Rapport veille technique

# Sommaire:

[**Notions :**](#_4uchfst4fjga) **3**

[Vocabulaire :](#_sl6guftf71my) 3

[Aide au diagnostic (aide à la décision) :](#_by78pri72je1) 3

[Apprentissage par renforcement :](#_dc9i3wf6h35q) 3

[**Utilisation d’un système expert :**](#_dmn28x72vmso) **5**

[**Partie analyse : qu’allons-nous utiliser ?**](#_5yzi5vwl90wn) **6**

[Langages de programmation possibles/utiles :](#_6r1va77s1imz) 6

[Pistes potentielles :](#_l1n5fecx1jje) 6

[**Conclusion :**](#_uiao44sw8tmi) **9**

[**Bibliographie :**](#_u6cyzntrfjj9) **10**

# 

# Notions :

## Vocabulaire :

Moteur d’inférence : Élément d’un système qui applique des règles sur une base de connaissances pour créer de nouvelles connaissances pertinentes.

Base de connaissances : Comparable à une base de données.

Information: élément de connaissance (voix, donnée, image) susceptible d'être conservé, traité ou transmis.

Système expert : Outil utilisé dans l'intelligence artificielle afin de reproduire le raisonnement d’un être humain.

Moteur de règle : Système logiciel qui permet l’exécution d’une ou plusieurs déclarations de haut niveau structurées qui permettent de contraindre, contrôler et influencer un aspect du [métier](https://fr.wikipedia.org/wiki/M%C3%A9tier_(activit%C3%A9)).

Règle : déclarations de haut niveau structurées qui permettent de contraindre, contrôler et influencer un aspect du [métier](https://fr.wikipedia.org/wiki/M%C3%A9tier_(activit%C3%A9)).

## Aide au diagnostic (aide à la décision) :

Une aide au diagnostic médical est un système d'aide à la décision médicale utilisé dans le processus du diagnostic médical, qui est composé d'une base de données et d'un moteur de recherche. Ce moteur de recherche donne accès aux informations de la base de données, et peut proposer des diagnostics différentiels en fonction de données pré-renseignées après un examen clinique, des estimations pronostiques, ou signaler des informations manquantes pour établir un diagnostic. Il existe plusieurs applications qui font de l’aide au diagnostic (AideDiag, Axilios). Les systèmes d’aide à la décision médicale (SADM) sont des applications informatiques dont le but est de fournir aux cliniciens en temps et lieux utiles les informations décrivant la situation clinique d’un patient ainsi que les connaissances appropriées à cette situation, correctement filtrées et présentées afin d’améliorer la qualité des soins et la santé des patients.

## Apprentissage par renforcement :

En IA, l’apprentissage par renforcement consiste, pour un agent autonome, à apprendre les actions à prendre, à partir d’expériences, de façon à optimiser une récompense quantitative au cours du temps. L'agent est plongé au sein d'un environnement, et prend ses décisions en fonction de son état courant. En retour, l'environnement procure à l'agent une récompense, qui peut être positive ou négative. L'agent cherche, au travers d'expériences itérées, un comportement décisionnel optimal, en ce sens qu'il maximise la somme des récompenses au cours du temps. Plus l’agent sera confronté à diverses expériences, plus il sera efficace dans les décisions qu’il prendra. L’application de cet apprentissage se fait dans plusieurs domaines comme la robotique, la gestion de ressource, le vol d’hélicoptères et même la chimie.

## 

# Utilisation d’un système expert :

Il s’agit d’un outil d’aide à la décision.

À l’aide d’un raisonnement effectué à l’aide de faits et de règles, le logiciel va être capable de répondre à une question donnée. Il va être capable de reproduire le raisonnement d’un être humain. Un système expert est composé de 3 parties : une base de faits, une base de règles et un [moteur d'inférence](https://fr.wikipedia.org/wiki/Moteur_d%27inf%C3%A9rence).

Le moteur d'inférence est capable d'utiliser la base de faits et celle de règles pour produire de nouveaux faits, et va parvenir jusqu'à prévoir la réponse à la question experte posée. L’utilisation de la logique formelle va permettre à ces systèmes d'utiliser un raisonnement déductif. La règle d’inférence principale est : Si le fait P est vrai et que P implique la règle Q alors, Q est vrai (nouveau fait).

