

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTRE DE L'ENSEIGNEMENT SUPERIEUR
ET DE LA RECHERCHE SCIENTIFIQUE



UNIVERSITE SAAD DAHLAB DE BLIDA
FACULTE DES SCIENCES
DEPARTEMENT D'INFORMATIQUE

MEMOIRE DE FIN D'ETUDES

Pour l'obtention d'un
Diplôme de Master en Informatique

Option :

-Ingénierie des Logiciels
-Système informatique et réseaux

THÈME :

**Conception et réalisation d'un système
d'aide au diagnostic automobile**

Présenté par :

Mr. BOUDIBA Abderraouf

Mr. BERKANI Kamel

Membres du jury :

Mme. BACHA Sihem

Mme. GHEBGHOUB Yasmine

Encadré par :

Mr. Cherif Zahar Amine

Présidente

Examinatrice

Promotion 2019/2020

Remerciements

Tout d'abord, nous remercions notre Dieu le tout puissant de nous avoir donné la force d'atteindre notre objectif et accomplir notre travail.

*Nous voudrions dans un premier temps remercier, notre encadreur **Mr. Cherif Zahar Amine** pour sa patience, sa disponibilité et surtout ses judicieux conseils, qui nous ont aidé à bien mener ce mémoire.*

Nous remercions également Mme Bacha S. qui a accepté de présider et d'honorer de sa présence le jury de soutenance du présent mémoire de Master. Qu'elle trouve ici l'expression de notre reconnaissance et de notre profond respect.

Nos remerciements vont également à Mme Ghebghoub pour nous avoir fait l'honneur en participant à l'évaluation de ce travail.

Enfin, nous remercions aussi nos parents et nos amis qu'on n'a pas cités et qui ont toujours été là pour nous. Leur soutien inconditionnel et leurs encouragements nous ont été d'une grande aide.

Dédicaces

J *e dédie ce fruit de mes longues années d'études tout d'abord :*

A *mes très chers parents, qui sont la lumière de ma vie, qui ont tant souffré et sacrifiés pour que je sois heureux, pour leurs conseils, leur affection et leurs encouragements.*

J *e vous remercie pour tous vos efforts fournis pour moi, que Dieu vous garde, vous protège, et vous bénisse la vie.*

Et je le dédie :

A *tous les membres de ma famille, mon frère « Sofiane », ma sœur « Sofia ». A toutes mes proches*

A *tous mes amis et mes collègues avec qui*

J'ai partagé de très bons moments tout le long de ces années.

A *tous les enseignants et les collègues de la promotion de la deuxième année Master 2019/2020*

Boudiba

Abderraouf

Résumé :

Le présent travail a pour but de fournir un outil d'aide à chaque automobiliste lui permettant de s'auto-dépanner en cas de panne de sa voiture.

Pour faire ce travail, nous avons passé en revue les différentes approches de la représentation des connaissances, afin de pouvoir faire une évaluation concrète de leurs points forts et de leurs défauts et pouvoir ainsi adopter la meilleure approche pour notre travail.

Notre choix à porter sur l'ontologie. Depuis quelques années, l'idée de la représentation des connaissances par l'ontologie a émergée comme solution potentielle à la résolution de plusieurs problèmes en informatique, particulièrement en relation avec les systèmes experts.

Nous avons cherché quelles devaient être les étapes d'une méthode permettant l'implémentation d'un système centré sur l'ontologie et nous sommes arrivés au plan pratique suivant :

- Construire une ontologie
- Construire une interface graphique à l'aide d'un langage de programmation
- Relier l'interface graphique avec l'ontologie par des requêtes en utilisant SPARQL

Nous avons ensuite évalué notre travail et nous avons suggéré des améliorations en perspectives.

Mots clé; ontologie, problème, connaissance, automobile, diagnostic, application

Abstract :

The purpose of this work is to provide a tool to help each motorist to help themselves in the event of a car breakdown.

To do this work, we reviewed the different approaches to knowledge representation, in order to be able to make a concrete assessment of their strengths and weaknesses and thus be able to adopt the best approach for our work.

Our choice focused on ontology. In recent years, the idea of knowledge representation through ontology has emerged as a potential solution to several problems in computer science, particularly in relation to expert systems.

We looked for the steps of a method that would allow the implementation of an ontology-centered system and came up with the following practical plan:

- Building an ontology
- Build a graphical user interface using a programming language
- Link the GUI with the ontology through queries using SPARQL

We then evaluated our work and suggested improvements in the future

المخلص

والغرض من هذا العمل هو توفير أداة مساعدة لكل سائق للمساعدة الذاتية في حالة تعطل السيارة ولإنجاز هذا العمل، قمنا باستعراض النهج المختلفة لتمثيل المعرفة، حتى نتمكن من إجراء تقييم ملموس لمواطن قوتها وأوجه قصورها، وبالتالي نكون قادرين على اعتماد أفضل نهج لعملنا وفي السنوات الأخيرة، برزت فكرة تمثيل المعارف عن طريق تبويب المعارف كحل. واختيارنا هو تبويب المعارف محتمل لحل عدة مشاكل حاسوبية، لا سيما فيما يتعلق بنظم الخبراء

وقد نظرنا في الخطوات التي ينبغي اتخاذها في سبيل تنفيذ نظام يستند إلى نظام قائمة على نظام الأحافير، وقد وضعت

الخطوة العملية التالية

بناء نظام تبويب المعارف

قم بإنشاء واجهة رسومية باستخدام لغة برمجة

SPARQL ربط واجهة المستخدم الرسومية بعلم تبويب المعلومات بالاستعلامات باستخدام

ثم قمنا بتقييم عملنا واقترحنا فرص التحسين

Table des matières

1-Introduction générale :.....	1
2-Problématique :	1
3-Objectif	3
Chapitre I : Etat de l’art (système questions-réponses et méthodes de Représentation des connaissances) :.....	5
1-Introduction :.....	5
2-Type de système questions-réponses :	5
3-Les problèmes liés à la représentation des connaissances	7
4-Les techniques de représentation des connaissances :	9
4-1-Réseau sémantique :.....	9
4-2-Le calcul des prédicats	11
4-3-les graphes conceptuels :.....	13
4-4-les systèmes de schémas (frames) :.....	15
4-5-les scripts :.....	18
4-6-Réseaux bayésiens :	18
4-7-Logique de description :.....	20
4-8-Les ontologies :	23
5-Analyse et comparaison :	31
Chapitre II : Conception d’une application d’aide au diagnostic automobile	40
Chapitre III : Implémentation :	45
Conclusion générale :	59

Table des figures :

Figure 1:Garfield est un chat [Tourigny et Capus, 2001]	9
Figure 2:Le bébé dort avec une boucle et sans boucle [Tourigny et Capus, 2001]	9
Figure 3:relation le plus petit [Tourigny et Capus, 2001]	9
Figure 4:relation est_un [Tourigny et Capus, 2001]	10
Figure 5 :unification un seul arc [Tourigny et Capus, 2001]	10
Figure 6:séparation[Tourigny et Capus, 2001]	10
Figure 7:la non existence d'héritage [Tourigny et Capus, 2001]	10
Figure 8:la non existence d'une relation [Tourigny et Capus, 2001]	11
Figure 9:la forme graphique [Tourigny et Capus, 2001]	13
Figure 10:application de sowa [Tourigny et Capus, 2001]	14
Figure 11: Exemple d'une représentation des propositions ou des situations [Tourigny et Capus, 2001]	14
Figure 12:classes et relations est_un et sorte_de [Tourigny et Capus, 2001]	15
Figure 13: Les données de description de ce système [Tourigny et Capus, 2001]	16
Figure 14:conflits d'héritage [Tourigny et Capus, 2001]	17
Figure 15: le critère de proximité inférentielle [Tourigny et Capus, 2001]	17
Figure 16: Réseau bayésien [Emmanuel ADAM]	18
Figure 17:Inférence - héritage et instances [Julien Velcin ,2019]	22
Figure 18:A-box[Julien Velcin ,2019] Figure 19:T-box [Julien Velcin ,2019]	23
Figure 20:ontologie-représentation de l'exemple [Julien Velcin ,2019]	25
Figure 21:triplet RDF [Michel Tétréault,2011]	27
Figure 22:table RDF [Michel Tétréault,2011]	27
Figure 23:graphe élémentaire des deux triplets [Michel Tétréault,2011]	27
Figure 24:relation de typification [Michel Tétréault,2011]	28
Figure 25:RDF classe [Michel Tétréault,2011]	28
Figure 26:RDF sous-classes [Michel Tétréault,2011]	29
Figure 27:RDF subPropertyOf/domain et rang [Michel Tétréault,2011]	29
Figure 28 :ontologie d'un réseau informatique [simon malenfant-corriveau,2017]	36
Figure 29: ontologie d'un système de détection d'intrusion [simon malenfant-corriveau,2017]	36
Figure 30:combinaison des ontologie réseau et SDI [simon malenfant-corriveau,2017]	37
Figure 31: diagramme cas d'utilisation	41
Figure 32 : Diagramme de séquence	42
Figure 33: Diagramme de classes	43
Figure 34: Graphe représentant les classes de l'ontologie et le nombre d'instances pour chaque classe	50
Figure 35: graphe représentant l'ontologie utilisant OntoGraf	51
Figure 36:les causes	51
Figure 37:les symptômes	52
Figure 38: les questions	52
Figure 39: classe erreur	53
Figure 40: les solutions	53
Figure 41: première fenêtre	55
Figure 42 : première question	56
Figure 43 : fenêtre symptôme avec question	57
Figure 44:résultat du quiz	58

Liste des tableaux :

Tableau 1: la notion de phrase dans le calcul des prédicats [Tourigny et Capus, 2001]	12
Tableau 2: règles d'interférence [Tourigny et Capus, 2001]	12
Tableau 3: type des référents [Tourigny et Capus, 2001]	13
Tableau 4: Représentation textuelle et convention d'écriture [Tourigny et Capus, 2001]	14
Tableau 5: table de pluie [Emmanuel ADAM]	19
Tableau 6: relation MB-pluie [Emmanuel ADAM]	19
Tableau 7: table de P(MA/A, Pluie) [Emmanuel ADAM]	19
Tableau 8: syntaxe de la logique de description [Nhan le Thanh]	21
Tableau 9: langage FLE [Nhan le Thanh]	21
Tableau 10: SHOIN et SHIQ [Nhan le Thanh]	22
Tableau 11: comparaison entre le réseau sémantique et l'ontologie [Abdel-badeeh m. Salem, Marco Alfonso, 2008]	32
Tableau 12: Comparaison entre les techniques de représentation [Laura Uusitalo. (2007), MARTIN F.R. (2019), Abdel-Badeeh M. Salem, Marco Alfonso. (2008), Jean-Gabriel Ganascia]	35
Tableau 13: Exemple des pannes moteur présentées dans le livre [A.BENZADI & D.BOURAÏ]	50

Introduction générale

Introduction générale

Introduction générale :

1-Introduction :

On définit la connaissance ou les connaissances comme « ce qu'on a appris par l'étude » ou « par la pratique. »

Dans le but de résoudre des problèmes complexes qui relèvent de l'intelligence artificielle, il faut un bon bagage de connaissances et des outils de manipulation de ces connaissances. Les connaissances concernent des faits, considérés vrais dans un certain monde. Pour représenter ces faits on a recours à un formalisme ou mode de représentation. Au niveau des faits, on traitera des objets et des relations qu'ils entretiennent. Au niveau du formalisme, on définira des symboles et des opérations sur ces symboles.

Le langage naturel est un outil très précis pour représenter les connaissances mais difficile à manipuler en informatique. L'arithmétique et la logique, plus formels, sont aussi des outils de représentation. Un bon choix de représentation peut être déterminant pour la résolution d'un problème. On commence par identifier les objets pertinents et connaître la position de chaque objet qu'on appellera « état. »

Une représentation se compose :

- d'une partie lexicale « symboles pour les objets et relations » ;
- d'une partie structurelle « les contraintes sur ces symboles » ;
- d'une partie procédurale « création et modification de l'information » ;
- d'une partie sémantique : « associer un sens aux descriptions formelles » [Tourigny et Capus, 2001]

2-Problématique :

Sur les sites dédiés à la mécanique automobile, on retrouve un certain nombre de pannes. Par exemple, si vous signalez que vous avez une perte de vitesse accompagnée d'une fumée blanchâtre, vous aurez un immense développement sur les causes probables : vanne EGR encrassée, débitmètre hors service, capteur de pression atmosphérique calaminé etc. Chacun de ces symptômes possibles est également conjugué à d'autres symptômes.

Introduction générale

Le problème des ennuis mécaniques et que souvent la facture est salée et nombreux sont les propriétaires de véhicules qui cherchent eux même à connaître la panne pour y remédier d'autant que souvent aussi, la panne se résume à un dépôt de saletés quelque part.

En matière de diagnostic automobile, les propriétaires de véhicules ont pu bénéficier des nouvelles technologies embarquées par les constructeurs automobiles sur les véhicules modernes.

Le système informatique d'un véhicule génère des DTC (diagnostic troubles codes) standardisés, ces DTC peuvent être lu par des scanner OBD (on-board diagnostic) qui collectent les informations transmises par les calculateurs électroniques du véhicule et informent sur l'état du véhicule, permettant ainsi à des utilisateurs s'y connaissant un peu en mécanique d'utiliser ces interfaces afin de réaliser certaines opérations d'entretien sur leur véhicule.

Ces scanner OBD se révèlent de très bons outils pour diagnostiquer un véhicule, cependant ils restent très peu accessibles pour des novices, et ne proposent pas de solutions de réparation.

Une autre approche au diagnostic automobile a été adopté par une équipe de l'Institut de recherche en informatique de Toulouse, qui a développé un logiciel de recherche d'information sémantique pour le diagnostic automobile.

Cette solution utilise les bases d'incidents connus. Ces rapports de pannes suivent généralement une structure bien définie et sont écrits en langue naturelle, ce qui ne permet pas de récupérer facilement l'information recherchée. (Reymonet et al., 2017)

Contrairement aux moteurs de recherche fondés sur des techniques statistiques qui manquent de précision, cette solution exploite le contenu sémantique des bases d'incidents, grâce à l'indexation sémantique qui établit un lien entre les chaînes de caractères et le sens qu'elles dénotent par l'utilisation du concept d'ontologie qui représente le domaine du diagnostic automobile et de sa terminologie pour prendre en compte toutes les manifestations linguistiques d'un terme.

Introduction générale

3-Objectif :

Le projet consiste à développer une application mobile qui permet à un utilisateur de trouver les causes pour une panne qui apparaît sur son véhicule et leurs solutions, d'une façon simple, juste par des questions / réponses, guidé jusqu'au bout par l'application.

Chapitre I :

Etat de l'art

Chapitre I : Etat de l'art (système questions-réponses et méthodes de Représentation des connaissances) :

1-Introduction :

La représentation des connaissances constitue un concept majeur de nos jours.

Plusieurs façons de représenter les connaissances permettent de mieux exploiter les outils de traitement de ces connaissances. Il est donc nécessaire de voir les différentes techniques utilisées, en informatique, pour la représentation des connaissances.

Avant de présenter les différentes techniques de représentation de connaissances, nous allons brièvement, présenter les différents types de systèmes de questions-réponses.

2-Type de système questions-réponses :

2-1-Types de système de questions-réponses selon le domaine :

Domaine fermé :

Un système de question réponse à domaine ouvert est un système qui permet à des experts ou à des non experts d'interroger un corpus d'un domaine spécialisé, dans le but d'obtenir des réponses à des questions qui relèvent du domaine d'application traité par le système.

