

Vincent Beaudoin (111 103 778)  
Alexandre Picard-Lemieux (111 103 625)  
Gabriel Legault (111 089 063)  
Clément Spies (111 139 346)

Intelligence artificielle I  
IFT-2003

TP #3  
Concevoir un système à base de règles

Travail présenté à  
Laurence Capus

Département d'informatique et de génie logiciel  
Université Laval  
Hiver 2016

## Table des matières

Introduction .....	2
Description du sujet .....	3
Schéma conceptuel .....	3
Explication du problème à résoudre.....	4
Base de connaissances .....	4
Validation .....	6
Bilan de l'expérimentation .....	7
Difficultés rencontrées pour déterminer l'expertise .....	7
Identification d'une autre coquille .....	7
Conclusion .....	8
Bibliographie .....	9

## Introduction

L'intelligence artificielle est un domaine très vaste. Il existe plusieurs manières de résoudre des problèmes. Le système à base de connaissances, aussi connu sous le nom de système expert, est un outil capable de reproduire les mécanismes cognitifs d'un expert, dans un domaine particulier<sup>1</sup>.

Le système à base de connaissances que nous avons pris pour ce travail est un système de voyage simplifié. Ce système se base principalement sur la distance pour choisir son mode de transport.

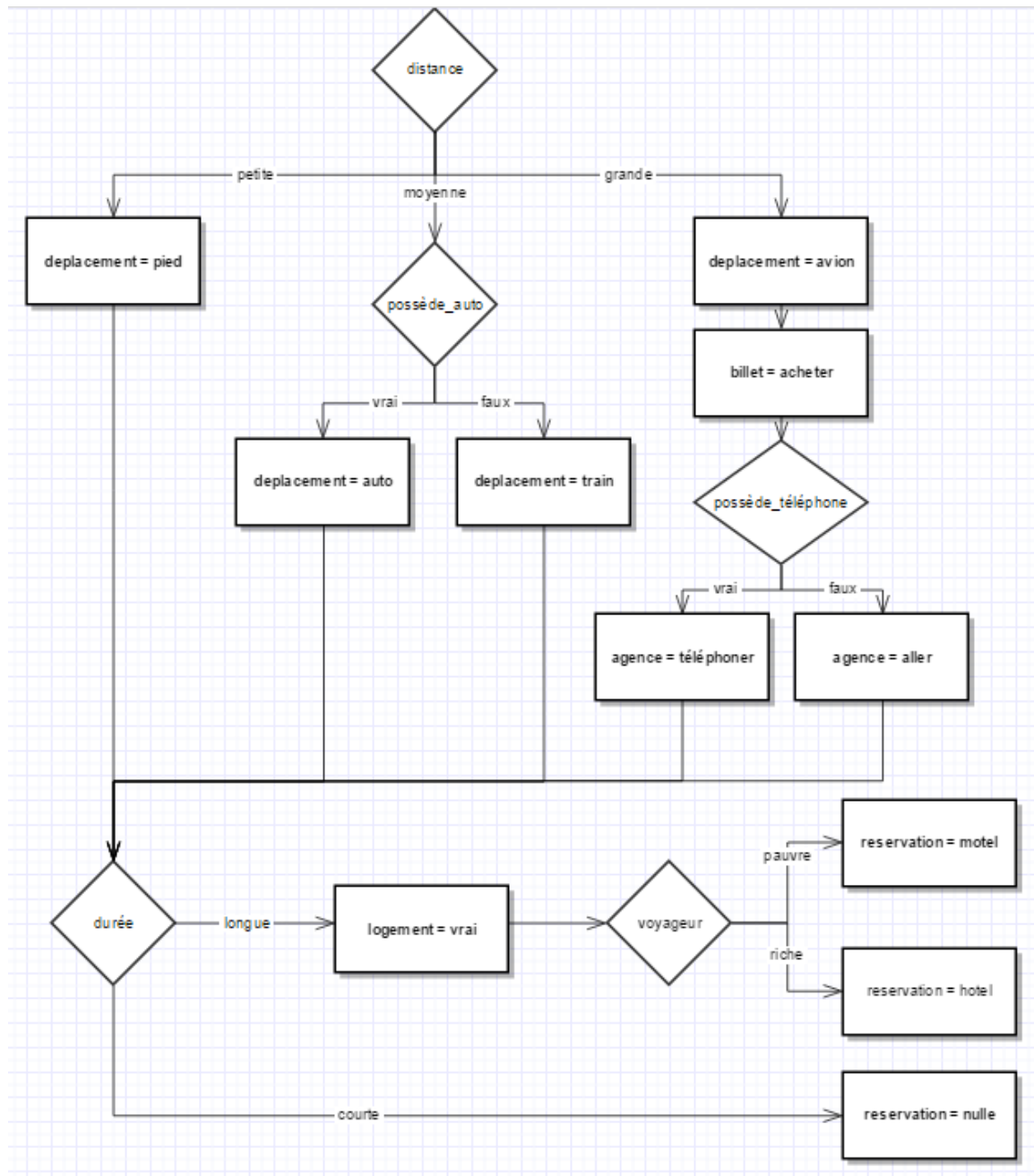
Nous ferons alors une description du sujet pour par la suite développer le système. Ensuite, une validation et un bilan de l'expérimentation seront faits.

