**République Algérienne Démocratique et Populaire**

**Ministère d'Enseignement Supérieur et de la Recherche Scientifique**
**Université ABBAS LAGHROUR- Khenchela-**
**département MI**

# Mémoire

Présenté en vue d'obtenir le diplôme de

## Master en Informatique (LMD)

# *FOUILLE DE DONNEES SEMANTIQUE, CONTRIBUTION DANS LE CADRE DE LA METHODES D'ARBRES DE DECISION.*

Présenté Par :

**Hallaci Leyla**

**Bensaidi Nadia**

**Encadrer Par : Dr Hamem Mounir.**

**Année universitaire 2019/2020**


بسم الله الرحمن الرحيم

# REMERCIEMENTS

*Nous remercions en premier lieu الله le tout puissant de nous avoir accordé la puissance et la volonté pour terminer ce travail.*

*Nous tenons à exprimer notre reconnaissance à Monsieur Dr Hamem Mounir.*

*Pour avoir dirigé ce mémoire, pour son suivi permanent, ses lectures attentives, ses conseils judicieux et le soutien constant qu'il nous 'a prodigué au cours de ce travail.*

*En nous tenons à remercier tous ceux qui de près ou de loin, ont contribué à la réalisation de ce modeste travail.*

*Nous remercions* chaleureusement les membres du jury de mémoire pour avoir accepté de juger ce travail.

**Merci**

***Résumé* :**

Parmi les techniques utilisées en Data Mining, les arbres de décisions sont devenus un concept majeur et ont reçu une attention considérable de recherche. Le domaine d'arbre de décision est assez vaste aujourd'hui. Un chercheur dans ce domaine trouve des difficultés pour comprendre les différents concepts de base les logiciels, les algorithmes d'apprentissage …….. Etc. Les ontologies, l'un des modèles de représentation de connaissances les plus utilisés, répond à cette problématique. Ainsi, nous proposons dans ce mémoire de construire une ontologie qui sert à décrire les taches, les entités de base et les algorithmes d'arbres de décisions dont le but de partager une compréhension commune de cette méthode et d'expliquer ce qui est considéré comme implicite.

***Mots clés :*** *Ontologie, Fouilles de données, Arbre de décision, Connaissance.*

***ABSTRACT:***

Among the techniques used in Data Mining, decision trees have become a major concept and have among the techniques used in Data Mining, decision trees have become a major concept and received considerable research attention. The decision tree is large enough now. A researcher in this field find difficulties to understand the different basic concepts software, learning algorithms... Etc. Ontologies, one of the most used knowledge representation models, meets this problem. Thus, we propose in this memory to build an ontology that is used to describe the tasks, basic entities and the algorithms of decision trees with the aim to share a common understanding of this method and explain what is considered as implied. received considerable research attention. The decision tree is large enough now. A researcher in this field find difficulties to understand the different basic concepts software, learning algorithms... Etc. Ontologies, one of the most used knowledge representation models, meets this problem. Thus, we propose in this memory to build an ontology that is used to describe the tasks, basic entities and the algorithms of decision trees with the aim to share a common understanding of this method and explain what is considered as implied.

***Key words:*** *Ontology*, *Data mining*, Decision tree, *Knowledge*

# Table de Matière

# Liste des figures

# Abréviations

- **OWL : W**eb **O**ntology **L**anguage
- **RDF : R**esource **D**escription **F**ramework
- **RDF-S : R**esource **D**escription **F**ramework- **S**cema
- **W3C : W**orld **W**ide **W**eb **C**onsortium
- **XML :** e**X**tensible **M**arkup **L**anguage
- **XMLS :** e**X**tensible **M**arkup Langage **S**chema
- **HTML** :**H**yper**T**ext **M**arkup **L**anguage
- **URI** : **U**niform **R**esource **I**dentifier
- **DAML : D**ARPA **A**gent Markup **L**anguage
- **OntoDT : Onto**logy of **D**ecision **T**ree
- **XHTML :** Extensible HyperText Markup Language
- **RSS : R**eally **S**imple **S**yndication
- **Math ML :** Mathematical Markup Language
- **CSS : C**ascading **S**tyle **S**heets
- **FaCT : Fa**st **C**lassification of **T**erminologies
- **DL : D**escription **L**ogic
- **DT: D**ecision **T**ree
- **DTA** : **O**ntology-guided **D**ecision **T**rees **A**ssistance
- **DTs** : **D**ecision **T**ree Schéma
- **SMIL : S**ynchronized **M**ultimedia **I**ntegration **L**anguage

# *Introduction Générale*

**Problématique**

Depuis le commencement de l'humanité, l'humain apprend sur le monde qui l'entoure, son intelligence lui a permis d'acquérir des connaissances. La mémoire de l'humain est très restreinte, à comparer à celle d'un ordinateur. Pour mieux se servir de ses connaissances, elle doit associer une classe à cette connaissance. L'humain se sert de son raisonnement pour associer une connaissance à une classe. Cette classification est le regroupement d'idées qui permet de distinguer un objet par rapport à un autre. Pour cela, les chercheurs en intelligence artificielle se servent de techniques diverses pour associer des idées à une classe. Pour mieux classifier une situation particulière, l'intelligence artificielle s'inspire souvent de la nature pour mieux interpréter des connaissances. Plusieurs approches peuvent être utilisées par exemple: **les arbres de décision**, les algorithmes génétiques…..etc.

Un arbre de décision est un outil d'aide à la décision qui représente la situation plus ou moins complexe que l'on représente sous la forme graphique d'un arbre de façon à faire apparaître à l'extrémité de chaque branche (ou feuille) les différents résultats possibles en fonction des décisions prises à chaque étape. L'arbre de décision est un outil utilisé dans des domaines variés (sécurité, fouille de données, médecine, etc.). Sa lisibilité, sa rapidité d'exécution et le peu d'hypothèses nécessaires a priori expliquent sa popularité actuelle.

Le domaine d'arbre de decision est assez vaste aujourd'hui. Un chercheur dans ce domaine trouve des difficultés pour comprendre les différents concepts de base et les logiciels, les Algorithme d'apprentissage Etc.

Les ontologies, l'un des modèles de représentation de connaissances les plus utilisées, répond à cette problématique.

Le domaine des ontologies connaît un grand essor depuis le début des années 90 et leur champ d'application ne cesse de s'élargir. L'ingénierie des connaissances a grandement contribué à diffuser le terme « ontologie » qui est devenu un élément clé dans toute une gamme d'applications faisant appel aux connaissances dans divers domaines tels que les systèmes d'aide à la décision,

systèmes d'enseignement assisté par ordinateur (notamment le e-Learning), les systèmes de résolution de problèmes, les systèmes de gestion de connaissances, le génie logiciel, domaine médicale, domaine de data mining, l'arbre de decision …etc.

- **Objectif de travail** :

  L'objectif de ce mémoire consiste à la construction d'une ontologie du domaine d'arbre de décision.

  La construction de notre ontologie s'est réalisée en trois phases fondamentales :

  i) **la conceptualisation** essentiellement fondée sur la méthode *METHONTOLOGY* qui a donné lieu à une ontologie conceptuelle.
  ii) **l'ontologisation** consiste en une formalisation partielle, sans perte d'information, du modèle conceptuel.
  iii) **l'opérationnalisation** qui a abouti à une ontologie opérationnelle cohérente et consistante.