Le système expert va utiliser un ensemble de réponses vraies ou fausses afin d’obtenir un résultat. Mais pour obtenir ce résultat, le système expert doit avoir un moteur d’inférence qui applique les règles afin de prévoir des résultats avec les connaissances déjà connues

Il existe 3 méthodes de fonctionnement :

* Le moteur à "chaînage avant" qui cherche à déterminer les informations recherchées en utilisant les faits et les règles qu’il connaît. C’est une méthode de déduction qui applique des règles en partant des prémisses pour en déduire de nouvelle conclusions. Ces conclusions enrichissent la mémoire de travail et peuvent devenir les prémisses d'autres règles. L'un des avantages du chaînage avant sur le chaînage arrière est que la réception de nouvelles données peut déclencher de nouvelles inférences, ce qui rend le moteur plus adapté à des situations dynamiques dans lesquelles les conditions sont susceptibles de changer.
* Le moteur à "chaînage arrière" qui cherche à déterminer les faits en utilisant les règles qu’il connaît et les objectifs qu’il doit obtenir.
* Le moteur à “chaînage mixte" est une combinaison des 2 autres méthodes.

Les données médicales données par le docteur vont donc nous permettre de créer nos règles et les différents critères entrés par le clinicien deviendront nos faits ce qui nous permettra d’orienter les médecins vers la bonne étiologie.

Cette technique va nous permettre de reproduire le comportement d’un être humain à l’aide d’un ensemble de règles et de fait ce qui est utile dans notre projet où le machine learning est inutile dû au faible nombre de données.

# Partie analyse : qu’allons-nous utiliser ?

Comme nous l’avons vu plus haut, l'un des avantages du chaînage avant sur le chaînage arrière est que la réception de nouvelles données peut déclencher de nouvelles inférences, ce qui rend le moteur plus adapté à des situations dynamiques dans lesquelles les conditions sont susceptibles de changer. Dans notre cas, le nombre d’étiologies de l’uvéite sont tellement nombreuses que nous serons obligés d’ajouter des données, ce qui nous dirige à utiliser le moteur de chaînage avant. De plus, le besoin du client est de déduire les étiologies à partir des critères et du type d’uvéite observés, ainsi ce mode de fonctionnement est le principe même du chaînage avant.

## Langages de programmation possibles/utiles :

Pour le frontend, nous utiliserons HTML/CSS qui est nécessaire pour la création du site web et la gestion de la mise en page de ce dernier. Nous allons certainement ajouter du code en JavaScript afin d’ajouter des fonctionnalités dynamiques aux pages du site web

En ce qui concerne le traitement de nos données, il y a plusieurs langages informatiques possibles que nous pourrions utiliser : PHP, Java, Python ou C/C++/C#.

C/C++/C# : aucune connaissance par l’équipe et ne semble pas correspondre aux besoins assez basiques du projet.

PHP : fonctionne très bien avec les base de données mais nous n’en aurons pas si le client trouve ça inutile. Aucune connaissance actuelle de la technologie.

Java : application sur ordinateur mais utilise beaucoup de mémoire. Bonne connaissance de la technologie de la part de l’équipe.

Python : nombreux outils et fonctionnalités qui facilitent la programmation mais plus long pour les scripts compilés. Pourrait nous permettre d’utiliser des fonctions d’intelligence artificielle si le client veut intégrer cette fonctionnalité dans l’application. Assez bonne connaissance de la technologie par l’équipe et peut-être simplement appréhendé.

Après réflexion, nous avons pensé que l’utilisation de Python serait la plus optimale d’une part pour notre projet tuteuré, mais également dans l’optique d’autres améliorations avec par exemple un modèle de Machine Learning qui pourrait être implémenté une fois qu’il y aurait une quantité d’informations suffisantes.

## Pistes potentielles :

sysexpert: Programme développé sur Python permettant la mise en place de système expert et donc de moteur de règle par chaînage avant ou arrière sur une base de connaissances.