---

<sup>1</sup> Wikipédia. [En ligne]. [HTTPS://FR.WIKIPEDIA.ORG/WIKI/SYST%C3%A8ME\\_EXPERT](https://fr.wikipedia.org/wiki/Syst%C3%A8me_expert) (Page consultée le 31 mars 2016)

## Description du sujet

### Schéma conceptuel



## Explication du problème à résoudre

Le problème à résoudre est de définir quels moyens seront utilisés (transport, paiement, logement) par une personne qui voyage d'un point A à un point B en fonction des critères suivants: distance qui sépare les deux points, si la personne est riche ou pauvre, si elle possède une auto, si elle possède le téléphone et la durée du voyage.

Nous nous sommes inspirés de l'exemple qui se trouve dans ce document<sup>2</sup>. Voici les règles :

- R1 : SI distance = petite ALORS déplacement = pied
- R2 : SI distance = moyenne ET possède\_auto = vrai ALORS déplacement = auto
- R3 : SI distance = moyenne ET possède\_auto = faux ALORS déplacement = train
- R4 : SI distance = grande ALORS déplacement = avion
- R5 : SI déplacement = avion ALORS billet = acheter
- R6 : SI billet = acheter ET possède\_téléphone = vrai ALORS agence = téléphoner
- R7 : SI billet = acheter ET possède\_téléphone = faux ALORS agence = aller
- R8 : SI durée = longue ALORS logement = vrai
- R9 : SI durée = courte ALORS logement = faux
- R10 : SI logement = vrai ET voyageur = riche ALORS reservation = hotel
- R11 : SI logement = vrai ET voyageur = pauvre ALORS reservation = motel
- R12 : SI logement = faux ALORS reservation = nulle

## Base de connaissances

Déplacement	Agence	Réservation
Pied	X	Nulle
Pied	X	Hôtel
Pied	X	Motel
Auto	X	Nulle
Auto	X	Hôtel
Auto	X	Motel
Train	X	Nulle
Train	X	Hôtel

---

<sup>2</sup> Laboratoire d'InfoRmatique en Image et Systèmes d'information. [En ligne].

[HTTP://LIRIS.CNRS.FR/AMILLE/ENSEIGNEMENTS/MASTER\\_PRO/BIA/SBC\\_AM.PDF](http://liris.cnrs.fr/amille/enseignements/master_pro/bia/sbc_am.pdf) (Page consultée le 31 mars 2016)

Train	X	Motel
Avion	Téléphoner	Nulle
Avion	Téléphoner	Hôtel
Avion	Téléphoner	Motel
Avion	Aller	Nulle
Avion	Aller	Hôtel
Avion	Aller	Motel

## Validation

Nous avons testé plusieurs cas. Nous avons réinitialisé l'interpréteur Prolog entre chaque test pour s'assurer que la pile soit vide.