- **Organisation du mémoire**

Ce mémoire est constitué de 4 chapitres structuré de la manière suivante :

- **Le Chapitre 1** s'intéresse à présenter une généralité sur le data Mining et les arbres de décisions.
- **Le Chapitre 2** présente les ontologies développées dans le domaine d'arbre de décision et web sémantique.
- **Le chapitre 3** introduit une présentation générale d'un cycle de vie d'une ontologie suit d'une représentation de la méthode avec une structure de notre ontologie.
- **Le chapitre 4** présente l'implémentation et l'évaluation de notre ontologie avec quelques résultats et expérimentations.

Ce mémoire s’achève par une conclusion générale en présentant un récapitulatif du contexte de recherche de notre étude et notre contribution tout en planifiant les perspectives que nous envisageons pour compléter ce travail.

# *Chapitre 1*

# *FOUILLE DE DONNEES ET ARBRE DE DECISION*

## 1.1 Introduction :

Les entreprises, mais aussi, dans une certaine mesure, les administrations, subissent aujourd'hui l'intensification de la concurrence ou la pression des administrés. Ces facteurs les poussent à porter une attention toujours plus grande aux clients (d'autant plus que leurs richesses aujourd'hui résident dans leurs clients), à améliorer constamment la qualité de leurs produits et à accélérer de manière générale leurs processus de mise sur le marché de nouveaux produits et services.

Pour répondre à ces besoins de découvertes, un ensemble d'architectures, de démarches et d'outils, certains nouveaux, d'autres existants depuis longtemps, a été regroupé sous le terme «*Data Mining*».

Ce premier chapitre présente les concepts de fouille de données, où les différentes étapes du processus d'extraction de connaissances à partir des données sont décrites. Nous insistons sur les différentes approches de mise en œuvre d'un modèle de fouille de données.

## 1.2 Etapes d'un processus d'extraction de connaissances à partir des données :

### Définition

*«L'Extraction de Connaissances à partir des Données (ECD) est un processus itératif et interactif d'analyse d'un grand ensemble de données brutes afin d'en extraire des connaissances exploitables par un utilisateur analyste qui y joue un rôle central»* [Zig01].

D'après [Fay96], un processus d'*ECD* est constitué de quatre phases qui sont : *le nettoyage et intégration des données, le prétraitement des données, la fouille de données* et enfin *l'évaluation et la présentation des connaissances.*

**La figure 1.1** récapitule ces différentes phases ainsi que les enchaînements possibles entre ces phases. Cette séparation est théorique car en pratique, ce n'est pas toujours le cas. En effet, dans de nombreux systèmes, certaines de ces étapes sont fusionnées [Kod98].

**Connaissances**

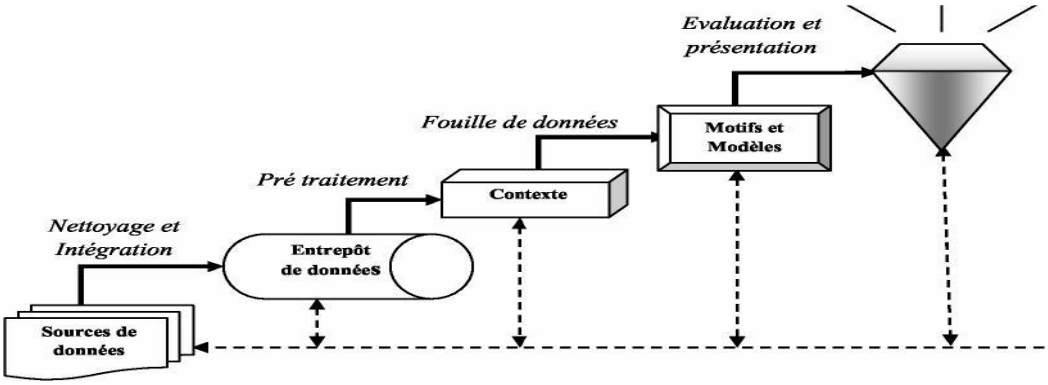



**Figure 1.1**: Processus d'extraction de connaissances à partir des données [Fay96].

### 1.2.1 Nettoyage et intégration des données

Le nettoyage des données consiste à retravailler ces données bruitées, soit en les supprimant, soit en les modifiant de manière à tirer le meilleur profit.

L'intégration est la combinaison des données provenant de plusieurs sources (base de données, sources externes, etc.). Le but de ces deux opérations est de générer des entrepôts de données et/ou des magasins de données spécialisés contenant les données retravaillées pour faciliter leurs exploitations futures.

### 1.2.2 Pré-traitement des données

Il peut arriver parfois que les bases de données contiennent à ce niveau un certain nombre de données incomplètes et/ou bruitées. Ces données erronées, manquantes ou

inconsistantes doivent être retravaillées si cela n'a pas été fait précédemment. Dans le cas contraire, durant l'étape précédente, les données sont stockées dans un entrepôt. Cette étape permet de sélectionner et transformer des données de manière à les rendre exploitables par un outil de fouille de données.

Cette seconde étape du processus d'*ECD* permet d'affiner les données. Si l'entrepôt de données est bien construit, le pré-traitement de données peut permettre d'améliorer les résultats lors de l'interrogation dans la phase de fouille de données.

### 1.2.3 Fouille de données :

La fouille de données *(data mining* en anglais), est le cœur du processus d'*ECD*. Il s'agit à ce niveau de trouver des pépites de connaissances à partir des données. Tout le travail consiste à appliquer des méthodes intelligentes dans le but d'extraire cette connaissance. Il est possible de définir la qualité d'un modèle en fonction de critères comme les performances obtenus, la fiabilité, la compréhensibilité, la rapidité de construction et d'utilisation et enfin l'évolutivité. Tout le problème de la fouille de données réside dans le choix de la méthode adéquate à un problème donné. Il est possible de combiner plusieurs méthodes pour essayer d'obtenir une solution optimale globale.

Nous ne détaillerons pas d'avantage la fouille de données dans ce paragraphe car elle fera l'objet de la section 1.3.

### 1.2.4 Evaluation et présentation :

Cette phase est constituée de l'évaluation, qui mesure l'intérêt des motifs extraits, et de la présentation des résultats à l'utilisateur grâce à différentes techniques de visualisation. Cette étape est dépendante de la tâche de fouille de données employée. En effet, bien que l'interaction avec l'expert soit importante quelle que soit cette tâche, les techniques ne sont pas les mêmes. Ce n'est qu'à partir de la phase de présentation que l'on peut employer le terme de *connaissance* à condition que ces motifs soient validés par les

experts du domaine. Il y a principalement deux techniques de validation qui sont la technique de validation statistique et la technique de validation par expertise.

**La validation statistique :**

consiste à utiliser des méthodes de base de statistique descriptive. L'objectif est d'obtenir des informations qui permettront de juger le résultat obtenu, ou d'estimer la qualité ou les biais des données d'apprentissage. Cette validation peut être obtenue par :

- ✓ le calcul des moyennes et variances des attributs,
- ✓ si possible, le calcul de la corrélation entre certains champs,
- ✓ ou la détermination de la classe majoritaire dans le cas de la classification.

**La validation par expertise :**

est réalisée par un expert du domaine qui jugera la pertinence des résultats produits. Par exemple pour la recherche des règles d'association, c'est l'expert du domaine qui jugera la pertinence des règles.

Pour certains domaines d'application (le diagnostic médical, par exemple), le modèle présenté doit être compréhensible. Une première validation doit être effectuée par un expert qui juge la compréhensibilité du modèle. Cette validation peut être, éventuellement, accompagnée par une technique statistique.