Ce type de systèmes nécessite des ressources spécialisées dépendantes du domaine d'application telles que des lexiques ou des ontologies.

Domaine ouvert :

Un système de question à domaine ouvert est un système qui permet à ses utilisateurs de poser des questions plus génériques, qui ne sont pas restreintes à un domaine d'expertise particulier.

Ce type de systèmes dépend de ressources généralistes telles que Wikidata et DBpedia, deux projets qui ont pour objectif d'exploiter les connaissances de l'encyclopédie Wikipedia.

Chapitre I : Etat de l'art

2-2-Type de systèmes selon le type d'entrée (input) :

Entrée libre :

Le système accepte n'importe quel type d'entrée, d'un simple mot clé à une phrase complète, ce qui permet d'avoir un système simple d'utilisation qui ne nécessite pas de formation pour l'utilisateur.

Entrée semi-libre :

Le système n'accepte que des entrées formulées d'une certaine façon, ce qui nécessite une formation pour la bonne formulation des requêtes des utilisateurs.

Entrée contrôlée :

Le système guide l'utilisateur, en ne lui proposant qu'un ensemble d'entrées possibles.
[researchgate]

2-3-Type de systèmes selon le paradigme implémenté :

Recherche d'information (IR) :

Système qui utilise des moteurs de recherche sur des corpus pour répondre aux questions, et qui applique des méthodes de filtre et de classement sur les réponses potentielles.

Traitement automatique des langues (NLP) :

Système qui se sert des techniques de traitement des langues naturelles pour extraire des réponses à partir de corpus documentaire.

Quelques techniques qui peuvent être utilisées : reconnaissance d'entités nommées, étiquetage morpho-syntaxique (pos tagging), extraction terminologique.

Base de connaissances :

Système qui a recours à des sources de données structurées (base de connaissances), on interroge ces bases de connaissances en utilisant des langages de requête tel que SPARQL.

Chapitre I : Etat de l'art

Hybride :

Les systèmes hybrides prennent avantage de toutes ces techniques, cette approche permet une combinaison des techniques d'IR, NLP, base de connaissances. [sciencedirect]

3-Les problèmes liés à la représentation des connaissances :

Problème 1.

L'exception.

On ne peut pas généraliser des lois sur des objets du monde réel de même nature s'il existe quelques exceptions. Par exemple, les pays d'Europe ne sont pas tous inclus dans l'union européenne.

Problème 2.

L'univers en évolution.

Dans un SBC, des informations sont créées, inférées, modifiées ou détruites. Certaines sont en relation avec de nouvelles informations ; il s'en suit donc la mise à jour des connaissances rendues périmées par ces modifications. [Christine Jouve,1992]

Problème 3.

Modalité des connaissances.

Toutes les connaissances n'ont pas le même statut. Un autre critère est la possibilité de modification des connaissances. Leur statut peut alors être soit intangible, soit modifiable. Un troisième type de statut repose sur le mode d'acquisition de la connaissance, soit c'est une donnée, soit elle a été déduite par inférence. Certaines connaissances sont certaines (des relations, des concepts, des théorèmes et des algorithmes de résolution). D'autres sont incertaines (Ce sont des heuristiques, des stratégies utilisées par les experts humains, l'expression d'un point de vue). La connaissance du statut est nécessaire ; on ne peut utiliser la même stratégie dans le cas d'informations certaines ou dans le cas d'incertitude. [Christine Jouve,1992]

Chapitre I : Etat de l'art

Problème 4.

Conservation des ambiguïtés.

Pour représenter la réalité il faut collecter des informations précises afin de créer un système efficace. Ce système doit faire son raisonnement sans besoin d'autres informations. Le but est de réduire la plage d'information.

Problème 5.

Informations incomplètes, incertaines ou implicites.

Même si certaines informations sont incomplètes, le système doit être capable de raisonner et de fournir des résultats incertains, pondérés par des estimations de l'incertitude

Problème 6.

Quantificateurs.

Les quantificateurs reconnus par la logique classique ne suffisent pas pour représenter tout type de connaissance.

Dans le cas où on veut parler de la 'majorité', la 'plupart', 'généralement', 'souvent'.

[Christine Jouve,1992]

Problème 7.

Connaissances temporelles et spatiales.

Quand on veut représenter des connaissances temporelles ou spatiales, il est difficile de fixer le degré de finesse que ce soit pour une unité de temps ou pour des coordonnées.

Qu'est-ce qu'un lieu ? :

Une ville ? une place ? des coordonnées géographiques ?

Chapitre I : Etat de l'art

4-Les techniques de représentation des connaissances :

4-1-Réseau sémantique :

un réseau sémantique comprend des nœuds avec plusieurs formes, il permet de représenter des objets. Des liens orientés ou des arcs représentent les liens entre ses objets. Il est possible de représenter des relations unaires en utilisant des boucles.

Exemple : Garfield est un chat

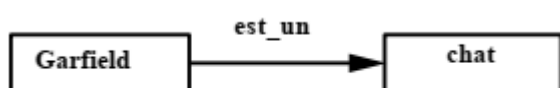


Figure 1:Garfield est un chat [Tourigny et Capus, 2001]

Exemple : Le bébé dort avec une boucle et sans boucle. (Plusieurs façons possibles de représenter la même information)



Figure 2:Le bébé dort avec une boucle et sans boucle [Tourigny et Capus, 2001]

On peut représenter des discours, des paragraphes ou des phrases avec le réseau sémantique.

Énoncé conditionnel :

Représenter les relations en pointillé.

Exemple :

Si $a < b$ et $b < c$ alors $a < c$

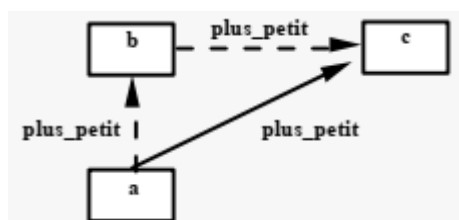


Figure 3:relation le plus petit [Tourigny et Capus, 2001]

Chapitre I : Etat de l'art

Si Napoléon était empereur et Joséphine était son épouse alors Joséphine serait impératrice

Est_un :

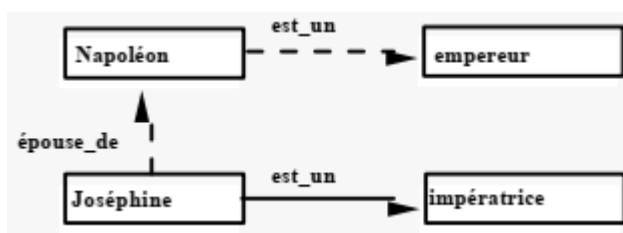


Figure 4:relation est_un [Tourigny et Capus, 2001]

Unification (assortiment de patrons ou filtrage) :



Figure 5 :unification un seul arc [Tourigny et Capus, 2001]

Selon la première figure les deux arcs sont unifiables il y a la possibilité d'utilisé un seul arc par contre dans la deuxième figure ce n'est pas le cas.

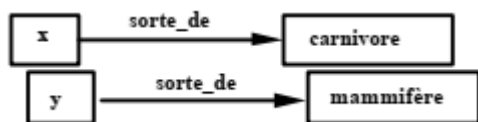


Figure 6:séparation[Tourigny et Capus, 2001]

Héritage et propriété collective :

Il existe des cas où la classe n'hérite pas de sa superclasse, la représentation est comme suit :

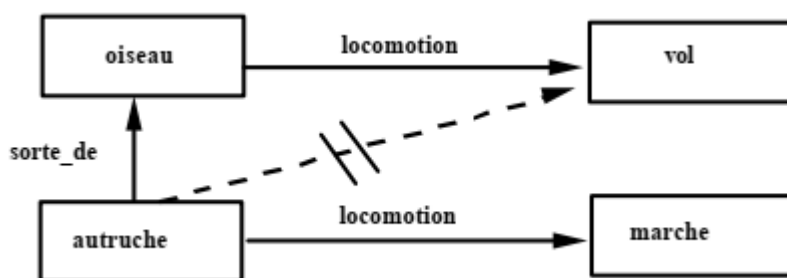


Figure 7:la non existence d'héritage [Tourigny et Capus, 2001]

Chapitre I : Etat de l'art

Non-existence d'une relation :

Falsum est noté graphiquement \otimes

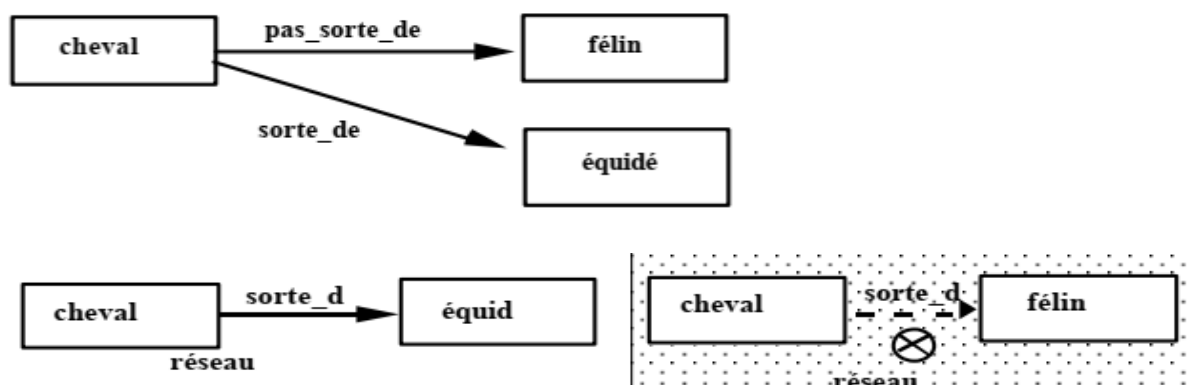


Figure 8: la non existence d'une relation [Tourigny et Capus, 2001]

4-2-Le calcul des prédicats :

Le calcul des prédicats est un moyen pour la représentation des connaissances. Il peut exprimer des relations avec des symboles pour représenter des ensembles, de l'arithmétique, du calcul relationnel et de la logique mathématique.

Les suites de symboles s'appellent des **phrases** dans le calcul des prédicats.

Une phrase est satisfaite si elle correspond à un fait vrai dans ce monde par interprétation.

Une phrase est **valide** si toutes les interprétations sont satisfaites. Dans le sens contraire la phrase est inconsistante.

Il existe des phrases qui sont désignées comme étant vraies, ce sont des **axiomes**.

A partir d'hypothèses on obtient des conclusions, c'est le processus d'inférence.

Les **théorèmes** sont des phrases obtenues à partir des axiomes.

phrase ::= phrase_atomique phrase_logique phrase_quantifiée
phrase_atomique ::= nom_de_prédicat { (liste_de_termes) } (terme opérateur_de_prédicat terme)
phrase_logique ::= (\emptyset phrase) (phrase $\dot{\cup}$ phrase) (phrase $\dot{\cup}$ phrase)
phrase_quantifiée ::= ("x: [phrase]) (\$x: [phrase])
liste_de_termes ::= terme terme, liste_de_termes
terme ::= constante variable expression fonctionnelle

Chapitre I : Etat de l'art

constante ::= nombre nom_de_constante
variable ::= nom_de_variable
expression fonctionnelle ::= nom_de_fonction { (liste_de_termes) } (terme opérateur_de_fonction terme) (opérateur_de_fonction terme)
opérateur_de_fonction ::= + - * / Ç È C (complément)
opérateur_de_prédicat ::= = ¹ < > £ ³ Í Ê Ì É Î

Tableau 1: la notion de phrase dans le calcul des prédicats [Tourigny et Capus, 2001]

Prémises du schéma	Conclusion du schéma	Aussi connu comme	
1. Non non P	P	Double négation	Schémas Noyaux
2. Si P ou Q alors R; P	R	Modus Ponens disjonctif	
3. P ou Q; non P	Q	Élimination de la disjonction	
4. Pas à la fois P et Q; P	non-Q	Conjonction niée	
5. P ou Q; Si P alors R; Si Q alors R	R		
6. P ou Q; Si P alors R; Si Q alors S	R ou S		
7. Si P alors Q; P	Q	Modus Ponens	
8. P; Q	P et Q	Introduction de la Conjonction	Schémas auxiliaires
9. P et Q	P	Élimination de la conjonction	
10. P et (Q ou R)	(P et Q) ou (P et R)		
11. P et non-P	Incompatibilité	Contradiction	
12. P ou Q; Non P; Non Q	Incompatibilité	Contradiction	

Tableau 2: règles d'interférence [Tourigny et Capus, 2001]

Chapitre I : Etat de l'art

4-3-les graphes conceptuels :

Ce mode de représentation de connaissances a été défini par Sowa (1984) pour traduire la logique sous forme graphique. L'idée de base consiste à représenter les connaissances par les nœuds et les flèches d'un graphe. [Tourigny et Capus, 2001]

Les types de nœuds :

- nœud conceptuel : un rectangle représente une entité.
- nœud relationnel : une ellipse représente une propriété ou une relation.

On peut les noter sous forme graphique ou textuelle (forme linéaire).

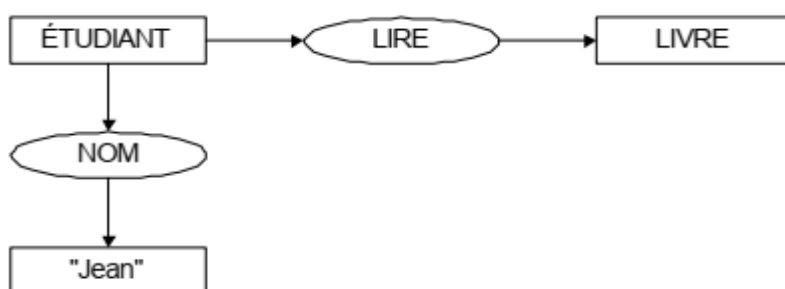


Figure 9:la forme graphique [Tourigny et Capus, 2001]

- Représentation textuelle : [« Jean »] <- (NOM) <- [ÉTUDIANT] -> (LIRE) -> [LIVRE].

Référent	Représentation	Interprétation
universel	LIVRE : "	tous les livres
générique	LIVRE : * LIVRE	un livre (ce référent est implicite)
particulier	LIVRE : #45	le livre
singulier	LIVRE : 'Les misérables'	un livre intitulé 'Les misérables'
ensemble générique	LIVRE : {*}	des livres
ensemble particulier	LIVRE : {#45, #89}	les livres
ensemble singulier	LIVRE : {'Les misérables', 'Les trois mousquetaires'}	un livre intitulé 'Les misérables' et un livre intitulé 'Les trois mousquetaires'
mesure générique	Température : @20°	une température de 20 degrés
mesure particulière	Longueur : #67 @30m	la longueur de 30 mètres
cardinalité d'un ensemble	LIVRE : {*} @2	deux livres

Tableau 3:type des référents [Tourigny et Capus, 2001]

Chapitre I : Etat de l'art

L'exemple précédent d'après Sowa (1984) devient :

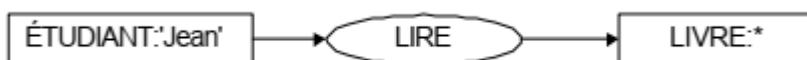


Figure 10:application de sowa [Tourigny et Capus, 2001]

Représentation textuelle et convention d'écriture

<i>Représentation graphique</i>	<i>Représentation textuelle</i>
concept	[TYPE : référent]
relation	(TYPE)
flèche	® ou \neg

Tableau 4: Représentation textuelle et convention d'écriture [Tourigny et Capus, 2001]

Les contextes et les liens de coréférence :

Représenter des propositions ou des situations.