Base de faits	Interpréteur Prolog	Explications
fait( distance(petite) ). fait( duree(courte) ).	?-ch_avant. nouveau fait : deplacement(pied) nouveau fait : logement(faux) nouveau fait : reservation(nulle) La BC est saturée true.	Cette commande ajoute de nouveaux faits à la pile.
fait( distance(petite) ). fait( duree(longue) ). fait( voyageur(riche)).	?- ch_arriere(reservation(X)). X = hotel .	Si on veut savoir quel type de réservation serait pertinente avec le type de voyage.
fait( distance(petite) ). fait( duree(longue) ). fait( voyageur(riche)).	?- ch_arriere(deplacement(X)). X = pied .	Si on veut savoir quel type de transport serait pertinent avec le type de voyage.
fait( distance(moyenne) ). fait( possede_auto(vrai) ).	?-ch_arriere(deplacement(X)). X= auto.	Si on veut savoir quel type de transport serait pertinent avec le type de voyage.
fait( distance(moyenne) ). fait( possede_auto(faux) ).	?-ch_arriere(deplacement(X)). X= train.	Si on veut savoir quel type de transport serait pertinent avec le type de voyage.
fait( distance(grande) ). fait( possede_auto(faux) ).	?-ch_arriere(deplacement(X)). X= avion.	Si on veut savoir quel type de transport serait pertinent avec le type de voyage.

## Bilan de l'expérimentation

### Difficultés rencontrées pour déterminer l'expertise

Les difficultés que nous avons rencontrées pour déterminer l'expertise ont été lors de la conception de nos règles représentant la base de connaissances. Nous voulions proposer des règles plus réalistes et à jour dans le domaine d'un système de voyage, mais nous ne sommes pas des spécialistes dans le domaine. Alors nos règles sont assez simplifiées. Cela fait que nos règles ne sont pas vraiment une représentation exacte de ce que le monde réel offre.

De plus, nous avons eu de la difficulté lors de nos validations de notre système à base de règle, car la pile de notre coquille ne s'effaçait pas après avoir exécuté une validation. Cela faussait nos résultats. Nous aurions pu faire un changement à la coquille qui était fournie dans les notes de cours en y ajoutant une fonction pour vider la pile entre chaque chargement du fichier. Cela pourrait éviter les erreurs dues à des règles superflues. Cependant, notre système à base de connaissances était bien adapté pour ce type de structure.

### Identification d'une autre coquille

Nous avons trouvé une coquille sur internet qui se nomme APES<sup>3</sup> qui veut dire « A Prolog Expert System ». Cette coquille est très bien faite, car les opérateurs qui ont été conçus pour faire la grammaire ne sont pas vraiment loin du langage utilisé. Ce qui fait en sorte que les projets peuvent être maintenable par l'expert du domaine sans exiger une connaissance de Prolog ou des compétences en programmation.

Cette coquille aurait pu nous permettre de poser des questions plus complexes en plus de nous permettre d'établir des règles plus complexes. De plus, cette coquille aurait pu nous permettre de ne pas avoir le problème de pile que nous avons vécu. Comme expliquer plus haut, la maintenabilité qu'offre cette coquille aurait pu améliorer notre système en le rendant plus accessible aux experts du domaine pour des améliorations et d'ajout de connaissance.

---

<sup>3</sup> A Prolog Expert System. [En ligne]. [HTTP://APES.SOURCEFORGE.NET/](http://APES.SOURCEFORGE.NET/) (Page consultée le 5 avril 2016)



## Conclusion

En conclusion, notre expérimentation a été un succès. Nous avons expérimenté la conception des systèmes à base de connaissances, utilisé une coquille de système à base de connaissances, analysé les limites de la conception des systèmes à base de connaissances.

Il pourrait être intéressant d'expérimenter avec ce système à l'aide d'autres systèmes à bases de connaissances.

## Bibliographie

Wikipédia. [En ligne]. [HTTPS://FR.WIKIPEDIA.ORG/WIKI/SYST%C3%A8ME\\_EXPERT](https://fr.wikipedia.org/wiki/SYST%C3%A8ME_EXPERT) (Page consultée le 31 mars 2016)

Laboratoire d'InfoRmatique en Image et Systèmes d'information. [En ligne].  
[HTTP://LIRIS.CNRS.FR/AMILLE/ENSEIGNEMENTS/MASTER\\_PRO/BIA/SBC\\_AM.PDF](http://liris.cnrs.fr/amille/enseignements/master_pro/bia/sbc_am.pdf) (Page consultée le 31 mars 2016)

A Prolog Expert System. [En ligne]. [HTTP://APES.SOURCEFORGE.NET/](http://apes.sourceforge.net/) (Page consultée le 5 avril 2016)