Grâce aux techniques d'extraction de connaissances, les bases de données volumineuses sont devenues des sources riches et fiables pour la génération et la validation de connaissances. La fouille de données n'est qu'une phase du processus d'*ECD*, et consiste à appliquer des algorithmes d'apprentissage sur les données afin d'en extraire des modèles (motifs). L'extraction de connaissances à partir des données se situe à l'intersection de nombreuses discipline [Kod98]**,** comme l'apprentissage automatique, la reconnaissance de formes, les bases de données, les statistiques, la représentation des connaissances, l'intelligence artificielle, les systèmes experts, etc. (Figure 1.2).

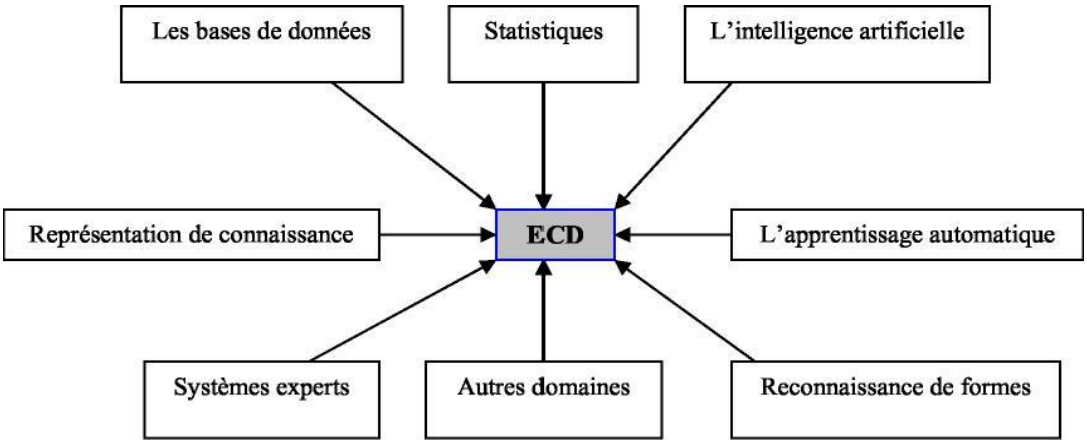



**Figure1. 2 :** L'Extraction de connaissances à partir des données à la confluence de nombreux domaines [Kod98]

## 1.3 Eléments de fouille de données

Les concepts de fouille de données et d'extraction de connaissances à partir de données sont parfois confondus et considérés comme synonymes. Mais, formellement on considère la fouille de données comme une étape centrale du processus d'extraction de connaissances des bases de données *(ECBD* ou *KDD* pour *Knowledge Discovery in Databases* en anglais) [Lie07]**.**

### 1.3.1 Historique

L'expression *"data mining"* est apparue vers le début des années 1960 et avait, à cette époque, un sens péjoratif. En effet, les ordinateurs étaient de plus en plus utilisés pour toutes sortes de calculs qu'il n'était pas envisageable d'effectuer manuellement jusque là. Certains chercheurs ont commencé à traiter sans *a priori* statistique les tableaux de données relatifs à des enquêtes ou des expériences dont ils disposaient. Comme ils constataient que les résultats obtenus, loin d'être aberrants, étaient tout au contraire prometteurs, ils furent incites à systématiser cette approche opportuniste. Les

statisticiens officiels considéraient toutefois cette démarche comme peu scientifique et utilisèrent alors les termes " *data mining "* ou *"data fishing"* pour les critiquer.
Cette attitude opportuniste face aux données coïncida avec la diffusion dans le grand public de l'analyse de données dont les promoteurs, comme Jean-Paul Benzecri [Zig00]**,** ont également dû subir dans les premiers temps les critiques venant des membres de la communauté des statisticiens.
Le succès de cette démarche empirique ne s'est pas démenti malgré tout. L'analyse des données s'est développée et son intérêt grandissait en même temps que la taille des bases de données. Vers la fin des années 1980, des chercheurs en base de données, tel que Rakesh Agrawal [Agr93], ont commencé à travailler sur l'exploitation du contenu des bases de données volumineuses comme par exemple celles des tickets de caisses de grandes surfaces,convaincus de pouvoir valoriser ces masses de données dormantes. Ils utilisèrent l'expression *"database mining"* mais, celle-ci étant déjà déposée par une entreprise (Database mining workstation), ce fut *"data mining"* qui s'imposa. En mars 1989, Shapiro Piatetski [Sha91] proposa le terme *"knowledge discovery"* à l'occasion d'un atelier sur la découverte des connaissances dans les bases de données.
Actuellement, les termes data mining et knowledge discovery in data bases *(KDD*, ou *ECD* en français) sont utilisés plus ou moins indifféremment. Nous emploierons par conséquent l'expression *"data mining"*, celle-ci étant la plus fréquemment employée dans la littérature. La communauté de *"data mining "* a initié sa première conférence en 1995 à la suite de nombreux ateliers (workshops) sur le *KDD* entre 1989 et 1994. La première revue du domaine est de :

- *Data mining and knowledge discovery journal "* publiée par "*Kluwers* " a été lancée en 1997.

**1.3.2Définition**

*«Le data mining, ou fouille de données, est l'ensemble des méthodes et techniques*

*destinées à l'exploration et l'analyse de bases de données informatiques (souvent grandes), de façon automatique ou semi-automatique, en vue de détecter dans ces données des règles, des associations, des tendances inconnues ou cachées, des structures particulières restituant l'essentiel de l'information utile tout en réduisant la quantité de donnée»* [Kan03].

D'après [Had02]**,** la définition la plus communément admise de Data Mining est celle de [Fay98] **:** *«Le Data mining est un processus non trivial qui consiste à identifier, dans des données, des schémas nouveaux, valides, potentiellement utiles et surtout compréhensibles et utilisables».*

En bref, le data mining est l'art d'extraire des informations (ou même des connaissances) à partir des données [Tuf05]**.**

### 1.3.3 Principales tâches de fouille de données

On dispose de données structurées. Les objets sont représentés par des enregistrements (ou descriptions) qui sont constitués d'un ensemble de champs (ou attributs) prenant leurs valeurs dans un domaine. De nombreuses tâches peuvent être associées au Data Mining, parmi elles nous pouvons citer:

#### 1.3.3.1 La classification

Elle consiste à examiner les caractéristiques d'un objet et lui attribuer une classe, la classe est un champ particulier à valeurs discrètes. Des exemples de tâche de classification sont :

- attribuer ou non un prêt à un client,
- établir un diagnostic,
- accepter ou refuser un retrait dans un distributeur,
- attribuer un sujet principal à un article de presse,etc.

#### 1.3.3.2 L'estimation

Elle consiste à estimer la valeur d'un champ à partir des caractéristiques d'un objet. Le champ à estimer est un champ à valeurs continues. L'estimation peut être utilisée dans un but de classification. Il suffit d'attribuer une classe particulière pour un intervalle de valeurs du champ estimé. Des exemples de tâche d'estimation sont :

- Estimer les revenus d'un client.

#### 1.3.3.3 La prédiction

Cela consiste à estimer une valeur future. En général, les valeurs connues sont historisées. On cherche à prédire la valeur future d'un champ. Cette tâche est proche des précédentes. Les méthodes de classification et d'estimation peuvent être utilisées en prédiction. Des exemples de tâches de prédiction sont :

- prédire les valeurs futures d'actions,
- prédire, au vu de leurs actions passées, les départs de clients.

#### 1.3.3.4 Les règles d'association

Cette tâche, plus connue comme *l'analyse du panier de la ménagère,* consiste à déterminer les variables qui sont associées. L'exemple type est la détermination des articles (le pain et le lait, la tomate, les carottes et les oignons) qui se retrouvent ensemble sur un même ticket de supermarché. Cette tâche peut être effectuée pour identifier des opportunités de vente croisée et concevoir des groupements attractifs de produit.