Exemple : contexte SITUATION : une personne raconte son voyage.

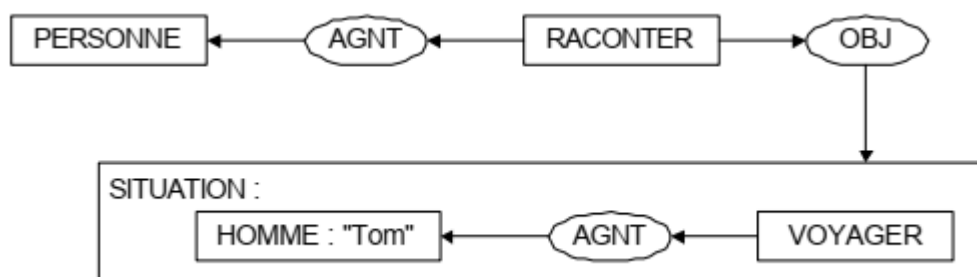


Figure 11: Exemple d'une représentation des propositions ou des situations [Tourigny et Capus, 2001]

Chapitre I : Etat de l'art

4-4-les systèmes de schémas (frames) :

Le réseau sémantique identifié avec un nœud accompagné des attributs et leurs valeurs est un **schéma** dans le cas où seuls les liens d'héritage (relations **d'appartenances**) sont exprimés.

Types de Frames :

Les frames « prototypes » :

Représentent une classe d'objets.

Décrivent le contexte de cette classe.

les frames « instances » :

Réalisations particulières d'une classe donnée.

Description des individus d'une classe.

Exemple :

personne -> classe : frame prototype

Sam -> individu : frame instance

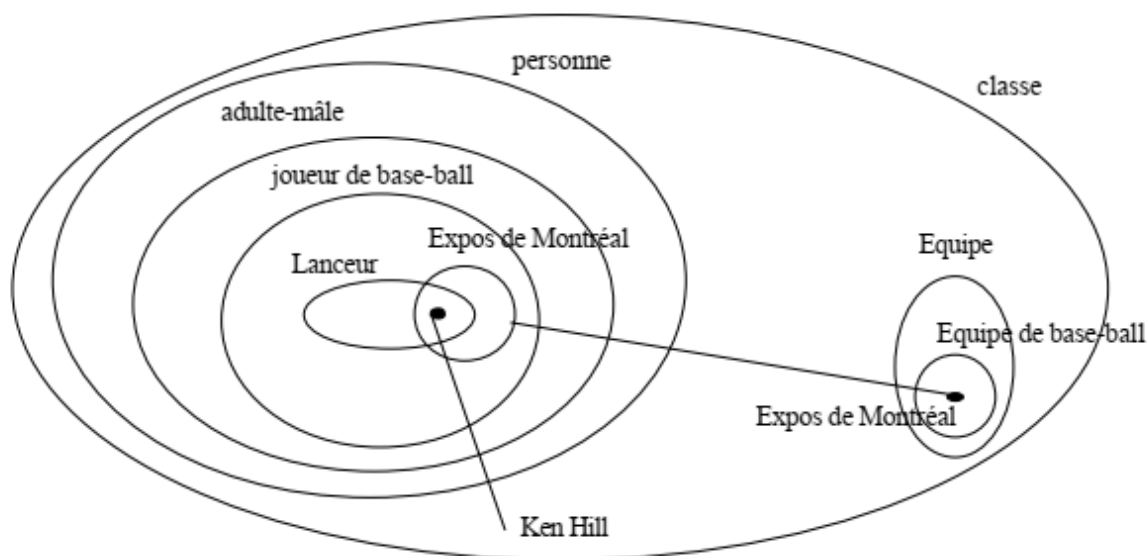


Figure 12: classes et relations est_un et sorte_de [Tourigny et Capus, 2001]

Cette figure représente les classes et les relations **est_un** et **sorte_de**.

Les données de description de ce système :

Chapitre I : Etat de l'art

Ken Hill	
est_un	<i>Lanceur</i>
est_un	<i>Expos de Montréal</i>
bras lanceur	<i>droit</i>

Lanceur	
sorte_de	<i>Joueur de baseball</i>
cardinal	<i>280</i>
*moyenne	<i>.180</i>

Joueur de base-ball	
sorte_de	<i>Adulte-mâle</i>
cardinal	<i>672 /* nombre total de joueurs */</i>
*taille	<i>1,85m</i>
*frappe	<i>idem dextérité</i>
*moyenne	<i>.250</i>

Adulte-mâle	
sorte_de	<i>personne</i>
cardinal	<i>2 milliards</i>
*taille	<i>1.75m</i>

Personne	
sorte_de	<i>classe</i>
cardinal	<i>6 milliards</i>
*dextérité	<i>droite</i>

Expos de Montréal	
est_un	<i>Équipe du base-ball majeur</i>
sorte_de	<i>Joueur de base-ball</i>
gérant	<i>Felipe Alou</i>
effectifs	<i>24</i>
*uniforme	<i>blanc à rayures bleues</i>
*nom_équipe	<i>Expos</i>

Équipe du base-ball majeur	
sorte_de	<i>équipe sportive</i>
est_un	<i>classe</i>
cardinal	<i>28 /* nombre d'équipes ... */</i>
effectifs	<i>24 / nombre de joueurs par équipe */</i>
*gérant	

Équipe sportive	
sorte_de	<i>classe</i>
est_un	<i>classe</i>
*gérant	
*uniforme	

Classe	
est_un	<i>classe</i>
sorte_de	<i>classe</i>

Figure 13: Les données de description de ce système [Tourigny et Capus, 2001]

Chapitre I : Etat de l'art

Chaque schéma a un nom et une liste d'attributs avec une valeur.

Les attributs avec le symbole * sont héritables par toutes les instances de la classe

Traitement des conflits d'héritage :

Exemple :

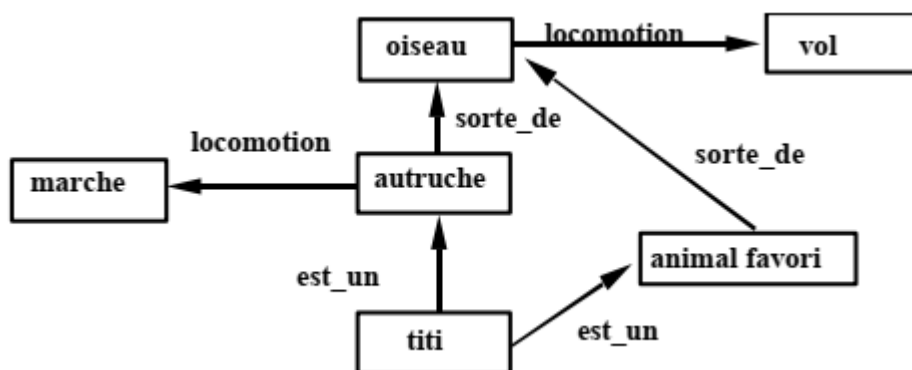


Figure 14: conflits d'héritage [Tourigny et Capus, 2001]

Le chemin "titi – autruche" donne priorité à "marche" pour l'attribut locomotion par contre "animal favori" lui donne la valeur vol. La solution est le plus court chemin entre "titi" et les classes dont titi hérite et fixée pour l'attribut locomotion.

Autre solution le critère de **proximité inférentielle**.

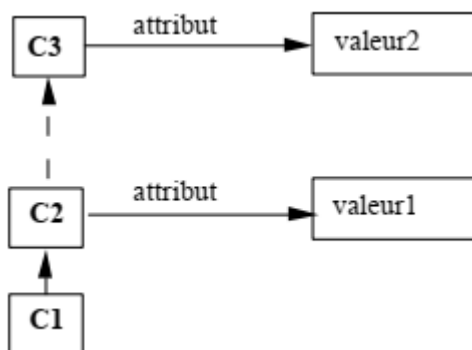


Figure 15: le critère de proximité inférentielle [Tourigny et Capus, 2001]

Chapitre I : Etat de l'art

Selon le critère de proximité inférentielle, C1 héritera de l'attribut valeur1

4-5-les scripts :

Un **script** est une représentation d'événements.

Les composantes d'un script sont :

- les **conditions d'entrée**, Exemples : la boutique est ouverte, le client a besoin de vêtement.
- le **résultat**, Exemples : le client a acheté des vêtements , le client n'a plus besoin de vêtement.
- les **objets spécifiques** (props), les entités utilisé : tables, chaises, garçon, client, chef, argent.
- les **rôles**, les actions des entités, Exemples : le vendeur présente les prix, le client choisit et paye.
- les **scènes**, les événements concernant les objets et les acteurs. Exemples : l'entrée au boutique, l'observation, l'achat, la sortie de la boutique.

Les scripts représentent un texte ou une histoire avec plus ou moins de détails.

4-6-Réseaux bayésiens :

Un Réseau Bayésien est un outil de présentation de connaissances probabilistes basé sur le théorème de Bayes, D'écrit les relations causales et le couplage entre les graphes et la probabilité pour l'aide à la décision et le diagnostic.

Exemple :

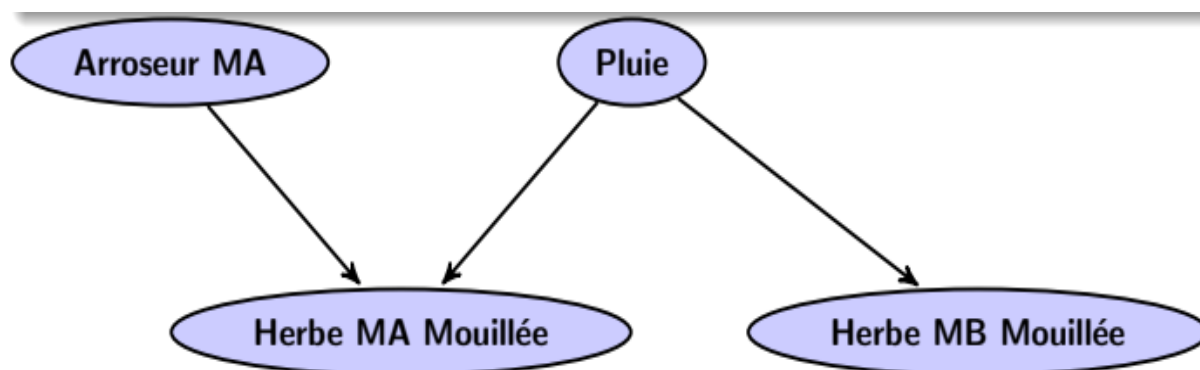


Figure 16: Réseau bayésien [Emmanuel ADAM]

Chapitre I : Etat de l'art

Les tables de probabilités sont construites d'après un nœud sachant la valeur des nœuds 'parents'.

Il pleut 4 jours sur 10 dans la région de MA & MB sachant que l'arroseur a 2 modes 'ON' et 'OFF' elle est en 10% du temps en mode 'ON'

P(A)		P(P_{luie})	
<i>true</i>	<i>false</i>	<i>true</i>	<i>false</i>
0.1	0.9	0.4	0.6

Tableau 5: table de pluie [Emmanuel ADAM]

L'herbe de MB mouillée sachant pluie :

P(MB P_{luie})			
$P_{luie} \downarrow$	MB \rightarrow	<i>true</i>	<i>false</i>
$(P_{luie} = true)$		0.99	0.1
$(P_{luie} = false)$		0.01	0.9

Tableau 6: relation MB-pluie [Emmanuel ADAM]

l'herbe de MA mouillée sachant l'état de l'arroseur et la pluie:

A	<i>on</i>	<i>on</i>	<i>off</i>	<i>off</i>
P_{luie}	<i>true</i>	<i>false</i>	<i>true</i>	<i>false</i>
$P(MA_{true} A, P_{luie})$	0.999	0.95	0.99	0.1
$P(MA_{false} A, P_{luie})$	0.001	0.05	0.01	0.9

Tableau 7: table de P(MA/A, Pluie) [Emmanuel ADAM]

Chapitre I : Etat de l'art

Théorème de Bayes :

$P(B/A) = P(A/B) / \sum_i (P(A/B_i) \times P(B_i))$ dont S est la somme

Il pleut 5 jours / 7 ! $P(\text{Pluie}) = 0:7$ et $P(\text{Pluie}^c) = 0:3$

Lorsqu'il pleut, je prends mon parapluie 3 fois sur 5 :

$P(\text{Para}/\text{Pluie}) = 0:6$

Sinon, au cas où, je prends mon parapluie 1 fois sur 10 :

$P(\text{Para}/\text{Pluie}^c) = 0:1$

Sachant que j'ai mon parapluie, quelle est la probabilité qu'il pleuve ?

Je vois personne avec un parapluie, quelle est la probabilité qu'il pleuve ?

$P(\text{Pluie}/\text{Para}) = (P(\text{Para}/\text{Pluie}) \times P(\text{Pluie})) / (P(\text{Para}/\text{Pluie}) \times P(\text{Pluie}) + P(\text{Para}/\text{Pluie}^c) \times P(\text{Pluie}^c)) = 0:6 \times 0:7 / (0:6 \times 0:7 + 0:1 \times 0:3) = 0:42 / 0:45 = 0:93$

Il y a 93% de chance qu'il pleuve quand je vois une personne avec un parapluie.

4-7-Logique de description :

Méthode pour la représentation de connaissances permet de représenter des individus qui sont des instances d'un concept (entité d'un domaine) et les relations entre eux avec un rôle.

- Représentation des concepts et des rôles au niveau terminologique TBox.
- Description et manipulation des individus au niveau des assertions ABox

Les axiomes de TBox : $A, C \sqcup D, C \sqsupseteq D, R \sqcup S, R \sqsupseteq S$ and $R^+ \sqcup R$

A concept atomique (N'hérite de rien sauf de Top satisfait), C, D concepts, R, S des rôles, et R^+ l'ensemble de rôles transitifs

Les axiomes de ABox sont sous forme $x:D, (x,y):R$

où x,y sont des noms d'individus, D est un concept et R est un rôle.

