de modéliser le très célèbre jeu Akinator pour en faire une version automobile. Cependant, ce sujet était trop facile et peu intéressant. Nous avons dans un second temps choisit de modéliser un système expert pour le choix des UVs en Génie Informatique à l’UTC. Néanmoins, nous fîmes face à un refus car ce sujet était trop rependu. Il avait déjà été fait à de nombreuses reprises. Ainsi nous avons décidé de faire un système expert pour l’œnologie.

Cependant, nos premiers tests permettant la résolution ne convergeaient pas vers la structure d'un système expert à proprement parler. En effet nous étions parti vers une structure de base de donnée. Suite à un rendez-vous avec Mme ABEL et à ses conseils, nous avons légèrement modifier notre système afin qu’il réponde aux critères. Finalement nous traiterons d’un système d'aide à l’achat d’un vin. Dans le but de cadrer le sujet nous limiterons notre système à la Bourgogne, le Champagne et le Bordelais. En effet, nous avons la chance d’avoir un terroir viticole de plus de 130 000 vins différents.

Notre base de règles est modélisée selon un formalisme d’ordre 0+. Cette dernière fonctionnera à partir des réponses aux questions entrées par l’utilisateur. Nous basons notre expertise sur le fameux « Guide Hachette des Vins 2016 ». Quand à notre moteur d’inférence, il fonctionnera en chainage avant. Le but de ce système expert est qu’il vous retourne une liste de vins en fonction de vos critères.

## Introduction

A l’approche des fêtes de noël vous décidez de vous mettre en quête des cadeaux à offrir à vos proches. Problème, cette année vous ne savez pas quoi offrir à votre père. En effet ce dernier dans la cinquantaine possède déjà tout pour être heureux. Alors après avoir passé en revue l’ensemble des derniers produits Hi-Tech et autres objets matériels qui finiront au grenier ou en dépôt vente, vous décidez de lui offrir l’une des bonnes choses de ce monde, une bouteille de vin. Nouvelle difficulté, votre père étant un épicurien vous ne pouvez pas acheter la première bouteille venue. Vous décidez donc de vous en remettre à un caviste. Néanmoins, cela ne vous suffit pas. Vous avez peur de l’arnaque. C’est là que vous réalisez que vous avez à disposition un système expert pour vous aider dans cette noble quête. Bonne dégustation.

# Représentation des connaissances

## Base de connaissance

Afin d’effectuer notre expertise, nous avons modéliser notre base de connaissances en nous basant sur le Guide Hachette des Vins 2016 (http://www.hachette-vins.com/). Nous avons donc commencé par saisir à la main nos connaissances en les structurant sous forme de a-list de la façon suivante :

* *((Nom CARRUADES\_DE\_LAFITE) (Prix 105) (Annee 2009) (Couleur ROUGE) (Petillant NON) (Geographie BORDELAIS) (Appellation PAUILLAC) (Bio NON) (Note 2) (Medaille NON) (Plat AU\_VERRE VIANDE\_NOIRE VIANDE\_ROUGE VIANDE\_BLANCHE) (Garde 0) (Cepage CABERNET))*

Malheureusement, il n’était pas possible de travailler correctement avec une telle structure. Cependant afin de les structurer correctement, nous devions parser le fichier.txt générer à la main pour obtenir un autre format transitoire. Pour effectuer cela, nous avons utilisé le cours de SR01 sur le python afin de créer un script faisant le travail pour nous. Nous avons donc obtenu un format de la forme :

* *CARRUADES\_DE\_LAFITE,105,2009,ROUGE,NON,BORDELAIS,PAUILLAC,NON,2,NON,AU\_VERRE VIANDE\_NOIRE VIANDE\_ROUGE VIANDE\_BLANCHE,0,CABERNET,*

Afin d’aboutir à notre format de donnée, nous avons de nouveau fait un script en python qui nous a générer les 176 vins de notre base de connaissance comme suit :

* *(setq vin1 (make-instance '$vin :nom "CARRUADES\_DE\_LAFITE" :prix 105 :annee 2009 :couleur "ROUGE" :petillant "NON" :geographie "BORDELAIS" :appellation "PAUILLAC" :bio "NON" :note 2 :medaille "NON" :plat "AU\_VERRE VIANDE\_NOIRE VIANDE\_ROUGE VIANDE\_BLANCHE" :garde "NON" :cepage "CABERNET"))*

Remarque : Le code des scripts sera joint dans un fichier.py

## Base de fait

Nous avons chois de représenter la base de fait « \*BF\* » comme une liste de faits *(fait\_1 fait\_2…)*, où chaque fait correspond à un couple (attribut valeur). Exemple :

* *((GARDE "NON") (MEDAILLE "NON") (NOTE 2)…)*

## Base de règles

La base de règles « \*BR\* » est représentée sous forme de liste de règles (R1R2...RN). Une règle R contient une liste de critères et un nom de vin. Exemple :

* *R1 : (((PRIX 105) (ANNEE 2009) (COULEUR "ROUGE") (PETILLANT "NON") (GEOGRAPHIE "BORDELAIS") (APPELLATION "PAUILLAC") (BIO "NON") (NOTE 2) (MEDAILLE "NON") (GARDE "NON") ...) "CARRUADES\_DE\_LAFITE")*

Une règle n’est déclenchée que si que si ses critères sont satisfaits (relativement à la base de fait). Seulement alors, la base de fait est mise à jour avec les critères. Une fois la base de fait remplie, on la compare avec la base de connaissance afin d’aboutir à la liste des vins correspondant à nos critères.

## Jeux d’essais

* Vous voulez un vin rouge entre 10€ et 20€

*\*BF\* :((GARDE "NON") (MEDAILLE "NON") (NOTE 0) (BIO "NON") (PETILLANT "NON") (ANNEEMAX 2015) (ANNEEMIN 1990) (PRIXMAX 20)*

*(PRIXMIN 10) (GEOGRAPHIE ("CHAMPAGNE" "BORDELAIS" "BOURGOGNE")) ...)*

* Vous voulez un vin bio

*\*BF\* :((GARDE "NON") (MEDAILLE "NON") (NOTE 0) (BIO "OUI") (PETILLANT "NON") (ANNEEMAX 2015) (ANNEEMIN 1900) (PRIXMAX 1000)*

*(PRIXMIN 0) (GEOGRAPHIE ("CHAMPAGNE" "BORDELAIS" "BOURGOGNE")) ...)*

* Vous voulez un vin blanc pétillant de 2013 de bourgogne pour 5€

*\*BF\* :* *:((GARDE "NON") (MEDAILLE "NON") (NOTE 0) (BIO "NON") (PETILLANT "NON") (ANNEE 2013) (PRIX 5) (GEOGRAPHIE ("BOURGOGNE")) (COULEUR ("BLANC"))*

## Arbre de déduction

# Programmation du Système expert

## Représentation Lisp

## Moteur d’inférences

## Fonctions de services

## Résultats

# Conclusion

Ce TP nous a permis de comprendre l'importante de la recherche et de la structure des données du système. Par ailleurs nous avons réalisé qu'étant donnée l'exhaustivité des cas du monde réel (malgré le fait que nous ayons encadré le sujet juste pour trois régions) les possibilités de vins sont incommensurablement grandes. Un tel système est par conséquent très coûteux à produire et maintenir. Les systèmes experts demeurent donc un bon assistant mais ne peuvent pas remplacer l'homme.

En résumé, ce TP nous a permis d'essayer de traiter un problème réel, complexe et dans apercevoir les limites.