#### 1.3.3.5 La segmentation

Consiste à former des groupes (clusters) homogènes à l'intérieur d'une population. Pour cette tâche, il n'y a pas de classe à expliquer ou de valeur à prédire définie *a priori,* il s'agit de créer des groupes homogènes dans la population (l'ensemble des enregistrements). Il appartient ensuite à un expert du domaine de déterminer l'intérêt et la signification des groupes ainsi constitués. Cette tâche est souvent effectuée avant les

précédentes pour construire des groupes sur lesquels on applique des tâches de classification ou d'estimation.

### 1.3.4 Les méthodes de data Mining

Pour tout jeu de données et un problème spécifique, il existe plusieurs méthodes que l'on choisira en fonction de :

- la tâche à résoudre,
- la nature et de la disponibilité des données,
- l'ensemble des connaissances et des compétences disponibles,
- la finalité du modèle construit,
- l'environnement social, technique, philosophique de l'entreprise ,etc.

On peut dégager deux grandes catégories de méthodes d'analyse consacrées à la fouille de données [Fio06]. La frontière entre les deux peut être définie par la spécificité des techniques, et marque l'aire proprement dite du *"Data Mining"*. On distingue donc :

**Les méthodes classiques**

On y retrouve des outils généralistes de l'informatique ou des mathématiques :

- Les requêtes dans les bases de données, simples ou multi-critères, dont la représentation est une vue,
- les requêtes d'analyse croisée, représentées par des tableaux croisés,
- les différents graphes, graphiques et représentations,
- les statistiques descriptives,
- l'analyse de données : analyse en composantes principales,

**Les méthodes sophistiquées**

Elles ont été élaborées pour résoudre des tâches bien définies. Ce sont :

- Les algorithmes de segmentation,
- les règles d'association,

- les algorithmes de recherche du plus proche voisin,
- les arbres de décision,
- les réseaux de neurones,
- les algorithmes génétiques,

La section suivante n'est pas une présentation exhaustive de l'ensemble des techniques de la fouille de données, mais une présentation de quelques méthodes sophistiquées pour fournir un aperçu du domaine.

#### 1.3.4.1 Segmentation (Clustering)

La segmentation est l'opération qui consiste à regrouper les individus d'une population en un nombre limité de groupes, les segments (ou clusters, ou partitions), qui ont deux propriétés : D'une part, ils ne sont pas prédéfinis, mais découverts automatiquement au cours de l'opération, contrairement aux classes de la classification. D'autre part, les segments regroupent les individus ayant des caractéristiques similaires et séparent les individus ayant des caractéristiques différentes (homogénéité interne et hétérogénéité externe).

La segmentation est une tâche d'apprentissage *"non supervisée"* car on ne dispose d'aucune autre information préalable que la description des exemples. Après application de l'algorithme et donc lorsque les groupes ont été construits, d'autres techniques ou une expertise doivent dégager leur signification et leur éventuel intérêt.

Nous présentons ici la méthode des *k*-moyennes car elle est très simple à mettre en œuvre et très utilisée. Elle comporte de nombreuses variantes et est souvent utilisée en combinaison avec d'autres algorithmes.

#### 1.3.4.1 .1Méthode des k-moyennes

La méthode est basée sur une notion de similarité entre enregistrements. Nous allons pour introduire l'algorithme, considérer un espace géométrique muni d'une distance. Deux points sont similaires s'ils sont proches pour la distance considérée. Pour pouvoir

visualiser le fonctionnement de l'algorithme, nous allons limiter le nombre de champs des enregistrements. Nous nous plaçons donc dans l'espace euclidien de dimension 2 et nous considérons la distance euclidienne classique. L'algorithme suppose choisi *a priori* un nombre *k* de groupes à constituer.

On choisit alors *k* enregistrements, soit *k* points de l'espace appelés centres. On constitue alors les *k* groupes initiaux en affectant chacun des enregistrements dans le groupe correspondant au centre le plus proche. Pour chaque groupe ainsi constitué, on calcule son nouveau centre en effectuant la moyenne des points du groupe et on réitère le procédé. Le critère d'arrêt est la stabilité, par lequel d'une itération à la suivante, aucun point n'a changé de groupe.

### 1.3.4.2 Règles d'association

Les règles d'association sont traditionnellement liées au secteur de la distribution car leur principale application est *"l'analyse du panier de la ménagère (market basket analysis)"* qui consiste en la recherche d'associations entre produits sur les tickets de caisse. Le but de la méthode est l'étude de ce que les clients achètent pour obtenir des informations sur *"qui"* sont les clients et *"pourquoi"* ils font certains achats. La méthode peut être appliquée à tout secteur d'activité pour lequel il est intéressant de rechercher des groupements potentiels de produits ou de services: services bancaires, services de télécommunications, par exemple. Elle peut être également utilisée dans le secteur médical pour la recherche de complications dues à des associations de médicaments ou à la recherche de fraudes en recherchant des associations inhabituelles.

Un attrait principal de la méthode est la clarté des résultats produits. En effet, le résultat de la méthode est un ensemble de *règles d'association.* Des exemples de règles d'association sont :

- Si un client achète des plantes alors il achète du terreau,
- Si un client achète une télévision, il achètera un magnétoscope dans un an.

Ces règles sont intuitivement faciles à interpréter car elles montrent comment des produits ou des services se situent les uns par rapport aux autres. Ces règles sont particulièrement utiles en marketing. Les *règles d'association* produites par la méthode peuvent être facilement utilisées dans le système d'information de l'entreprise. Cependant, il faut noter que la méthode, si elle peut produire des règles intéressantes, peut aussi produire des règles triviales (déjà bien connues des intervenants du domaine) ou inutiles (provenant de particularités de l'ensemble d'apprentissage).

**1.3.4.3 Les plus proches voisins**

La méthode des plus proches voisins *(PPV* en bref, *nearest neighbor* en anglais*)* est une méthode dédiée à la classification qui peut être étendue à des tâches d'estimation. La méthode *PPV* est une méthode de raisonnement à partir de cas. Elle part de l'idée de prendre des décisions en recherchant un ou des cas similaires déjà résolus en mémoire.

Contrairement aux autres méthodes de classification qui seront étudiées dans les sections suivantes (arbres de décision, réseaux de neurones, …), il n'y a pas d'étape d'apprentissage consistant en la construction d'un modèle à partir d'un échantillon d'apprentissage. C'est l'échantillon d'apprentissage, associé à une fonction de distance et d'une fonction de choix de la classe en fonction des classes des voisins les plus proches, qui constitue le modèle.

**1.3.4.4 Les réseaux de neurones**

Les réseaux de neurones sont apparus dans les années cinquante avec les premiers perceptrons, et sont utilisés industriellement depuis les années quatre-vingt. Un réseau de neurone *"ou réseau neuronal"* a une architecture calquée sur celle du cerveau, organisée en neurones et synapses, et se présente comme un ensemble de nœuds "ou neurones formels, ou unités" connectés entre eux, chaque variable prédictive continue correspondant à un nœud d'un premier niveau, appelé *couche d'entrée,* et chaque variable prédictive catégorique (ou chaque modalité d'une variable catégorique) correspondant également à un nœud de la couche d'entrée.