Exemple :

- Tbox: $\text{Homme} \sqsupseteq \text{Personne} \sqcap \text{Mâles}$
- Abox: $\text{Homme}(\text{Djamel})$
- Base de connaissances $\langle \text{Tbox}, \text{Abox} \rangle$

Chapitre I : Etat de l'art

La syntaxe de la LD :

$C \sqsubseteq A$	Subsorption de concepts	$\forall r.C$	Restriction universelle
\perp	Concept absurde	$\exists r.C$	Restriction existentielle
\top	Concept universel	$(\geq n \ r)$	Restriction supérieure de cardinalité
$C \sqcap D$	Conjonction de concepts	$(\leq n \ r)$	Restriction inférieure de cardinalité
$C \sqcup D$	Disjonction de concepts	$r \sqsubseteq a$	Subsorption de rôles
$\neg C$	Négation de concept	$r \wedge s$	Composition de rôles

Tableau 8:syntaxe de la logique de description [Nhan le Thanh]

Famille de langages LD :

Langage FLE :

Constructeur	Syntaxe	Exemple
Concept universel	\top	
Conjonction	$A \sqcap B$	Personne \sqcap Jeune
Quantification Universelle	$\forall R.C$	$\forall \text{enfant.Masculin}$
Quantification Existentielle	$\exists R.\top$	$\exists \text{enfant.}\top$

Tableau 9: langage FLE [Nhan le Thanh]

les AL :

$$AL = \{ \top, \perp, \neg A, C \sqcap D, \forall r.C, \exists r \}$$

$$ALC = AL \cup \{ \neg C \} \text{ (négation de concepts définis)}$$

$$ALU = AL \cup \{ C \sqcup D \} \text{ (disjonction de concepts)}$$

$$ALE = AL \cup \{ \exists r.C \} \text{ (quantification existentielle typée)}$$

$$ALN = AL \cup \{ \geq n \ r, \leq n \ r \} \text{ (cardinalité de rôles -remarque } \exists r \equiv (\geq 1 \ r))$$

$$ALR = AL \cup \{ r_1 \sqcup r_2 \} \text{ (conjonction de rôles)}$$

Chapitre I : Etat de l'art

SHOIN et SHIQ :

Sigle	Nom	Syntaxe	Exemple
S (\mathcal{ALC} et R^+)	Conjonction	$C \sqcap D$	personne \sqcap jeune
	Disjonction	$C \sqcup D$	vieux \sqcup jeune
	Négation	$\neg C$	\neg (personne \sqcap jeune)
	Q.U.	$\forall R.C$	\forall enfant.male
	Q.E.	$\exists R.C$	\exists enfant.male
	Rôle transitif (R^+)	$R.R$	père.père
\mathcal{H}	Rôle hiérarchique	$R \sqsubseteq S$	mère \sqsubseteq parent
\mathcal{O}	Concepts nominaux	$\{x\}$	{Terre, Mars, Venus}
\mathcal{I}	Inverse de rôle	R^-	enfant \Leftrightarrow parent
\mathcal{N}	Restriction de nombre	$\geq nR$	≥ 2 enfant
\mathcal{Q}	RN qualifiée	$\geq nR.C$	≥ 2 enfant.male

Tableau 10: SHOIN et SHIQ [Nhan le Thanh]

Exemple :

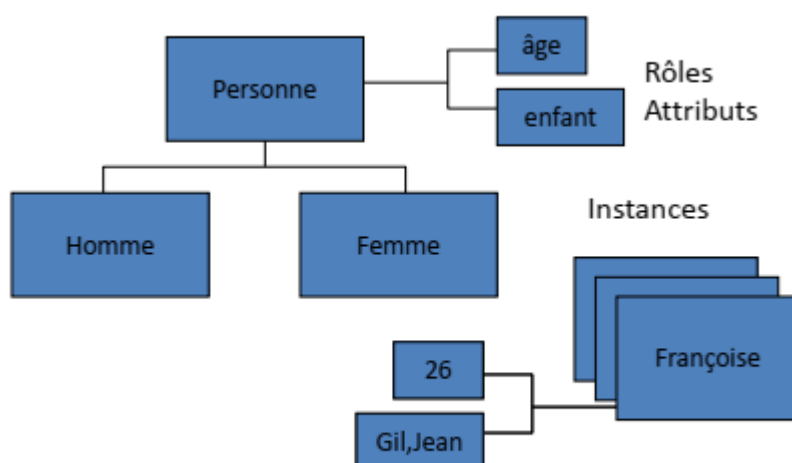


Figure 17: Inférence - héritage et instances [Julien Velcin ,2019]

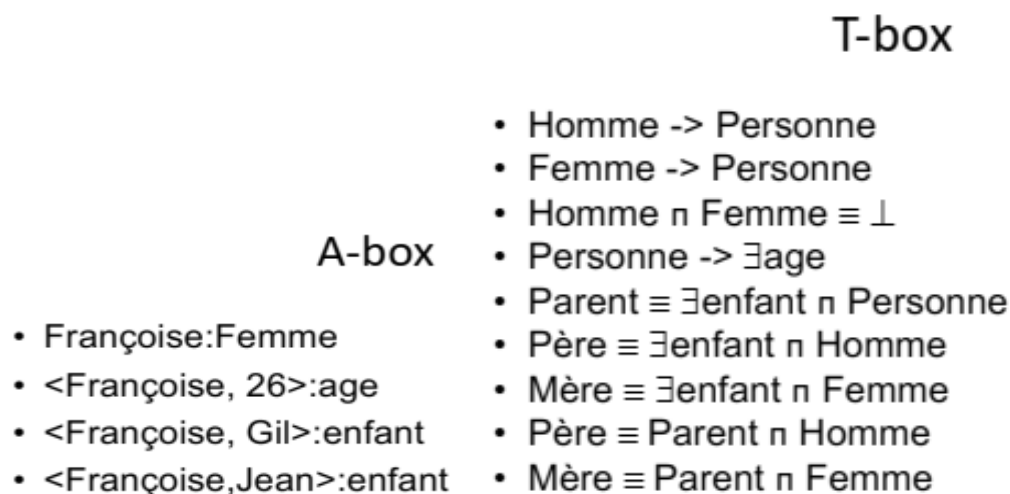


Figure 18:A-box[Julien Velcin ,2019]

Figure 19:T-box [Julien Velcin ,2019]

4-8-Les ontologies :

Dans l'univers informatique, le concept d'ontologie, qui est emprunté à la philosophie, a été utilisé dans plusieurs secteurs, principalement en intelligence artificielle, en représentation des connaissances, en traitement des langues naturelles, dans le web sémantique et en ingénierie logicielle [Breitman et coll., 2007].

Les ontologies rassemblent un vocabulaire de termes, des spécifications quant à la signification de ces termes et une description de la manière dont ils sont inter-reliés les uns aux autres [Ushold et Jasper,1999].

Les termes en question correspondent normalement à ce que l'on appelle « classes », « concepts » ou « choses ». Les spécifications correspondent pour leur part aux « propriétés » ou « attributs » que devrait avoir un objet tombant sous un certain concept. Et les descriptions d'interrelation représentent quant à elles les « relations » que certains concepts entretiennent conceptuellement avec d'autres concepts.

Les ontologies peuvent être utilisées pour décrire différents types de modèles. Elles peuvent par exemple décrire des taxonomies élémentaires comme elles peuvent décrire des modèles plus élaborés écrits dans une variante de la logique classique du premier ordre, comme la logique descriptive. [Breitman et coll., 2007].

Chapitre I : Etat de l'art

L'objectif d'une ontologie est de modéliser un ensemble de connaissances dans un domaine donné en langage interprétable par un ordinateur. Une ontologie est une description formelle et simplifiée du monde que l'on veut représenter. Une ontologie peut utiliser d'autres ontologies existantes (sources de données multiples), elle inclura nécessairement une spécification des objets et leurs significations.

Que représente-t-on dans une ontologie ?

- Concept : classe
- Relation : les relations entre les concepts
- Axiome : assertion considéré comme vrai
- Instances d'un concept (classe) : Omar est une instance de la classe personne.

Type d'ontologie :

- Par rapport à l'objet de la conceptualisation :
 - Ontologie de haut niveau : concepts très généraux, espace, temps ... Indépendant d'un domaine d'application
 - Ontologie de domaine : décrit le vocabulaire d'un domaine (médecine, automobile)
 - Ontologie d'application : décrit des concepts d'un domaine et d'une tâche particulière

Chapitre I : Etat de l'art



Figure 20:ontologie-représentation de l'exemple [Julien Velcin ,2019]

Exemple : ontologie représente une compagnie avec ses employées et les objets au tour d'elle

Les standards utilisés dans la création d'une ontologie :

URI :

L'URI est une sorte d'étiquette numérique (chaines de caractères) pouvant représenter à la fois l'adresse d'une ressource sur un réseau, lorsqu'il est considéré sous sa forme d'URL (Uniform Resource Locator), et/ou le « nom propre » de cette ressource, lorsqu'il est considéré sous sa forme d'URN (Uniform Resource Name). Une URL est donc un URI qui sert d'identificateur d'emplacement, alors qu'un URN est un URI qui sert à désigner nommer une ressource quelconque en dehors de toute référence à son emplacement. [Wikipédia/URI]

À titre de précision, mentionnons qu'un URI, vu sous son aspect d'URN, peut se décomposer en deux sous-parties : un URIfref (URI de référence) et un QName (nom qualifié). Un URIfref, ou identificateur de l'espace de nom, est un URI avec un fragment identificateur optionnel : le QName, qui identifie un nom particulier dans l'espace de nom identifié par l'URIfref. Il est à noter que l'URN peut s'écrire avec une syntaxe un peu différente de l'URL. Par exemple, une ressource qui utiliserait l'URI

Chapitre I : Etat de l'art

http://www.exemple.org/item1 pour identifiant, on pourrait l'interpeler par l'URN **urn:exemple:item1**

C'est de cette double fonction représentative (de localisation et de nomination) de l'URI que découle tout le potentiel de traiter sémantiquement les ressources web. En ce sens, elle révèle la base de connaissances à base d'ontologie en tant que dispositif de mappage d'un réseau d'URL, identifiant les ressources localisées à organiser, avec un réseau d'URN, définissant le réseau conceptuel des concepts et relations du domaine ontique auquel appartiennent les ressources à organiser. L'URI tient donc lieu dans ce portrait de pivot conceptuel permettant le mappage en question. C'est ainsi que l'annotation sémantique consiste en la mise en équivalence de deux URI, l'un utilisé pour sa capacité de pointer physiquement une ressource concrète et l'autre utilisé pour sa capacité de pointer un concept abstrait. [Michel Tétréault,2011]

Présentation de RDF (*Resource Description Framework*) :

RDF est un standard pour la représentation et l'échange de connaissances sur internet. Développé par le W3C, RDF est le langage de base du Web sémantique.

Règles de composition pour construire des phrases simples de type Sujet-Prédicat-Objet. On dira d'une telle phrase RDF de type **S-P-O** qu'elle forme un triplet RDF une connaissance (une proposition) tout ce qui est susceptible d'être vrai ou faux. Soumettre un mot seul à une telle épreuve de vérification n'a pas de sens. Par exemple, on ne peut dire d'un mot comme « chapeau » qu'il est vrai ou faux. Pour ce faire, il faut prédiquer le mot avec un attribut (propriété) et fixer cette dernière avec une valeur d'attribut. Ainsi, « le chapeau est de couleur rouge » est une affirmation.

De plus, une propriété intéressante de ce genre de structure conceptuelle triadique est la possibilité de les agréger et les représenter sous forme d'un réseau : les sujets et objets étant représentables par des nœuds que les prédicats peuvent relier en tant qu'arcs. C'est pourquoi dans la terminologie RDF, il y a souvent lieu de parler de graphe RDF. Il est à remarquer, avant de poursuivre, que dans un graphe RDF il est tout à fait possible pour un sujet d'être l'objet d'un autre triplet et/ou d'un objet de devenir le sujet d'un autre triplet.

Chapitre I : Etat de l'art



Figure 21: triplet RDF [Michel Tétréault,2011]

Une manière plus standard de penser le triplet RDF est sous la forme de table : la ligne représente un sujet, la colonne représente une propriété et la cellule la valeur (objet) de la propriété

	Propriété	
Sujet	Valeur	

Figure 22: table RDF [Michel Tétréault,2011]

Le document page.html a pour auteur Michel et a pour thème les ontologies ». En RDF, on pourrait exprimer cette description par les deux triplets suivants :

(page.html, aPourAuteur, Michel) et (page.html, aPourTheme, Ontologie)

Ce qui correspondrait au graphe élémentaire suivant :



Figure 23: graphe élémentaire des deux triplets [Michel Tétréault,2011]

Comme RDF a été conçu originalement pour décrire des ressources web, sa syntaxe formelle utilise généralement XML (on appelle alors cette forme RDF/XML)

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:local="http://mon_domaine.ca/Vocabulaire#">
```

Chapitre I : Etat de l'art

