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 3](#_Toc447646856)

[Description du sujet 4](#_Toc447646857)

[Schéma conceptuel 4](#_Toc447646858)

[Explication du problème à résoudre 5](#_Toc447646859)

[Base de connaissances 5](#_Toc447646860)

[Validation 6](#_Toc447646861)

[Bilan de l’expérimentation 7](#_Toc447646862)

[Conclusion 8](#_Toc447646863)

[Bibliographie 9](#_Toc447646864)

# Introduction

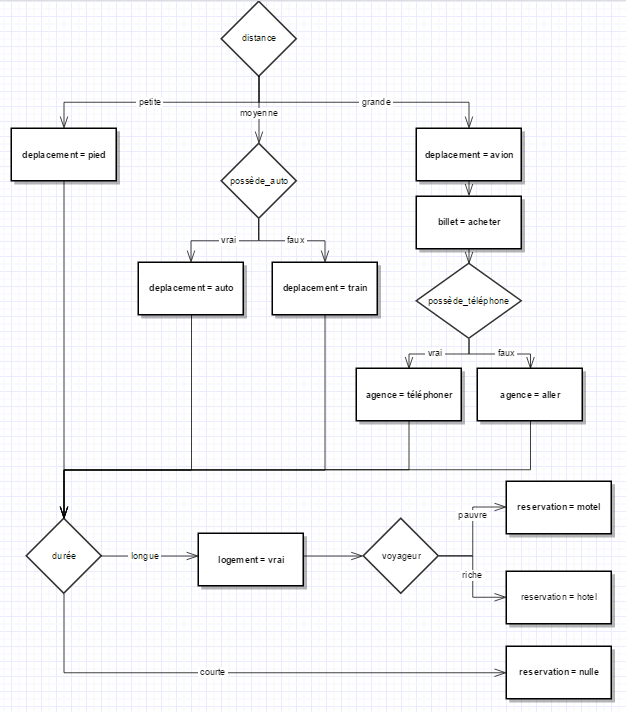
L’intelligence artificielle est un domaine très vaste. Il existe plusieurs manières de résoudre des problèmes. Le système à base de connaissances, aussi connu sous le nom de système expert, est un outil capable de reproduire les mécanismes cognitifs d'un expert, dans un domaine particulier[[1]](#footnote-1).

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 sera faite.

# 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.

## 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 |
| 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’interpréter 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

Nous n’avons pas vraiment eu à changer la coquille qui était fournis dans les notes de cours, car notre projet était bien adapter pour ce type de structure. Nous aurions pu améliorer la coquille en ajoutant une fonction pour vider la pile entre chaque chargement du fichier. Cela pourrait éviter les erreurs du à des règles superflues.

# Conclusion

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

Il pourrait être intéressant d'expérimenter avec ce système à l'aide de 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%25C3%25A8me_expert) (Page consultée le 31 mars 2016)

1. Wikipédia. [En ligne]. [https://fr.wikipedia.org/wiki/Syst%C3%A8me\_expert](https://fr.wikipedia.org/wiki/Syst%25C3%25A8me_expert) (Page consultée le 31 mars 2016) [↑](#footnote-ref-1)