Le cas échéant, lorsque le réseau est utilisé dans une technique prédictive, il y a une ou plusieurs variables à expliquer ; elle correspondant alors chacune à un nœud (ou plusieurs dans le cas des variables catégorielles) d'un dernier niveau : la *couche sortie.* Les réseaux prédictifs sont dits *"à apprentissage supervisé"* et les réseaux descriptifs sont dits *"à apprentissage non supervisé".* Entre la couche d'entrée et la couche sortie sont parfois connectés à des nœuds appartenant à un niveau intermédiaire : *la couche cachée.* Il peut exister plusieurs couches cachées [Tuf02]**.**

- ✓ **Définition**

*«Les réseaux de neurones sont des outils très utilisés pour la classification, l'estimation, la prédiction et la segmentation. Ils sont issus de modèles biologiques, sont constitués d'unités élémentaires (les neurones) organisées selon une architecture»* [Tuf05].

Un nœud reçoit des valeurs en entrée et renvoie 0 à *n* valeurs en sortie. Toutes ces valeurs sont normalisées pour être comprises entre 0 et 1(ou parfois entre -1 et 1), selon les bornes de la fonction de transfert. Une fonction de combinaison calcule une première valeur à partir des nœuds connectés en entrée et poids des connexions. Dans les réseaux les plus courants, les perceptrons, il s'agit de la somme pondérée $\sum n_i p_i$ des valeurs des nœuds en entrée. Afin de déterminer une valeur en sortie, une seconde fonction, appelée *fonction de transfert (ou d'activation),* est appliquée à cette valeur. Les nœuds de la couche d'entrée sont triviaux, dans la mesure où ils ne combinent rien, et ne font que transmettre la valeur de la variable qui leur correspond.

### 1.3.4.5 Les arbres de décision

La méthode des arbres de décision est l'une des plus intuitives et des plus populaires du data mining, d'autant plus qu'elle fournit des règles explicites de classement et supporte bien les données hétérogènes, manquantes et les effets non linéaires. Pour les applications relevant du marketing de bases de données, actuellement la seule grande concurrente de l'arbre de décision est la régression logistique, cette méthode étant préférée dans la prédiction du risque en raison de sa plus grande robustesse.

Remarquons que les arbres de décision sont à la frontière entre les méthodes prédictives et descriptives, puisque leur classement s'opère en segmentant la population à laquelle ils s'appliquent ,ils ressortissent donc à la catégorie des classifications hiérarchiques descendantes supervisées.

- **Description de l'algorithme**

La technique de l'arbre de décision est employée en classement pour détecter des critères permettant de répartir les individus d'une population en *n* classes (souvent *n=2*) prédéfinies. On commence par choisir la variable qui, par ses modalités, sépare le mieux les individus de chaque classe, de façon à avoir des sous-populations, que l'on appelle nœuds, contenant chacune le plus possible d'individus d'une seule classe, puis on réitère la même opération sur chaque nouveau nœud obtenu jusqu'à ce que la séparation des individus ne soit plus possible ou plus souhaitable.

Par construction, les nœuds terminaux (les feuilles) sont tous majoritairement constitués d'individus d'une seule classe avec une assez forte probabilité, quand il satisfait l'ensemble des règles permettant d'arriver à cette feuille. L'ensemble des règles de toutes les feuilles constitue le modèle de classement (Figure 1.3).

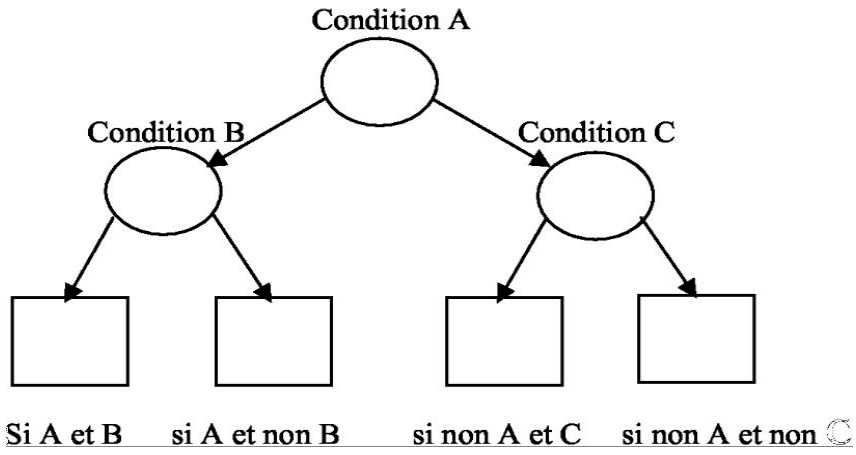



**Figure1. 3 :** Arbre de décision.

L'algorithme d'apprentissage par l'arbre de décision est décrite comme suit :

---

**Algorithme d'apprentissage par arbre de décision**

---

**Donnée :** un échantillon S de *m* enregistrements classés *(x, c(x))*

**Initialisation :** *A* ← arbre vide

noeud_courant ← racine

échantillon_courant ← *S*

**Répéter**

Décider si le noeud courant est terminal

**Si (** noeud_courant est terminal ) **alors**

Étiqueter le noeud courant par une feuille

**Sinon**

Sélectionner un test :

Créer les fils

Définir les échantillons sortants du noeud

**Fin si**

noeud_courant ← un noeud non encore étudié de *A*

échantillon_courant : échantillon atteignant noeud_courant

**Jusque** (noeud_courant = Ø)

élaguer l'arbre de décision *A* obtenu

**Sortie :** l'arbre *A* élagué

---

- **Critiques de la méthode**
  - ***Avantages***
- *Adaptabilité aux attributs de valeurs manquantes :* les algorithmes peuvent traiter les valeurs manquantes (descriptions contenant des champs non renseignés) pour

l'apprentissage, mais aussi pour la classification.

- *Bonne lisibilité du résultat :* un arbre de décision est facile à interpréter et à la représentation graphique d'un ensemble de règles. Si la taille de l'arbre est importante, il est difficile d'appréhender l'arbre dans sa globalité. Cependant, les outils actuels permettent une navigation aisée dans l'arbre (parcourir une branche, développer un noeud, élaguer une branche) et, le plus important, est certainement de pouvoir expliquer comment est classé un exemple par l'arbre, ce qui peut être fait en montrant le chemin de la racine à la feuille pour l'exemple courant.
- *Traitement de tout type de données :* l'algorithme peut prendre en compte tous les types d'attributs et les valeurs manquantes. Il est robuste au bruit.
- *Sélectionne des variables pertinentes :* l'arbre contient les attributs utiles pour la classification. L'algorithme peut donc être utilisé comme pré-traitement qui permet de sélectionner l'ensemble des variables pertinentes pour ensuite appliquer une autre méthode.
- *Donne une classification efficace :* l'attribution d'une classe à un exemple à l'aide d'un arbre de décision est un processus très efficace (parcours d'un chemin dans un arbre).
- *Disponibilité des outils :* les algorithmes de génération d'arbres de décision sont disponibles dans tous les environnements de fouille de données.
- *Méthode extensible et modifiable :* la méthode peut être adaptée pour résoudre des tâches d'estimation et de prédiction. Des améliorations des performances des algorithmes de base sont possibles grâce aux techniques qui génèrent un ensemble d'arbres votant pour attribuer la classe.

o ***Inconvénients***

- *Méthode sensible au nombre de classes :* les performances tendent à se dégrader lorsque le nombre de classes devient trop important.
- *Manque d'évolutivité dans le temps :* l'algorithme n'est pas incrémental, c'est-à-

dire, que si les données évoluent avec le temps, il est nécessaire de relancer une phase d'apprentissage sur l'échantillon complet (anciens exemples et nouveaux exemples).