```
<rdf:Description rdf:about="http://mon_domaine.ca/page.html">  
<local:aPourAuteur>Michel</local:aPourAuteur>  
<local:aPourTheme>Ontologie</local:aPourTheme>  
</rdf:Description>  
</rdf:RDF>
```

Le modèle RDF offre quelques ressources terminologiques une des plus importantes est la relation de typification, `rdf:type` [Michel Tétréault,2011]



Figure 24:relation de typification [Michel Tétréault,2011]

Présentation de RDFS :

Un vocabulaire élémentaire de base à partir duquel on pourra construire des ontologies. L'espace de nom de ce vocabulaire est `http://www.w3.org/2000/01/rdf-schema#` et son préfixe usuel est « `rdfs:` ». Les trois termes les plus saillants de ce vocabulaire sont sans conteste ceux de classe (`rdfs:Class`), de sous-classe (`rdfs:subClassOf`) et de sous-propriété (`rdfs:subPropertyOf`).

En ce qui concerne l'élément `rdfs:Class`, il s'agit d'un terme pour identifier le type des entités qui sont l'objet de la propriété `rdf:type`. Reprenons l'exemple où l'on avait déterminé que « `page.html` » était un type de « `PageWeb` ». Mais que représente exactement l'entité « `PageWeb` » ? Si on affirmait que « `page.aspx` » est aussi de type « `PageWeb` », on viendrait en quelque sorte à définir implicitement que ces deux ressources ont en commun d'être de type « `PageWeb` ». En ce sens, « `PageWeb` » n'est pas une entité simple mais une classe d'entités.

On peut exprimer ce fait en déclarant que son type est `rdfs:Class`.



Figure 25:RDF classe [Michel Tétréault,2011]

Chapitre I : Etat de l'art

RDFS propose une propriété pour attester qu'une classe est sous-classe d'une autre : `rdfs:subClassOf`

Si l'on définissait que « PageWeb » est une sous-classe de « DocumentÉlectronique », on pourrait alors déduire que « page.html » est un document électronique.



Figure 26:RDF sous-classes [Michel Tétréault,2011]

La propriété `rdfs:subPropertyOf` permet d'élaborer des hiérarchies de propriétés.

Les propriétés RDFS `rdfs:domain` et `rdfs:range` : la première indique le type de sujet sur lequel peut porter la propriété alors que la seconde pointe vers le type d'objet auquel s'applique la propriété. Ceci permet alors de pouvoir prendre le nœud représentant la propriété réifiée pour sujet d'une propriété `rdfs:subPropertyOf` qui pointerait vers une super-propriété qui lui est plus générale. Ainsi, par rapport à notre exemple, on pourrait définir la propriété « aPourAuteur » comme sous-propriété de « aPourCreateur » [Michel Tétréault,2011]

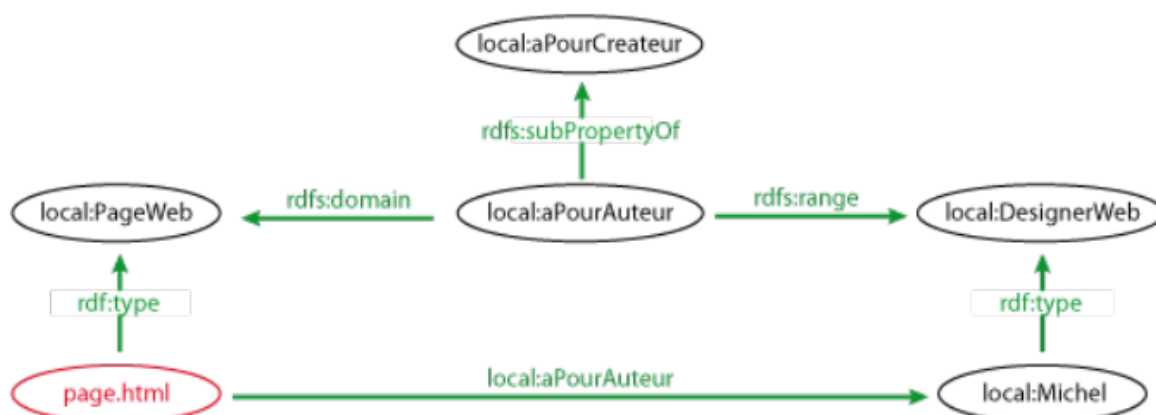


Figure 27:RDF subPropertyOf/domain et rang [Michel Tétréault,2011]

Chapitre I : Etat de l'art

Présentation de OWL :

OWL, qui est l'acronyme de Web Ontology Language, désigne un vocabulaire dédié à l'assertion de connaissances sous forme d'ontologie. Pour ce faire, ce standard a été conçu pour pouvoir être exprimable dans une syntaxe XML et reprendre des concepts et propriétés déjà définis dans RDF et RDFS. En ce sens, une base de connaissances OWL peut donc être écrite comme une collection de triplets RDF.

Un document OWL, qui décrit toujours une ontologie, se compose de faits et d'axiomes. Les faits assertent des choses sur les individus alors que les axiomes assertent des choses sur les concepts utilisés par l'ontologie.

Étant donné qu'une des motivations premières de construire une ontologie OWL est de permettre aux applications qui l'utilisent de pouvoir effectuer des inférences sur les données cumulées, OWL offre des opérateurs issus de la logique descriptive qui permettent de préciser et d'axiomatiser les descriptions en classe, propriété et individu fournis par RDF et RDFS.

OWL permet de définir des propriétés de propriété telles que :

owl:TransitiveProperty qui marque le caractère transitif d'une propriété.

owl:SymetricProperty qui marque le caractère symétrique d'une propriété.

owl:FunctionalProperty qui marque que la propriété ne peut avoir qu'une seule valeur pour un même sujet.

owl:InverseFunctionalProperty qui marque que pour une même valeur la propriété ne peut avoir qu'un seul sujet.

owl:inverseOf qui marque que la propriété est l'inverse d'une autre.

owl:equivalentProperty qui marque que la propriété est équivalente à une autre.

OWL définit aussi des constructeurs de classe qui servent à décrire des classes, non seulement en fonction des relations de leurs relations taxonomiques (ce que permettait rdfs:subClassOf), mais aussi en fonction des relations transversales qu'ils entretiennent entre eux.

En ce sens, OWL prédéfinit ainsi les classes chose (owl:Thing) et rien (owl:Nothing) qui représentent respectivement l'ensemble de tous les individus et l'ensemble vide.

Chapitre I : Etat de l'art

Il définit la relation owl:oneOf qui permet de décrire une classe par l'énumération exhaustive des individus qu'elle contient, le terme owl:restriction qui permet de décrire une classe par rapport à une certaine propriété que doivent avoir les individus qu'elle contient.

La relation owl:equivalentClass qui permet de déclarer deux classes comme étant équivalentes.

La relation owl:intersectionOf, qui permet de définir une classe coextensive de l'intersection de deux ou plusieurs autres classes.

La relation owl:unionOf qui permet de définir une classe coextensive de l'union de deux ou plusieurs autres classes.

La relation owl:complementOf qui permet de définir une classe par sa complémentarité avec une autre.

La relation owl:disjointWith qui permet de déclarer que l'extension d'une classe n'a aucun membre en commun avec l'extension d'une autre classe. [Michel Tétrault,2011]

5-Analyse et comparaison :

Contrairement aux graphes conceptuels, les frames offrent une connexion entre les types, leurs descriptions et les descriptions d'individus aussi, qui peut servir pour guider l'interprétation d'une scène, d'une situation ou d'une histoire ;

Les graphes conceptuels ont une rigueur logique de la représentation alliée à la considération d'aspects sémantiques.

D'un autre coté les réseaux sémantiques ont une bonne lisibilité et une bonne déclarativité.

Les scripts servent à communiquer, comprendre et raisonner sur une histoire, action ou un évènement ; ce sont une suite de contextes avec des transitions. Ils ne peuvent pas exprimer de connaissances incertaines, imprécises, hypothétiques.

Le besoin d'utiliser la notion de probabilité pour des connaissances probabilistes et la non existence de cette notion dans les méthodes déjà mentionnés a conduit à l'intégrer dans les réseaux bayésiens pour faciliter la compréhension et les calculs et améliorer l'expressivité.

D'autres manière d'exprimer les relations entre les individus sans se soucier de ses manipulations sont les calculs des prédicats de sorte qu'ils sont soit vrais soit faux avec des symboles et l'utilisation des formules et les axiomes pour démontrer et ce n'est pas le cas dans les représentations précédentes, son point fort est ses règles et ses théorèmes correctes, mais son inconvénient est de manipuler beaucoup d'hypothèses.

Chapitre I : Etat de l'art

Selon Tourigny et Capus, (2001) « Un autre avantage du calcul des prédicats est sa puissance pour formaliser le raisonnement et ses outils pour la déduction. »

Abdel-badeeh m.Salem, Marco Alfonse,2008 ont déclarés «aucune notion convenue sur la signification de la structure de représentation ,et il n'y a pas de possibilité d'exprimer les caractéristiques d'une propriété. L'ontologie définit le vocabulaire d'un domaine »

Le tableau résume la différence entre un réseau sémantique et une ontologie :

Caractéristiques	Réseau sémantique	Ontologie
Nom unique	Si deux objets ont des noms différents, on suppose qu'ils sont différents	Il n'y a pas d'hypothèse sur le fait que les objets soient identiques ou différents, à moins qu'il n'y ait une déclaration explicite sur la spécification de la relation
Monde ouvert vs fermé	Rien ne peut y être inscrit tant qu'il n'y a pas de place pour lui dans le modèle correspondant	Tout peut être entré dans l'ontologie sauf s'il viole une des contraintes
Capacité à définir des règles	Des règles peuvent être appliquées lors de la mise en œuvre du réseau sémantique en utilisant PROLOG	Aucune règle ne peut être appliquée lors de la mise en œuvre de l'ontologie en utilisant OWL

Tableau 11: comparaison entre le réseau sémantique et l'ontologie [Abdel-badeeh m. Salem, Marco Alfonse,2008]

Pour la logique de description son but est de donner un sens aux informations sur le web et être explicite pour mieux la lire. Tbox définit les concepts et les rôles et leur taxonomie, ABox décrit les individus en les nommant et en spécifiant leurs classes et attributs (se base sur la classification et l'instanciation)

On peut dire que l'ontologie est outil très pratique et important, elle regroupe des graphes conceptuels, des frames, des réseaux sémantiques. Ce sont tous des formalismes utilisables plus une manipulation de la logique de description etc. pour créer plus précisément une base de connaissances, alors l'idée est de conserver des formes de représentation de connaissances dans une seule ontologie.

Chapitre I : Etat de l'art

On peut résumer les points précédents par le tableau suivant :

Techniques	Avantages	Inconvénients
Ontologie	<ul style="list-style-type: none"> -Permet de spécifier les caractéristiques des propriétés (transitiveProperty, SymmetricProperty, etc.) -Permet des combinaisons booléennes (union, intersection, complément) des classes. -Réutilisation de connaissance de domaine. (Temps, Espace...) 	