Exemple de programmation permise par Sysexpert dans la documentation sur Github:

| Douleur = "Gorge"  Fièvre = Vrai  Sexe = "Homme" |
| --- |
| Appendicite = Vrai := Douleur == "Abdomen" & Vomissements == Faux |
| Intoxication alimentaire = Vrai := Douleur == "Abdomen" & Vomissements == Vrai |
| Dysménorrhée = Vrai := Douleur == "Abdomen" & Sexe == "Femme" |
| Rhume = Vrai := Douleur == "Gorge" & Fièvre == Vrai |
| Infarctus = Vrai := Douleur == "Poitrine" |
| Mal de gorge = Vrai := Douleur == "Gorge" & Fièvre == Faux |
| Rhume = Vrai := Douleur == "Aucun" & Toux == Vrai & Fièvre == Vrai |
| Refroidissement = Vrai := Douleur == "Aucun" & Toux == Vrai & Fièvre == Faux |

<https://github.com/theodeze/SystemeExpert/blob/master/docs/maladies.txt>

Cet exemple illustre assez bien ce que nous cherchons et semble illustrer notre besoin, le seul problème pourrait être le manque d’activité autour de ce programme, il n’a reçu que deux versions en deux jours d’intervalle en Novembre 2018 et peut avoir été entamé mais laissé à l’abandon, c’est pour cela qu’il nous faut explorer d’autres possibilités au cas où cette solution soit incomplète.

Prolog:

Langage de programmation qui permet de créer des règles logiques pouvant être utilisé pour créer des règles métiers. Ce langage peut être implémenté grâce à Pyke que nous développerons plus tard.

Voici un exemple de code issu du langage Prolog:

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Les faits de base sont père, homme et mari. \*/

pere(pat,elsa).

pere(pat,celine).

homme(pat).

mari(pat,christine).

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* règles \*/

/\* toute personne qui apparait au dessus et qui n'est pas un homme est une femme

\* soit elle est mariée, soit elle est enfant de quelqu'un. \*/

femme(F) :- mari(\_,F).

femme(F) :- pere(\_,F), not(homme(F)), not(mari(\_,F)). /\* le dernier pour éviter les doublons ! \*/

mere(M,E) :- pere(P,E),mari(P,M).

mere(X) :- mere(X,\_). /\* prévoyons également les prédicats d'arité 1,

c'est à dire à un seul argument : X est une mère si elle a des enfants \*/

pere(X) :- pere(X,\_).

enfant(E,P) :- pere(P,E).

enfant(E,M) :- mere(M,E).

enfant(X) :- enfant(X,\_).

/\* là, c'est une définition récurente \*/

descendant(D,A) :- enfant(D,A).

descendant(D,A) :- enfant(D,X),descendant(X,A).

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* pour les tests, il suffit de demander "aff." \*/

aff :- homme(X),write('Homme : '),write(X),nl,fail.

aff :- femme(X),write('Femme : '),write(X),nl,fail.

aff :- pere(X), write('Père : '), write(X),nl,fail.

aff :- mere(X), write('Mère : '), write(X),nl,fail.

aff :- enfant(X),write('Enfant : '),write(X),nl,fail.

aff :- descendant(D,A),write(D),write(' descend de '),write(A),nl,fail.

Pyke:

Pyke est un programme également développé en Python permettant de créer des moteurs d’inférences à partir du langage cité ci-dessus. Pyke possède sa dernière mise à jour en 2013 et peut ne plus être adapté aux besoins actuels ou être devenu obsolète ou du moins, avoir perdu en efficacité.

# 

# Conclusion :

Actuellement, Sysexpert et Pyke semblent adaptés à nos besoins, ils ont été développés sur le langage de programmation que l’on souhaite, à savoir Python. De plus, ils nous permettent de faire des moteurs de règles à chaînage avant ou arrière. Cependant, l’ancienneté de Pyke ainsi que son absence de nouveautés peuvent laisser craindre que ce ne soit pas adapté à nos besoins.

Nous continuerons nos recherches sur les différents outils de système expert afin de vérifier s’il n’en existe pas de plus adapté.

# 

# Bibliographie :

[Moteur d'inférence: Définition, Exemples, Fonctionnement](https://24pm.com/117-definitions/387-moteur-d-inference)

[Dompter le domaine grâce aux moteurs de règles](https://blog.ippon.fr/2020/02/25/dompter-le-domaine-grace-aux-moteurs-de-regles/)

[Système expert : définition, fonctionnement et exemples](https://www.journaldunet.fr/web-tech/guide-de-l-intelligence-artificielle/1501897-systeme-expert-definition-fonctionnement-et-exemples/)

<https://www.youtube.com/watch?v=yxbAnyWL0SM>

[theodeze/SystemeExpert: Système Expert / Intelligence Artificielle / Master 1 Info](https://github.com/theodeze/SystemeExpert)

[Welcome to Pyke](http://pyke.sourceforge.net)

<http://ptrau.free.fr/program/prolog/genealogie.htm>