## 1.4 Méta heuristiques pour l'extraction de connaissances

Les méta heuristiques sont des méthodes qui permettent de concevoir des algorithmes pour la résolution des problèmes d'optimisation auxquels les ingénieurs et les décideurs sont régulièrement confrontés. La majorité des problèmes d'extraction de connaissances peuvent s'exprimer des problèmes d'optimisation combinatoire. Or, de nombreux problèmes d'optimisation combinatoire sont NP-difficilles et ne pourront donc pas être résolus de manier exacte dans un temps raisonnable puisque la capacité de calcul des machines évolue linéairement alors que le temps nécessaire à la résolution de ces problèmes évolue exponentiellement. Lorsqu'on attaque à des problèmes réels, il faut se résoudre à un compromis entre la qualité des solutions obtenues et le temps de calcul utilisé.

Ces méthodes sont souvent inspirées par des systèmes naturels, qu'ils soient pris en physique (cas de recuit simulé), en biologie de l'évolution (cas des algorithmes génétiques) ou encore en éthologie (cas des algorithmes de colonies de fourmis ou de l'optimisation par essaims particulaires).

Selon [Jou03] les méta heuristiques peuvent être classées en deux groupes : les méthodes à solution unique et les méthodes à population de solutions.

**-Les méta heuristiques à solution unique**

Les méthodes itératives à solution unique sont touts basées sur un algorithme de recherche de voisinage qui commence par une solution initiale, puis l'améliore pas à pas en choisissant une nouvelle solution dans son voisinage [Bac99]. Les méthodes les plus utilisées pour l'extraction de connaissances sont : la méthode de descente, le recuit simulé et la méthode tabou.

**- Les métathéoriques à population de solutions**

Les méthodes d'optimisation à population de solutions améliorent, au fur et à mesure des itérations, une population de solutions. L'intérêt de ces méthodes est d'utiliser la population comme facteur de diversité. Les méthodes les plus utilisées sont : la recherche par dispersion (Scatter Search), les algorithmes génétiques, la programmation génétique, les algorithmes à essaim de particules, les systèmes immunitaires artificiels, les algorithmes à estimation de distribution et les colonies de fourmis.

## 1.5 Définition d'un Arbre de décision [ALAIN,2007]

Les arbres de décision sont la modélisation d'une classification. Ils apprennent à partir d'observations qu'on appelle des exemples. Un exemple est représenté par une série d'attributs et une classe associée, on doit connaître la classe parce que les arbres de décision travaillent sur la classification en mode supervisée Les arbres de décision sont un bon moyen d'illustrer le raisonnement pour distinguer les similitudes et les différences entre les attributs des exemples du jeu de données, ils sont souvent utilisés par les statisticiens pour illustrer le résultat d'une analyse.

Un arbre de décision est composé de nœuds en arborescence, le nœud à base de l'arbre est appelé la racine, chacun des nœuds sous la racine est soit une feuille ou un sous-arbre.

Dans la figure I.4, les nœuds B, D et E sont des nœuds terminaux et le nœud C est un sous arbre du nœud A.

### 1.5.1 Vocabulaire des arbres (Arbre, nœud, racine, feuille) [Bertrand, 2008]

- Un arbre est constitué de **noeuds** connectés entre eux par des **branches**.
- Un arbre de décision est constitué de **noeuds de décision**.
- Une branche entre deux noeuds est orientée : l'un des noeuds de la connexion est dit «**nœud parent** », et l'autre « **nœud enfant** ».
- Chaque nœud est connecté à un et **un seul nœud parent**, sauf **le**

**nœud racine** qui n'a pas de parent.

- Chaque nœud peut être connecté à **0 ou n noeuds enfants**.
- Les deux caractéristiques précédentes font qu'**un arbre n'est pas un réseau** (ou graphe).
- Un nœud qui n'a pas de parents est appelé « **nœud racine** » ou « racine ».
- Un nœud qui n'a pas de noeuds enfants est appelé « **nœud feuille** » ou « feuille ».

Dans l'arbre de décision de la figure I.5, les attributs A et B ont chacun deux valeurs distinctes, lorsque les exemples avec l'attribut A égal à a], ils correspondent à une seule classe. Dans le cas, où l'attribut A est égal à a2, les exemples correspondent à deux classes différentes, on a besoin alors de prendre l'attribut B pour diviser les exemples dans leurs classes respectives.

### 1.5.2 Structure interne d'un nœud d'un arbre de décision [Alain, 2007] :

Un nœud est soit un nœud terminal ou un sous arbre, un nœud terminal est un nœud avec une décision ou une classe. Un sous arbre est un nœud qui possède des descendants.

**Un nœud contient nécessairement les informations suivantes:**

- **Une étiquette:** C'est le nom du nœud dans l'arbre, il représente soit un attribut, si le nœud a des descendants ou une classe, si le nœud est terminal.
- **La colonne référence au jeu d'apprentissages:** Cette valeur est utilisée pour indiquer aux règles, la position de l'attribut dans le jeu d'apprentissages pour l'évaluation.
- **Le tableau des branches:** C'est le tableau qui contient le nom des branches. Dans un arbre de décision, les branches représentent les valeurs des attributs choisis pour le nœud.

- **Le tableau d'enfants:** Ce tableau contient d'autres noeuds de niveau inférieur.

**Les informations suivantes sont complémentaires:**

**La classe majoritaire (CART) :** C'est la classe la plus représentative dans le jeu d'apprentissages. Lors de l'élagage du nœud, cette information servira à remplacer le nom de la feuille.

- **Le nombre d'exemples classifiés par ce nœud**: C'est le nombre d'exemples dans le jeu d'apprentissages qui sont classifiés au niveau du nœud.
- **Le nombre d'erreurs de classification:** C'est le nombre d'exemples qui n'appartient pas à la classe majoritaire. On l'appelle aussi l'erreur apparente de l'arbre.
- **L'estimation du taux d'erreur réel (C4.5) :** Cette erreur est calculée en utilisant l'erreur apparente, elle est utilisée dans l'élagage de l'arbre, c'est la valeur qui détermine si le nœud sera élagué ou non.
- **Le tableau des classes**: C'est l'ensemble des classes qui font parti du jeu d'apprentissage au niveau du nœud.
- **Le tableau des cardinalités des classes:** Ce tableau contient les cardinalités de chacune des classes.
- **Jeu d'apprentissages et tableau des attributs non utilisés:** Le jeu d'apprentissages au niveau du nœud contient les exemples classés par le nœud et le tableau des attributs non utilisés contient les indexes restant du processus de construction de l'arbre, on garde ces informations pour pouvoir aller rechercher de l'information sur les similarités entre les exemples classés au niveau de ce nœud.

### 1.5.3Construction d'un arbre de décision [Alain, 2007]

Les arbres de décision sont construits à partir d'un jeu d'apprentissage, un jeu

d'apprentissage est une matrice, où les lignes représentent les exemples et les colonnes représentent les caractéristiques des exemples, la dernière colonne est réservée aux classes associées aux exemples. L'algorithme de construction a aussi besoin d'un tableau d'index qui constitue la liste de référence des attributs à traiter.

### 1.5.4les étapes de construction d'arbre de décision

L'algorithme de construction d'arbre de décision se divise en 3 étapes.