Réseau sémantique	<ul style="list-style-type: none"> -Proche du langage courant et facile à mettre en œuvre. -Adapté pour les problèmes de classification(taxonomie). 	<ul style="list-style-type: none"> - les relations représentables sont au plus binaires ; -mécanisme de déduction lourd.
Calcul des prédicats	<ul style="list-style-type: none"> -Permet d'exprimer des relations complexes -Puissant pour formaliser le raisonnement. 	-Symbolisme plus lourd par rapport au réseau sémantique
Script	-la capacité de représenter toutes les sortes de	Les scripts servent à communiquer, comprendre

Chapitre I : Etat de l'art

	<p>connaissances d'un domaine.</p> <p>Adapté à la représentation d'évènements.</p>	<p>et raisonner sur une histoire, action ou un évènement ; ce sont une suite de contextes avec des transitions. Ils ne peuvent pas exprimer de connaissances incertaines, imprécises, hypothétiques</p>
Frames	<ul style="list-style-type: none"> -Il est facile d'ajouter des slots pour de nouveaux attributs et relations. -Il est facile d'inclure des données par défaut et de rechercher des valeurs manquantes. 	<p>-Sémantique non précise.</p>
Logique de description	Sémantique formelle définie en logique de première ordre	
Réseau bayésien	<ul style="list-style-type: none"> - Permet aussi bien de calculer les probabilités des conséquences à partir des probabilités des causes que de calculer les probabilités des causes à partir des conséquences. - Permet de combiner des informations de différentes sources. - Peut donner de bon résultat même avec un petit ensemble de données. 	<ul style="list-style-type: none"> - Préférable de les utiliser avec des données discrètes que continues. - Difficile de collecter et de structurer la connaissance des experts.

Chapitre I : Etat de l'art

	- Plus rapide que les modèles de simulation.	
Graphe conceptuel	- Bonne lisibilité grâce à l'aspect visuel. - Bon niveau de déclarativité.	-Pas de sémantique claire.

Tableau 12: Comparaison entre les techniques de représentation [Laura Uusitalo. (2007), MARTIN F.R. (2019), Abdel-Badeeh M. Salem, Marco Alfonse. (2008), Jean-Gabriel Ganascia]

5-1-Discussion et conclusion sur la solution choisie :

Par flexibilité nous entendons la possibilité de réutiliser dans plusieurs contextes différents la même information. C'est-à-dire que, la plupart du temps, on peut réutiliser une ontologie développée en ajoutant un minimum d'information. Aucune modification n'est nécessaire.

Prenons un exemple simple pour illustrer ce point.

Admettons une ontologie très simplifiée d'un réseau informatique comme illustré à la Figure 28. Nous y définissons une hiérarchie de classes avec une classe de base "Machine" et une sous-classe "Serveur".

Chapitre I : Etat de l'art

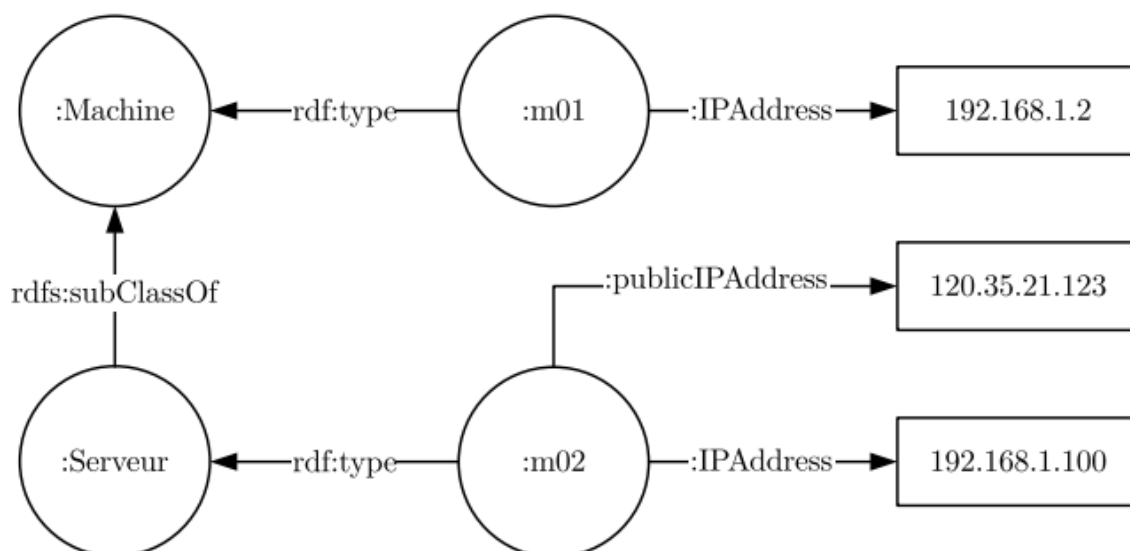


Figure 28 :ontologie d'un réseau informatique [simon malenfant-corriveau,2017]

Admettons maintenant un système de détection d'intrusion qui générerait ses alertes sous la forme d'une ontologie et qui aurait été exécuté sur le réseau décrit par l'ontologie de la Figure28. Ce cas est illustré à la Figure29. On y décrit une hiérarchie de classe pour l'alerte avec comme sous-classe 'Injection SQL' . On y trouve aussi une instance d'alerte de type injection SQL qui a été détectée sur l'ordinateur ayant comme adresse IP 192.168.1.100.

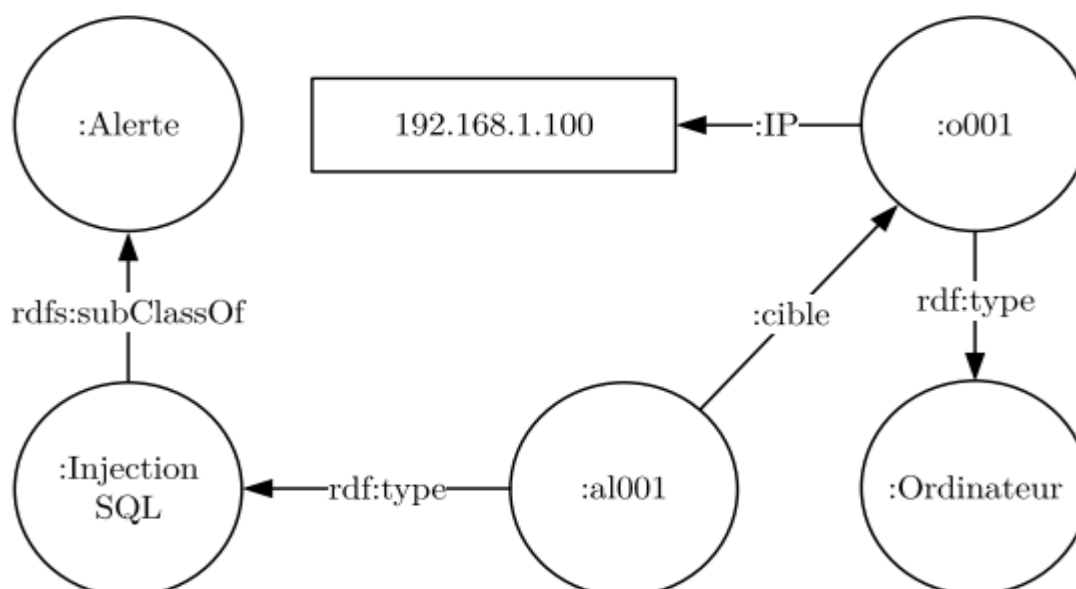


Figure 29: ontologie d'un système de détection d'intrusion [simon malenfant-corriveau,2017]

Chapitre I : Etat de l'art

On voudrait alors pouvoir fusionner les deux ontologies dans la même base de données ontologique.

En comparant les deux ontologies proposées, on voit qu'elles décrivent les mêmes choses, mais avec un vocabulaire différent. Pour regrouper les deux ontologies, il faudrait donc ajouter de l'information indiquant les termes synonymes. Il faudrait dire :

1. Les propriétés :IP et :IPAddress sont synonymes
2. Les classes :Ordinateur et :Machine sont synonymes
3. Toutes les instances de type :Machine ayant une même adresse IP sont équivalentes

On obtiendrait alors l'ontologie partiellement illustrée dans la Figure30 .On a regroupé les noms synonymes sous un même nœud. En interrogeant cette ontologie, l'administrateur de notre exemple pourrait par exemple constater que la machine attaquée est définie comme étant un serveur et il pourrait trouver son adresse publique en plus de l'information qu'il avait déjà dans l'ontologie du système de détection d'intrusion.

On peut conclure de cet exemple très simple que la combinaison de l'information représentée dans les ontologies est beaucoup plus facile et naturelle que si elle était représentée dans une base de données relationnelle, par exemple. Ce qu'on entend par "naturel" ici est que ce processus se rapproche beaucoup de la manière dont l'être humain emmagasine la connaissance.

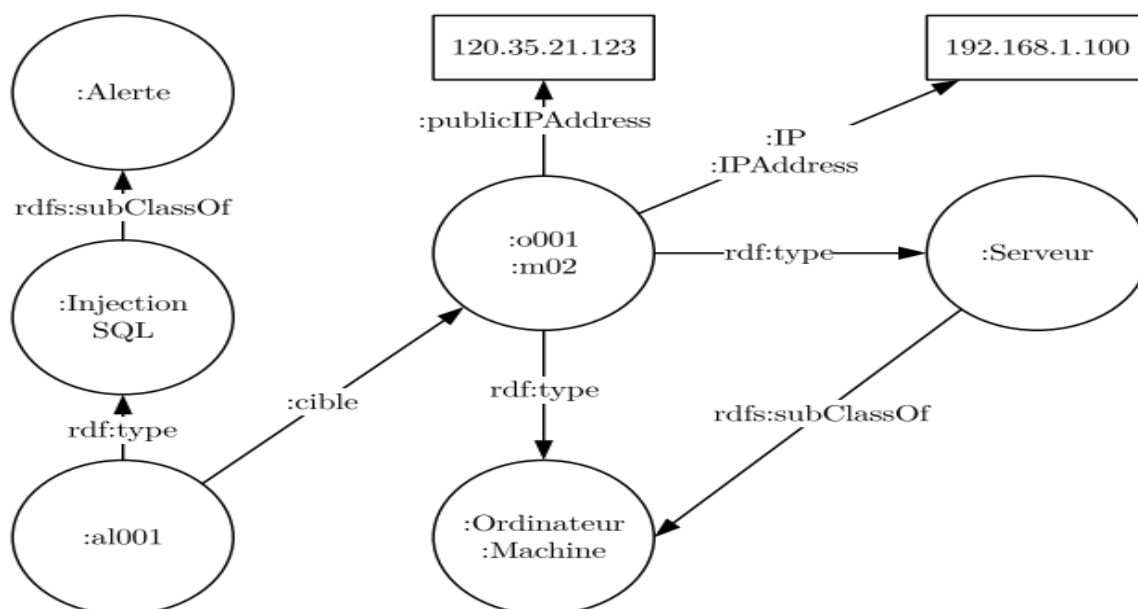


Figure 30:combinaison des ontologie réseau et SDI [simon malenfant-corriveau,2017]

Chapitre I : Etat de l'art

Nous pouvons illustrer l'avantage de l'abstraction du point de vue de la représentation des connaissances par l'ontologie. Dans ce cas, l'abstraction se traduit de deux manières. Premièrement, elle se manifeste par les propriétés "subClassOf" et "subPropertyOf", qui permettent de faire des hiérarchies de classes et de propriétés [simon malenfant-corriveau,2017]

D'après ce que nous avons vu précédemment, le meilleur outil à ce jour pour la représentation des connaissances est l'ontologie et de ce fait, notre solution sera basée sur l'utilisation d'une ontologie pour la création de notre base de connaissances.

Conclusion :

Dans ce chapitre nous avons présenté une vue générale sur les différentes approches de la représentation des connaissances. En particulier, nous avons introduit les principales définitions, éclaircis les notions présentées et faire des comparaisons. Enfin, nous notons que le domaine de la représentation des connaissances est un domaine très vaste et très complexe, et loin d'être englobé dans sa totalité dans le cadre d'un seul mémoire.

Chapitre II :
Conception d'une
application d'aide au
diagnostic automobile

Chapitre II : Conception d'une application d'aide au diagnostic automobile

1-Introduction à la spécification des besoins :

Dans ce chapitre, nous allons décrire la conception et la modélisation de notre système. Pour cela, nous allons utiliser le langage UML, qui est un langage de modélisation graphique qui va nous permettre de comprendre et de décrire nos besoins, de spécifier et de documenter le système. La conception de notre application se base sur :

- Les diagrammes des cas d'utilisations.
- Les diagrammes de séquences.
- Les diagrammes de classes.

Afin de pouvoir réaliser cette conception nous devons recenser les acteurs du système ainsi que leurs interactions avec celui-ci et qui seront représentés dans les diagrammes des cas d'utilisation. Nous allons aussi donner les événements du système de manière chronologique qui seront établies dans les diagrammes de séquences, enfin nous décrirons la structure interne du système dans le diagramme de classes.

2-Diagramme de cas d'utilisation :

Les diagrammes de cas d'utilisation sont des diagrammes UML utilisés pour donner une vision globale du comportement fonctionnel d'un système logiciel. Dans un diagramme de cas d'utilisation, les utilisateurs sont appelés acteurs (actors), ils interagissent avec les cas d'utilisations (use cases). Pour notre système, on peut distinguer un seul acteur principal : **utilisateur**. Notre application doit donc offrir un ensemble de fonctionnalités. Ces fonctionnalités sont classées par acteur selon des cas d'utilisations.

Chapitre II : Conception

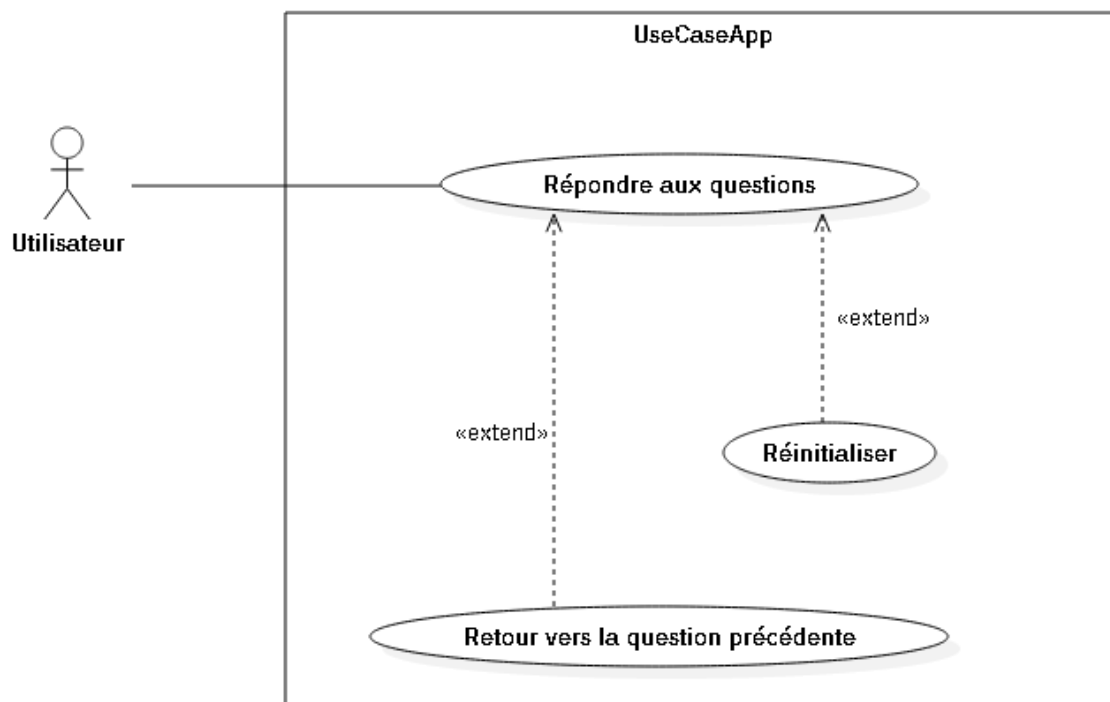


Figure 31: diagramme cas d'utilisation

Au lancement de l'application l'utilisateur démarre le processus de questions-réponses en appuyant sur le bouton 'commencer'

L'utilisateur après avoir appuyé sur le bouton 'commencer', voit s'afficher des questions, il doit juste répondre à ces questions. Il peut revenir en arrière en cas d'erreur ou bien réinitialiser complètement les questions depuis le début.

Chapitre II : Conception

3-Le diagramme de séquence :

L'utilisateur va interagir avec l'application avec une séquence de messages entre lui et le système pendant cette interaction. On peut représenter les messages transmis entre les objets (le système et l'utilisateur), en utilisant les diagrammes de séquences suivants :

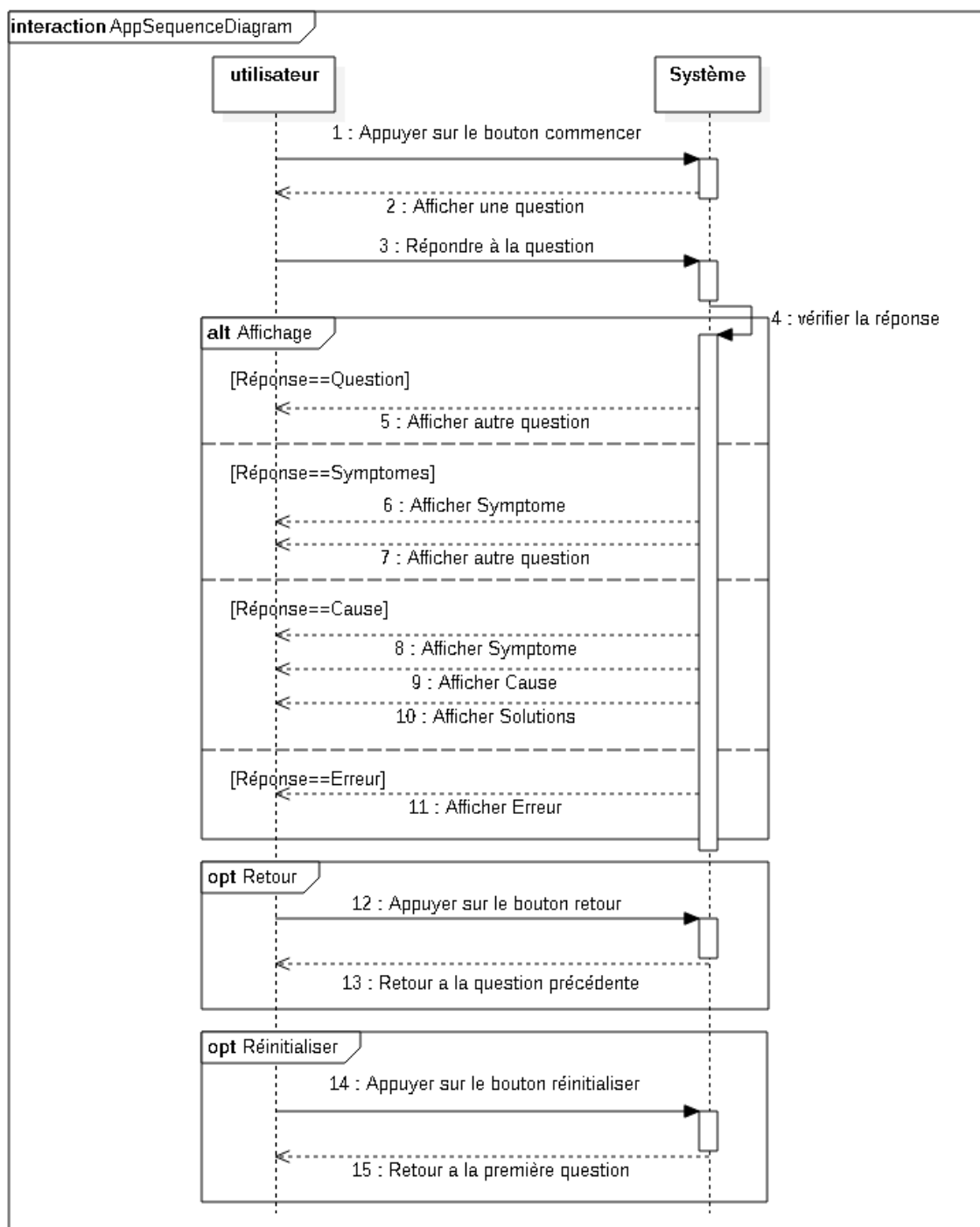


Figure 32 : Diagramme de séquence

Chapitre II : Conception

Après avoir répondu aux questions l'application va afficher soit d'autres questions soit un symptôme soit une cause et ses solutions.

4-Diagramme de classe :

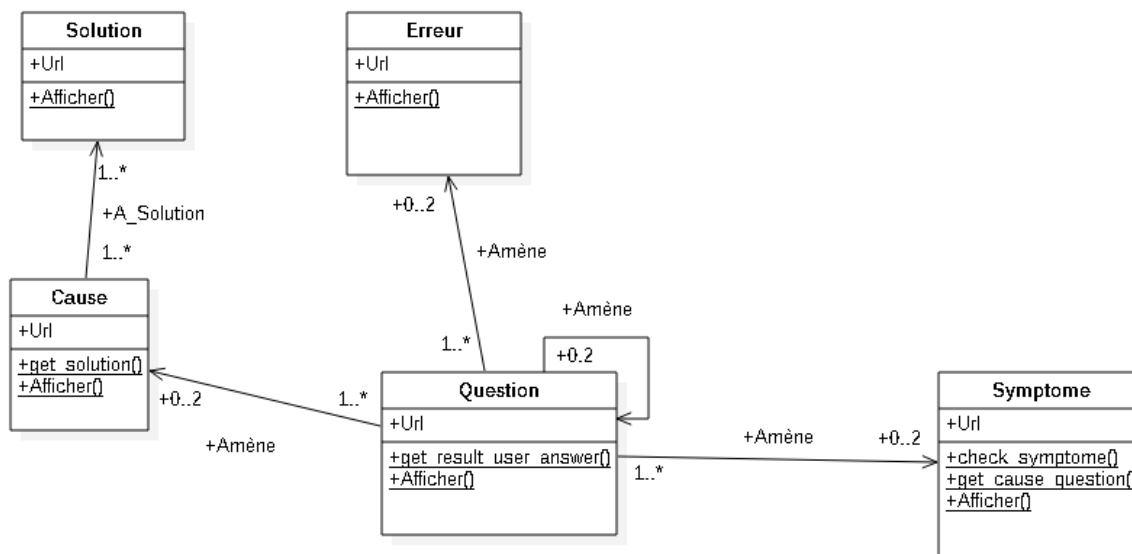


Figure 33: Diagramme de classes

Chapitre III :

Implémentation

Chapitre III : Implémentation

Chapitre III : Implémentation :

1-Introduction :

Pour développer un outil informatique, on a besoin d'utiliser différents outils et technologies.

Dans notre cas (le développement d'un outil informatique basé sur une ontologie), nous avons utilisé les outils suivants :

- Un langage de programmation (Python)
- Un Environnement de Développement Intégré (EDI) (Pycharm)
- Un Système de gestion de données d'ontologie (Protegé)

Nous allons présenter l'environnement de travail ainsi que les outils et les langages de programmation utilisés, qui nous ont permis de réaliser notre système. Nous présenterons les outils avec des illustrations

2-Les outils utilisés :

2-1-Python :

Python est un langage de programmation, dont la première version est sortie en 1991. Créé par **Guido van Rossum**, la **Python Software Foundation**, créée en 2001. Ce langage a été baptisé ainsi en hommage à la troupe de comiques les « Monty Python ». Il favorise la programmation impérative structurée, fonctionnelle et orientée objet. Il est doté d'un typage dynamique fort, d'une gestion automatique de la mémoire par ramasse-miettes et d'un système de gestion d'exceptions ; Le langage Python est placé sous une licence libre et peut aussi être traduit en Java ou .NET..[Wikipédia /python]

À quoi peut servir Python ?

Python est un langage puissant, à la fois facile à apprendre et riche en possibilités. Dès l'instant où vous l'installez sur votre ordinateur, vous disposez de nombreuses fonctionnalités intégrées au langage

Il est, en outre, très facile d'étendre les fonctionnalités existantes, Ainsi, il existe ce qu'on appelle des **bibliothèques** qui aident le développeur à travailler sur des projets particuliers.

Chapitre III : Implémentation

Plusieurs bibliothèques peuvent ainsi être installées pour, par exemple, développer des interfaces graphiques en Python.

Concrètement, voilà ce qu'on peut faire avec Python :

- De petits programmes très simples, chargés d'une mission très précise sur votre ordinateur ;
- Des programmes complets, comme des jeux, des suites bureautiques, des logiciels multimédias, des clients de messagerie.
- Des projets très complexes, comme des progiciels (ensemble de plusieurs logiciels pouvant fonctionner ensemble, principalement utilisés dans le monde professionnel).

Voici quelques-unes des fonctionnalités offertes par Python et ses bibliothèques :

- créer des interfaces graphiques.
- faire circuler des informations au travers d'un réseau.
- dialoguer d'une façon avancée avec votre système d'exploitation.

Python est un langage de programmation **interprété**, c'est-à-dire que les instructions que vous lui envoyez sont « transcrites » en langage machine au fur et à mesure de leur lecture. D'autres langages (comme le C / C++) sont appelés « langages **compilés** ». Les avantages d'un langage interprété sont la simplicité et la portabilité (un langage tel que Python est censé fonctionner aussi bien sous Windows que sous Linux ou Mac OS, et on ne devrait avoir à effectuer aucun changement dans le code pour le passer d'un système à l'autre). [openclassroom/python]

2-2-La bibliothèque RDFlib :

RDFlib est un module Python qui permet de manipuler des graphes et des triplets RDF. Les ontologies OWL étant stockées dans un graphe RDF, nous pouvons utiliser RDFlib pour manipuler celui-ci.

2-3-SPARQL :

est un langage d'interrogation de données définies en RDF, ce langage est en cours de standardisation au niveau du W3C. Sa syntaxe est très proche de SQL.[W3C06]

Exemple de requête SPARQL :

SELECT ?sujet ?predicat ?objet

Chapitre III : Implémentation

FROM dataSet_name

WHERE {

 ?sujet ?predicat ?objet .

}

- La clause SELECT définit la liste des variables que l'on désire obtenir. Une variable est composée de caractères alphanumériques et commence par un point d'interrogation ' ? '.
- La clause FROM définit l'emplacement des documents RDF utilisés pour la requête.

La clause FROM n'est obligatoire que lorsque l'on souhaite parcourir un graphe nommé et non celui par défaut.

- La clause WHERE définit le triplet RDF (sujet - prédicat - objet), les éléments de ce triplet sont décrits soit par les valeurs de l'ontologie interrogée soit par des variables.

2-4-Kivy :

Kivy est une bibliothèque libre et open source pour Python, Cette bibliothèque fonctionne sur Android, iOS, GNU/Linux, OS X et Windows. [Wikipédia/Kivy]

La langue KV est un langage dédié à la description des composants de l'interface utilisateur. Il inclut des concepts tels que les définitions de règles (qui ressemblent un peu à CSS pour les applications web) etc. le langage se compose de plusieurs éléments :

Règles : Une règle est similaire à une règle CSS. Une règle s'applique à des widgets spécifiques (ou à des classes de ceux-ci) dans votre arborescence de widgets et les modifie d'une certaine manière. Vous pouvez utiliser les règles pour spécifier un comportement interactif ou les utiliser pour ajouter des représentations graphiques aux widgets auxquels elles s'appliquent. Vous pouvez cibler une classe spécifique de widgets (similaire au concept CSS d'une classe) en utilisant l'attribut cls (par exemple cls=MyTestWidget).

Un widget racine : Vous pouvez utiliser KV pour créer toute votre interface utilisateur. Un fichier KV ne doit contenir qu'un seul widget root au maximum.

Classes dynamiques : Les classes dynamiques vous permettent de créer de nouveaux widgets et de nouvelles règles à la volée. [api-kivy]

Chapitre III : Implémentation

2-4-1-KivyMD :

Le **Material Design** est un ensemble de règles de design proposées par Google et qui s'appliquent à l'interface graphique des logiciels et applications.

Le but de KivyMD est d'imiter le design proposé par google, pour les développeurs utilisant le Framework KIVY.

2-4-2-Buildozer :

Buildozer est un outil utilisé pour la création d'application Android pour les applications développée avec Kivy.

2-5-Logiciel pour l'ontologie :

2-5-1-Protégé :

Protégé est un système auteur pour la création d'ontologies. Il a été créé à l'université Stanford et est très populaire dans le domaine du Web sémantique et au niveau de la recherche en informatique.

Protégé est développé en Java. Il est gratuit et son code source est publié sous une licence libre (la Mozilla Public License).

Protégé peut lire et sauvegarder des ontologies dans la plupart des formats d'ontologies : RDF, RDFS, OWL, etc.

Il possède plusieurs concurrents tels que Hozo, OntoEdit et Swoop [archive]. Il est reconnu pour sa capacité à travailler sur des ontologies de grandes dimensions. [Wikipédia/protégé]

Chapitre III : Implémentation

3-Présentation de l'application réalisée :

Pour notre projet d'application, nous avons eu besoin des connaissances d'experts dans le domaine du dépannage automobile, pour d'abord comprendre les besoins de représentation de connaissances pour la création de notre ontologie, puis pour remplir notre base de connaissances.

Pour cela nous avons utilisé le livre de A.BENZADI & D.BOURAÏ - Initiation Au Dépannage Automobile; Editeur: EDC OMEGA, qui nous a permis de remplir notre base de connaissances avec les pannes de moteur données dans ce livre.

Exemple question-symptôme-cause-solution depuis le livre :

Symptôme	Cause	Solution	Question
Le moteur tourne lentement mais ne démarre pas	Batterie déchargée	-Recharger Ou remplacer. -Mettre en route le moteur au moyen d'une batterie ou de câbles volants.	-Est-ce qu'il y a une mauvaise odeur du capot -Est-ce que les équipements fonctionnent faiblement ou pas du tout -Est-ce que le voyant de batterie est allumé -Est-ce que la tension est sous 10v ou entre 11v et 12,7v
	Démarrreur défectueux	Réparer ou remplacer	-Est-ce que le démarrreur tourne -Vérifier le balai, est ce que le courant passe

Chapitre III : Implémentation

	Mauvaises connexions dans le circuit de démarrage	Vérifier si les câbles sont hors services, Resserrer les connexions.	-Est-ce qu'il reste allumé après le démarrage du moteur (le démarreur) -Si le démarreur ne s'active pas mais que vous tournez à nouveau l'interrupteur a clé, est ce que cela fonctionne
--	---	--	---

Tableau 13:Exemple des pannes moteur présentées dans le livre [A.BENZADI & D.BOURAÏ]

L'ontologie, la première étape de ce projet.

Elle se compose de 5 classes (question, erreur, symptôme, cause, solution) chaque classe a ses instances (Individual).

La classe « erreur » : si le problème de l'utilisateur dépasse le champ de connaissances du système, l'application proposera à l'utilisateur de consulter un spécialiste.

Les classes sont reliés entre elles par des ObjectProperty, pour chaque ObjectProperty on définit un domaine et un range.

Sujet prédicat objet == class ObjectProperty class

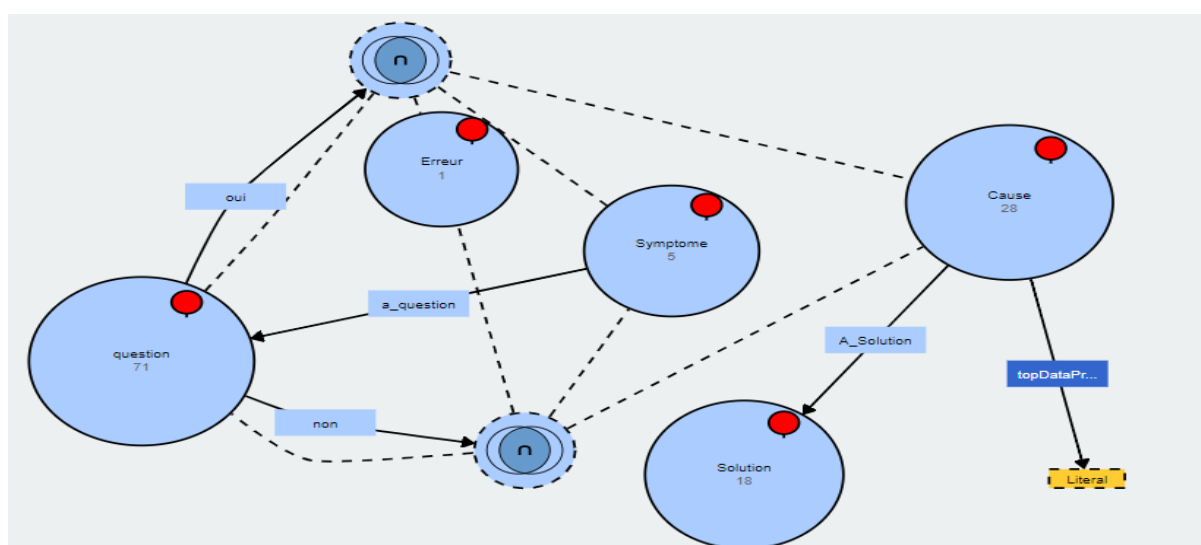


Figure 34: Graphe représentant les classes de l'ontologie et le nombre d'instances pour chaque classe

Chapitre III : Implémentation

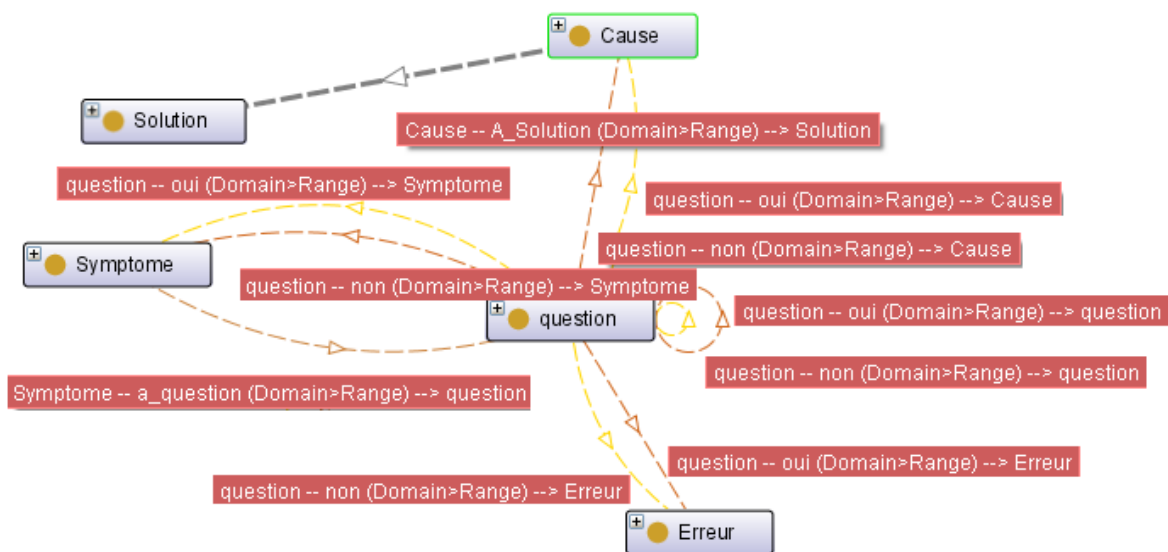


Figure 35: graphe représentant l'ontologie utilisant OntoGraf

Cause et ses instances (INDIVIDUAL) :

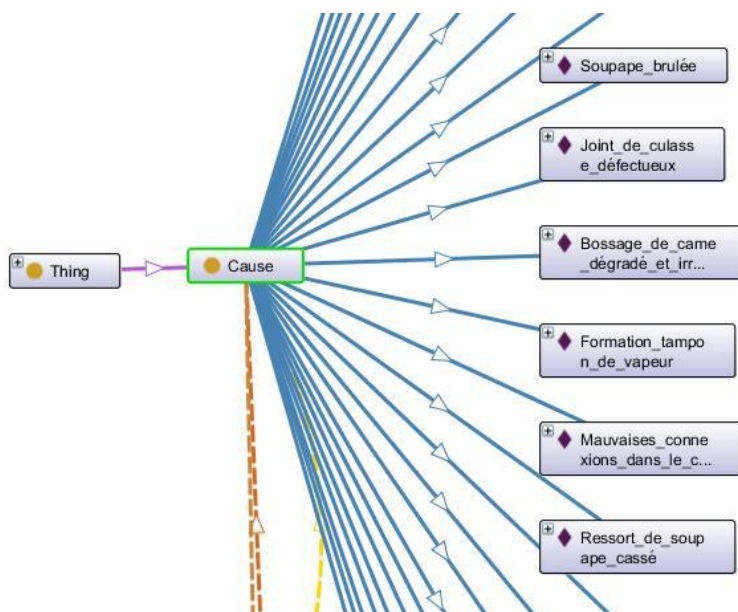


Figure 36:les causes

Chapitre III : Implémentation

Symptôme et ses instances (INDIVIDUAL) :

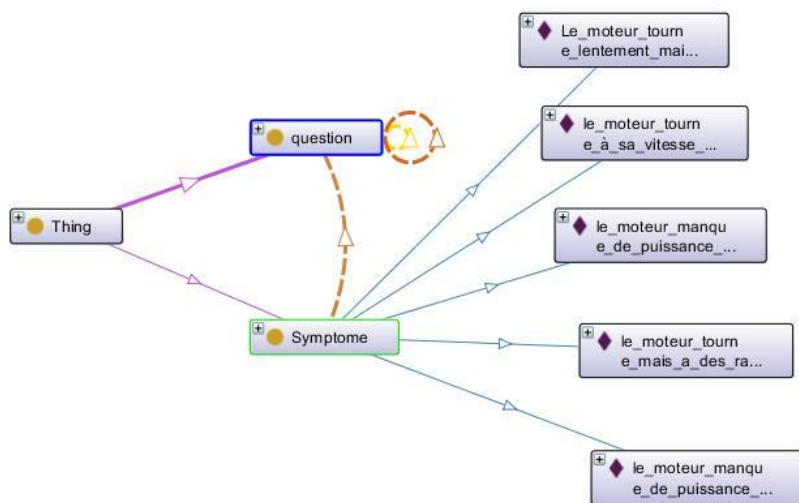


Figure 37:les symptômes

Question et ses instances (INDIVIDUAL) :

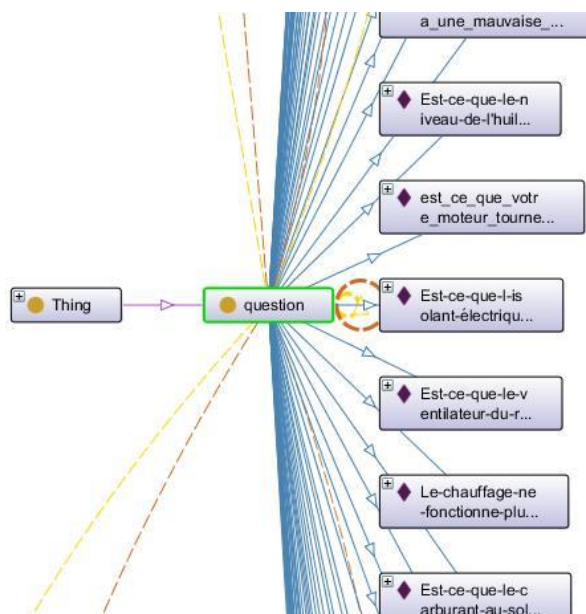


Figure 38: les questions

Chapitre III : Implémentation

Erreur et ses instances (INDIVIDUAL) :

La classe erreur intervient quand le problème de l'utilisateur dépasse les compétences de la base de connaissance.



Figure 39: classe erreur

Solution et ses instances (INDIVIDUAL) :

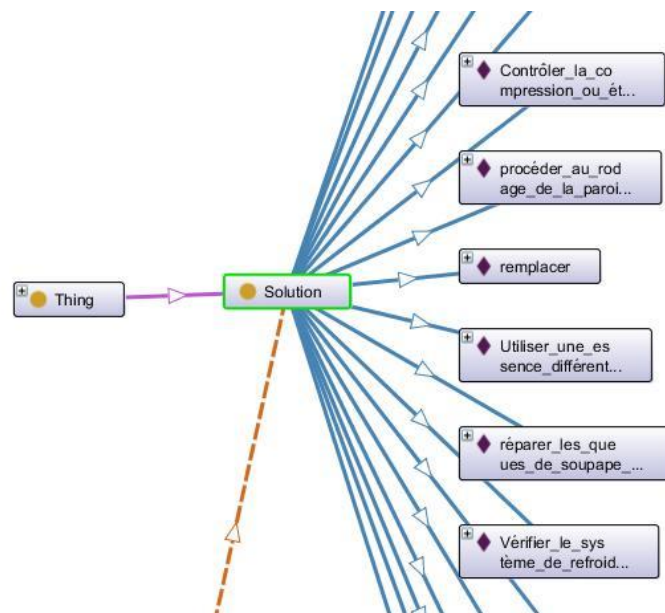


Figure 40: les solutions

Chapitre III : Implémentation

Les requêtes pour interroger l'ontologie avec SPARQL :

Réponse de l'utilisateur :

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX car: <http://www.semanticweb.org/lenovo/ontologies/2020/4/untitled-ontology-6#>

SELECT distinct ?autrequestion ?classreponse
WHERE {
  car: "" + question + "" > car: "" + user_answer1 + "" ?autrequestion. ?autrequestion a
  ?classreponse
}
```