- **Etape 1 :** consiste à vérifier si on doit faire un nœud terminal pour représenter les exemples du jeu d'apprentissage. Pour faire un nœud terminal, on doit respecter une des conditions suivantes: Tous les exemples du jeu d'apprentissage appartiennent à la même classe ou tous les attributs ont été utilisés pour les noeuds précédents. Cette, étape permet d'arrêter l'expansion de la branche de l'arbre.
  - La deuxième et la troisième étape se produisent lorsqu'on ne respecte pas les critères de la première.
- **Etape 2 :** consiste à trouver l'attribut pour représenter le nœud de l'arbre. Les algorithmes de construction d'arbre de décision utilisent une mesure de segmentation par rapport aux attributs à traiter. Nous allons voir en détail les différentes techniques plus tard.
- **Etape 3 :** étape consiste à éclater le jeu d'apprentissages pour créer les branches du nœud, chacune des branches du nœud prend une des différentes valeurs que l'attribut du nœud peut prendre. Pour chacune des branches qu'on aura créées, il faut recommencer le processus en prenant les exemples correspondants à la branche..

### 1.5.5Mesure de segmentation

La mesure de segmentation est l'heuristique qui permet de choisir l'attribut qui permettra de répartir le mieux le jeu d'apprentissages. Cette mesure est souvent

une mesure statistique. L'objectif principal est de construire des arbres de décision relativement simple. On recherche un arbre petit et simple plutôt qu'un arbre grand qui est complexe.

Le choix des attributs à tester est une étape cruciale pour la construction d'un arbre. Pour cela, la mesure de segmentation doit évaluer toutes les possibilités de choix pour chacun des niveaux d'un arbre de décision.

**1)Gain informationnel :**

Le gain informationnel est une mesure de segmentation qui utilise l'entropie de Shannon. ID3 et C4.5 utilisent le gain pour choisir l'attribut pour représenter le nœud. Il conserve seulement les informations absolument nécessaires pour classer un objet. À chaque fois, qu'on doit choisir un attribut pour partitionner l'ensemble d'exemples, il faut choisir celui dont l'entropie de classification est la plus petite. En général, le gain privilégie généralement les attributs ayant un grand nombre de valeurs.

Le gain informationnel (voir l'équation 2.2) est la différence entre la répartition des classes par rapport au jeu d'apprentissage et la répartition des valeurs des attributs par rapport aux classes.

$$Info(S) = -\sum_{j=1}^{k} \frac{freq(C_j, S)}{|S|} * \log_2 \left( \frac{freq(C_j, S)}{|S|} \right) \qquad (2.1)$$

$$Gain(X) = Info(T) - \sum_{i=1}^{nbTest} \frac{|T_i|}{|T|} * Info(T_i) \qquad (2.2)$$

La fonction *freq (Cj, S)* trouve la fréquence des exemples qui correspondent à la classe Cj dans le jeu d'apprentissage S, ITI représente le nombre d'exemples à évaluer, *nbTest* est le nombre de valeurs pour l'attribut testé, ITi I est le nombre d'exemples qui correspond à la valeur i de l'attribut testé.

## 2)Ratio de gain

C4.5 utilise une notion complémentaire au gain informationnel qu'on appelle le ratio de gain. Il est utilisé pour pondérer le gain qui favorise les attributs qui ont beaucoup de valeurs. On calcule toujours le gain informationnel, cependant on calcule aussi la répartition des valeurs de l'attribut par rapport au jeu d'apprentissage.

Ce facteur permet d'éviter de tomber dans le sur apprentissage. Le Split Info représente l'information potentielle générée en partitionnant le jeu d'apprentissage T en n sous- ensembles, elle montre la proportion de l'information générée par l'éclatement par un attribut

$$SplitInfo(X) = -\sum_{i=1}^{n} \frac{|T_i|}{|T|} * \log_2\left(\frac{|T_i|}{|T|}\right) \quad (2.3)$$

Le ratio de gain sélectionne le test de façon à optimiser le ratio, on prend toujours en compte du gain informationnel, mais on tient compte de la répartition des valeurs des attributs pour choisir l'attribut pour partitionner le jeu d'apprentissage

$$GainRatio(X) = \frac{Gain(X)}{SplitInfo(X)} \quad (2.4)$$

## 3)Problème de la construction d'un arbre : la scission

Deux problèmes vont intervenir pour construire l'arbre de décision :

- Le problème du nœud : quelle variable choisit-on à chaque nœud ?
- Le problème de la branche : quelles branches définit-on sous chaque

nœud. Autrement dit quelles catégories choisit-on pour les prédicateurs ?

Ces deux problèmes seront finalement liés : c'est **le problème de la scission.**

### 1.5.6les algorithmes de construction d'arbre de décision

Il existe plusieurs algorithmes de construction d'arbre, les plus populaires sont :

| *Nom de l'algorithme* | *Développeur* | *Année* |
|---|---|---|
| CHAID | Kass | 1980 |
| CART | Breiman, et al. | 1984 |
| ID3 | Quinlan | 1986 |
| C4.5 | Quinlan | 1993 |
| SLIQ | Agrawal, et al. | 1996 |
| SPRINT | Agrawal, et al. | 1996 |

**Figure1. 4 :** Les algorithmes d'inductions des arbres de décision [Chami ,2010]

### 1.5.6.1L'algorithme de segmentation et de régression : CART [Bertrand, 2008]

**--Principe**

Le CART est un algorithme qui utilise un arbre binaire. Pour chaque nœud, on choisit la variable qui, par ses catégories (ses classes de valeurs), sépare le mieux les individus en fonction des catégories de la variable cible. Le choix du nœud est fonction du choix des branches du nœud.

**Algorithme**

**Début**

L'algorithme part de la racine de l'arbre. Boucle de parcours de l'arbre

À chaque nœud de décision, l'algorithme fait une recherche exhaustive sur toutes les catégories de toutes les variables et mesure à chaque fois la **valeur de la scission** obtenu.

L'algorithme choisit la scission optimale.

Il n'y a qu'une scission par nœud puisque l'arbre est binaire.

Fin de boucle

**Fin**

- TECHNIQUE

$$\varphi(s \mid t) = 2P_G P_D \sum(i=1 \text{ à } nbClasses) \mid P(i \mid t_G) - P(i \mid t_D) \mid$$

**Mesure de la qualité d'une scission**

**φ (s | t):** mesure de la qualité d'une scission au nœud t La meilleure scission parmi toutes les scissions possibles au nœud t est celle qui a la plus grande valeur pour **φ (s | t):**

- **PG :** (nb enregistrements à tG) / nb Total
- **PD :** (nb enregistrements à tD) / nb Total
- **tG :** nœud enfant gauche du nœud t
- **tD :** nœud enfant droit du nœud t
- **nb Total :** nombre d'enregistrements dans tout l'ensemble d'apprentissage
- **nb Classes :** nombre de catégories de la variable cible
- **P (i | tG):** (nb enregistrements pour la classe i à tG) / (nb enregistrements à t)
- **P (i | tD):** (nb enregistrements pour la classe i à tD) / (nb

enregistrements à t) Le nombre d'enregistrement sur un nœud correspond au nombre d'enregistrements restants après les décisions déjà prises.