Requête Cause :

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX car: <http://www.semanticweb.org/lenovo/ontologies/2020/4/untitled-ontology-6#>

SELECT distinct ?symptome ?question
WHERE {
  car: "" + symptome + "" car:a_question ?question. }
```

Requête Solution :

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX car: <http://www.semanticweb.org/lenovo/ontologies/2020/4/untitled-ontology-6#>

SELECT distinct ?solution
WHERE {
  car: "" + cause + "" car:A_Solution ?solution. }
```

Chapitre III : Implémentation

Images de l'application :

Pour la première fenêtre nous avons un bouton start pour commencer le quiz.

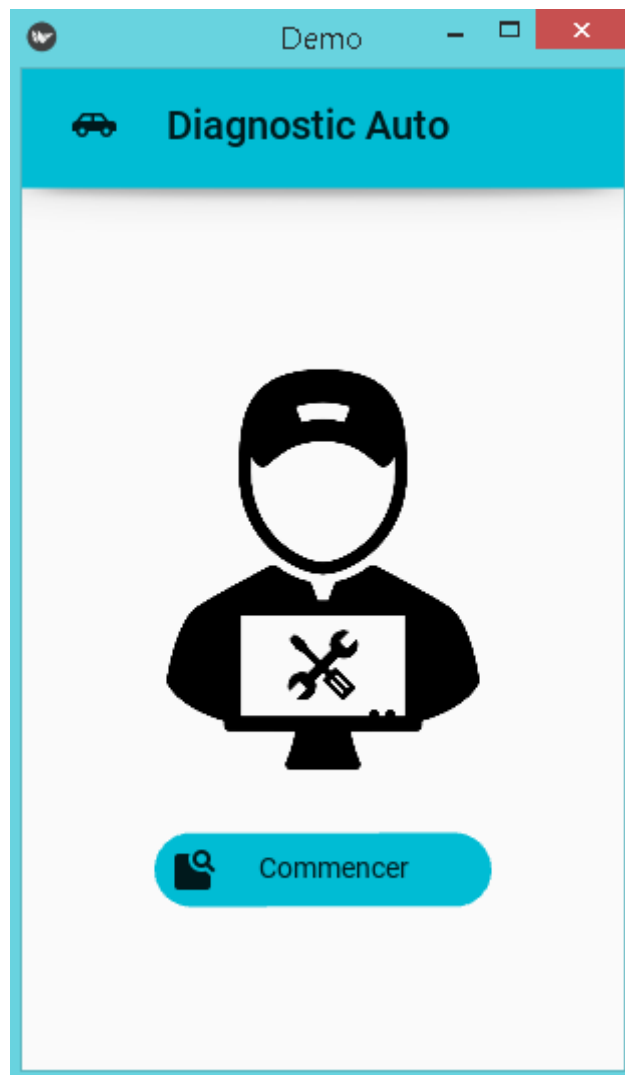


Figure 41: première fenêtre

Chapitre III : Implémentation

On commence notre quiz avec la première question et l'utilisateur a le choix de répondre soit par oui soit par non.



Figure 42 : première question

Chapitre III : Implémentation

L'utilisateur continue de répondre aux questions jusqu'à ce que le symptôme s'affiche et pose d'autres questions pour trouver la cause de la panne.

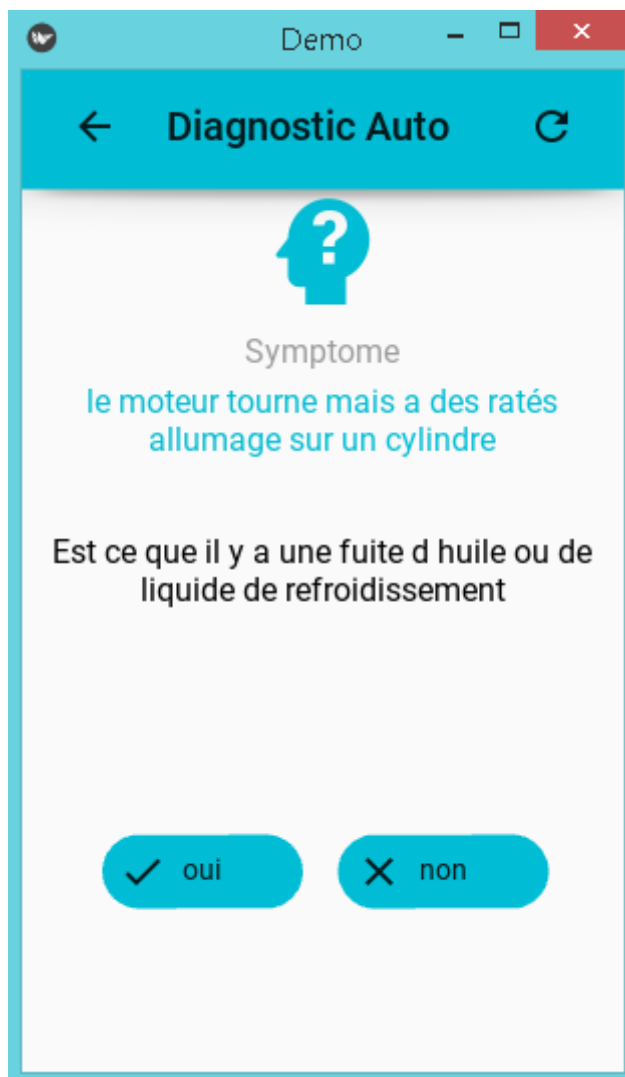


Figure 43 : fenêtre symptôme avec question

L'utilisateur répond aux questions jusqu'à ce que l'application propose une panne avec des solutions.

Si l'utilisateur considère qu'il a mal répondu à une des questions, il peut revenir en arrière en appuyant sur le bouton 'retour' en haut à gauche ou réinitialiser à la première question en appuyant sur le bouton 'réinitialiser' en haut à droite.



Figure 44:résultat du quiz

Conclusion :

Dans cette partie nous avons donné quelques définitions sur les technologies que nous avons utilisé dans le développement de notre application ainsi que le schéma de l'ontologie de notre base de connaissances, dans le but de faire la présentation de l'application qui est simple à utiliser et peut aider des propriétaires de véhicule qui auraient une panne de voiture et qui aimeraient en savoir plus, sans avoir besoin de perdre son temps à chercher l'information sur le net.

Conclusion générale

Conclusion générale :

La problématique que nous avons choisie pour ce mémoire nous a amené à nous intéresser aux méthodes de représentation de connaissances, dans le but de représenter les connaissances liées au domaine du diagnostic automobile.

Dans un premier temps nous avons fait un état de l'art sur les différentes techniques et méthodes disponibles en matière de représentation des connaissances. Puis nous avons fait une comparaison de ces techniques et méthodes qui nous a amené à choisir les ontologies en tant que méthode la plus aboutie sémantiquement.

Pour le travail réalisé, l'ontologie est une représentation formelle et une exploitation sémantique des connaissances dans le domaine du diagnostic automobile.

Nous avons donc, tout d'abord, procédé à la création d'une base de connaissance, à partir d'un travail de documentation sur les principales pannes mécaniques automobiles.

Enfin nous avons développé une application mobile sous forme de système de questions-réponses qui servira d'interface utilisateur pour requêter cette base de connaissances.

Pour pouvoir évaluer d'une façon concrète l'application, il aurait été souhaitable de disposer de plus de connaissances en mécanique voire en électricité automobile. Par ailleurs, il aurait été souhaitable de disposer d'un mécanicien qui puisse nous donner son opinion.

Pour le moment, nous avons fait une self évaluation et elle est correcte comparée aux données récupérées du livre de mécanique auto cité.

L'objectif de cette première partie du travail était de fournir un squelette d'une application qui devra dans un second temps être enrichi, non par intervention humaine mais de façon automatique.

En effet, de nombreux site de mécanique automobile proposent des articles de fond et mêmes certains forums offrent des données précieuses. Le prochain travail consistera à construire un système d'enrichissement de la base de connaissances sans inférence humaine à partir de ces sites de qualité et des forums spécialisés de même qu'il devrait permettre d'annoter les diagnostics établis, d'ajouter le concours de l'utilisateur qui pourra faire profiter de ses compétences et expériences l'ensemble de la communauté des apprentis-mécanos.

Le travail dans sa phase initiale répond aux objectifs fixés par le cahier des charge

Références :

- *[Axel Reymonet, Jérôme Thomas, Nathalie Aussenac-Gillesè.(2010)]. Ontologies et Recherche d'Information : une application au diagnostic automobile.
- *[Bernard ESPINASSE,2004] - Réseaux Sémantiques, Frames et Scripts – Université d'Aix-Marseille.
- * [Emmanuel ADAM]. Intelligence Artificielle - Représentation des connaissances - Université Polytechnique des Hauts-De-France.
- *[Christine Jouve,1992]. Représentation des connaissances pour les problèmes de conception. Application à un système à base de connaissances pour la conception de réseaux informatiques : NEST Université Jean Monnet - Saint-Etienne. Consulté sur : <https://tel.archives-ouvertes.fr/tel-00832243>
- *[Tourigny et Capus, 2001] représentation des connaissances université Laval. Consulté sur : ww2.ift.ulaval.ca
- * [F.-Y. Villemin,2012] Représentation des Connaissances le CNAM paris. Consulté sur : <http://deptinfo.cnam.fr>
- * [Thierry Hamon,2018] Représentation des connaissances Réseaux sémantiques - Université d'Orléans. Consulté sur : <https://perso.limsi.fr/hamon/Teaching/Orleans/RC-2018-2019>
- *[Assia AMARIR, Habib BENLAHMER, El Houssine LABRIJI,2015]. l'utilisation du formalisme des graphes conceptuels sur les ontologies arabes - la troisième journée sur les technologies d'information et de modélisation. Laboratoire de technologie de l'information et modélisation. Maroc.
- *[Thérèse libourel,2019] Modéliser une ontologie -donipat - données interopérables pour le patrimoine Université de Montpellier
- *[Daniel Kayser, 1997] - La représentation des connaissances -Hermes.
- *[Nhan le Thanh] Introduction à la logique de description -Ecole doctorale STIC, UNSA
- *[Julien Velcin ,2019] représentation des connaissances université de Lyon.
- *[Wikipédia]. Python. Consulté sur : [https://fr.wikipedia.org/wiki/Python_\(langage\)](https://fr.wikipedia.org/wiki/Python_(langage))
- *[open Classroom/python]. Consulté sur : <https://openclassrooms.com/fr/courses/235344-apprenez-a-programmer-en-python/230659-decouvrez-python>

- *[livre python]. Consulté sur : [https://www.editions-
eni.fr/open/mediabook.aspx?idR=0f5557cfca86c91854743f58167ae3da](https://www.editions-eni.fr/open/mediabook.aspx?idR=0f5557cfca86c91854743f58167ae3da)
- *[W3C06]. Candidate recommandation W3C SPARQL, 2006. Consulté sur : <http://www.w3.org/TR/rdf-sparql-query/>.
- *[Wikipédia]. Kivy. Consulté sur : <https://fr.wikipedia.org/wiki/Kivy>
- *[api-kivy]. Consulté sur : <https://kivy.org/doc/stable/api-kivy.lang.html>
- *[Wikipédia]. Protégé. Consulté sur : [https://fr.wikipedia.org/wiki/Prot%C3%A9g%C3%A9_\(logiciel\)](https://fr.wikipedia.org/wiki/Prot%C3%A9g%C3%A9_(logiciel))
- *[simon malenfant-corriveau,2017]. Proposition d'une méthode de développement d'ontologie pour un système expert en sécurité - université de Montréal.
- *Breitman, K. K., Casanova, M.A., & Truszkowski, W. (2007). Semantic Web: Concepts, Technologies and Application. London: Springer-Verlag
- *Uschold, M. Jasper, R. (1999). A Framework for Understanding and Classifying Ontology Applications. Compte rendu de l'atelier du IJCAI-99 Ontologies ans Problem Solving Methods, Stockholm.
- *[Wikipédia]. URI. Consulté sur : <http://fr.wikipedia.org/wiki/URI>
- *[Michel Tétreault,2011] Modélisation d'une ontologie et conceptualisation d'une application sémantique dédiée au e-recrutement dans le domaine des technologies de l'information-Université de Montréal
- *[Laura Uusitalo. (2007)] Advantages and challenges of Bayesian networks in environmental modelling. Ecological modelling (312-318). Consulté sur: https://www.academia.edu/19274811/Advantages_and_challenges_of_Bayesian_networks_in_environmental_modelling
- * [Fernando SilvaParreiras,Marco, AntonioCalijorne Soares.(2020)].A literature review on question answering techniques, paradigms and systems. Journal of King Saud University - Computer and Information Sciences (635-646).
- *[MARTIN F.R. (2019)] How Important Are Semantic Networks in Artificial Intelligence? Consulté sur: <https://analyticsindiamag.com/semantic-networks-ai/>

*[Abdel-Badeeh M. Salem, Marco Alfonse. (2008)] Ontology versus Semantic Networks for Medical Knowledge Representation. Proceedings of the 12th WSEAS international conference on Computers(769–774)

*[Jean-Gabriel Ganascia]. Représentation des connaissances. Université Pierre et Marie Curie.

*[A.BENZADI & D.BOURAÏ] - Initiation Au Dépannage Automobile; Editeur: EDC OMEGA

* [Ashish Salunkhe.(2020)]. Evolution of Techniques for Question Answering over Knowledge Base A Survey. International Journal of Computer Applications. Consulté sur : <https://www.researchgate.net/publication/338632170>