### 1.5.6.2construction d'un arbre de décision par l'algorithme ID3 [Preux ,2011]

- **PRINCIPE**

ID3 de R. Quillan propose en 1986 qui a été raffine par la suite (C4.5 puis C5) Le principe de l'algorithme ID3 pour déterminer l'attribut à placer a la racine de l'arbre de décision : rechercher l'attribut qui possède le gain d'information maximum, le placer en racine, et itérer pour chaque fils, c'est-a-dire pour chaque valeur de l'attribut

- **CONSTRUCTION**

| Jour | Ciel | Température | Humidité | Vent | Jouer au tennis ? |
|---|---|---|---|---|---|
| 1 | Ensoleillé | Chaude | Élevée | Faible | Non |
| 2 | Ensoleillé | Chaude | Élevée | Fort | Non |
| 3 | Couvert | Chaude | Élevée | Faible | Oui |
| 4 | Pluie | Tiède | Élevée | Faible | Oui |
| 5 | Pluie | Fraîche | Normale | Faible | Oui |
| 6 | Pluie | Fraîche | Normale | Fort | Non |
| 7 | Couvert | Fraîche | Normale | Fort | Oui |
| 8 | Ensoleillé | Tiède | Élevée | Faible | Non |
| 9 | Ensoleillé | Fraîche | Normale | Faible | Oui |
| 10 | Pluie | Tiède | Normale | Faible | Oui |
| 11 | Ensoleillé | Tiède | Normale | Fort | Oui |
| 12 | Couvert | Tiède | Élevée | Fort | Oui |
| 13 | Couvert | Chaud | Normale | Faible | Oui |
| 14 | Pluie | Tiède | Élevée | Fort | Non |

**Figure 1.5** : Jeu de données « jouer au tennis

Sur un exemple, on montre la construction d'un arbre de décision par ID3. On considère l'ensemble exemples de le tableau I.4 L'attribut cible est donc Jouer au tennis ? Déroulons ID3:

1. création d'une racine
2. les exemples n'étant ni tous positifs, ni tous négatifs, l'ensemble des attributs n'étant pas vide, on calcule les gains d'information pour chaque attribut :

- ✓ LE GAIN INFORMATIQUE :

Pour une classe prenant n valeurs distinctes (numérotées de 1 a n), notons pi tq i ∈ [1;n] la proportion d'exemples dont la valeur de cet attribut est dans l'ensemble d'exemples considère

X. L'entropie de l'ensemble d'exemples X est

$$H(\mathcal{X}) = -\sum_{i=1}^{i=n} p_i \log_2 p_i$$

Soit une population d'exemples X. Le gain d'information de X par rapport a un attribut aj donne est la variation d'entropie causée par la partition de X selon aj :

$$Gain\ (\mathcal{X}, a_j) = H(\mathcal{X}) - \sum_{v \in\ valeurs(a_j)} \frac{|\mathcal{X}_{a_j=v}|}{|\mathcal{X}|}\ H(\mathcal{X}_{a_j=v})$$

Où Xaj=v ⊂ X est l'ensemble des exemples dont l'attribut considère aj prend la valeur v, et la notation indique le cardinal de l'ensemble X.

| Attribut | Gain |
|---|---|
| Ciel | 0, 246 |
| Humidité | 0, 151 |
| Vent | 0, 048 |
| Température | 0, 029 |

Donc, la racine de l'arbre de décision testera l'attribut « Ciel »

3. l'attribut « Ciel » peut prendre trois valeurs. Pour « Ensoleille », ID3 est appelé récursivement avec 5 exemples : {x1; x2; x8; x9; x11}. Les gains d'information des 3 attributs restants sont alors :

- Algorithme 1 ID3

**Nécessite:** 2 paramètres : l'ensemble d'exemples $\mathcal{X}$, l'ensemble d'attributs $\mathcal{A} = \{a_{j\in\{1,\dots p\}}\}$ où $p$ est le nombre d'attributs restants à considérer
Créer un nœud racine
**si** tous les élements de $\mathcal{X}$ sont positifs **alors**
racine. étiquette $\leftarrow \oplus$
return racine
**fin si**
**si** tous les éléments de $\mathcal{X}$ sont négatifs **alors**
racine. étiquette $\leftarrow \ominus$
return racine
**fin si**
**si** $\mathcal{A} = \emptyset$ **alors**
racine. étiquette $\leftarrow$ valeur la plus présente de la classe parmi les $\mathcal{X}$
return racine
**fin si**
$a^* \leftarrow \arg\max_{a\in\mathcal{A}} \text{gain}(\mathcal{X}, a)$
racine. étiquette $\leftarrow a^*$
**pour** toutes les valeurs $v_i$ de $a^*$ **faire**
ajouter une branche à racine correspondant à la valeur $v_i$
former $\mathcal{X}_{a^*=v_i} \subset \mathcal{X}$ dont l'attribut $a^*$ vaut $v_i$
**si** $\mathcal{X}_{a^*=v_i} = \emptyset$ **alors**
à l'extrémité de cette branche, mettre une feuille étiquetée avec la valeur la plus présente de la classe parmi les $\mathcal{X}$
**sinon**
à l'extrémité de cette branche, mettre ID3 $(\mathcal{X}_{a^*=v_i}, \mathcal{A} - \{a^*\})$
**fin si**
**fin pour**
return racine

4. pour la branche _ Pluie _ partant de la racine, ID3 est appelé récursivement avec 5 exemples : {x4;x5; x6; x10; x14} ; on continue la construction de l'arbre de décision récursivement ;
5. pour la branche « Couvert » partant de la racine, ID3 est appelé récursivement avec 4 exemples : {x3; x7; x12; x13} dans ce dernier cas, tous les exemples sont positifs : on affecte donc tout de suite la classe « oui » a cette feuille.

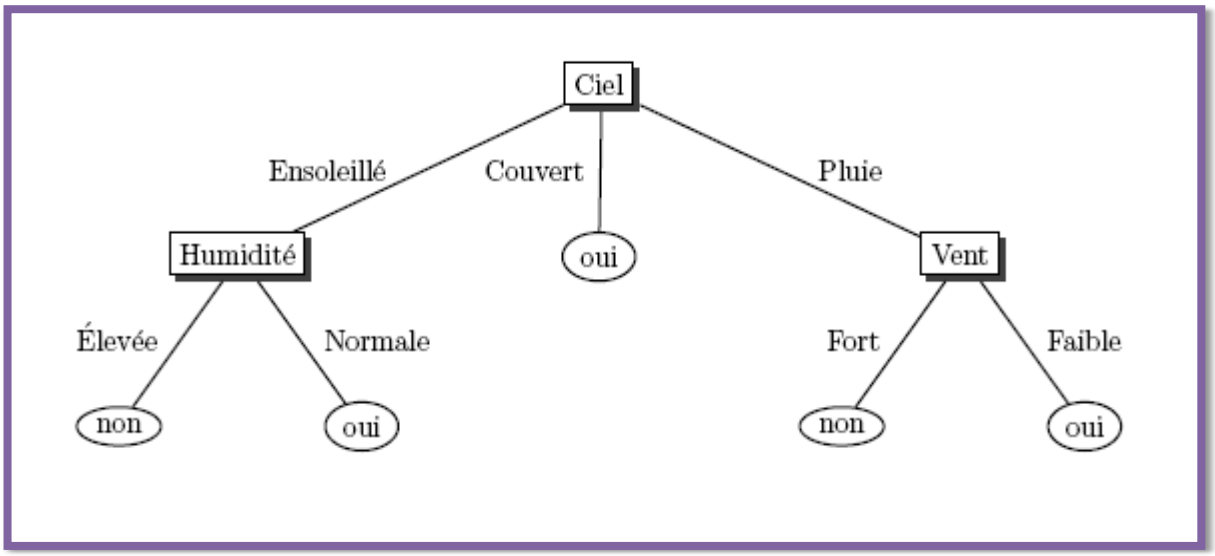



**Figure 1.6:** Arbre de décision obtenu pour l'exemple du texte « jouer au tennis ? »

« Ciel » vaut « Ensoleille », l'attribut « Vent » n'est pas pertinent si l'attribut « Ciel » vaut « Pluie », c'est l'attribut « Humidité » qui ne l'est pas.

**1.5.7Le but des algorithmes de construction d'arbre de décision.**[Alain, 2007]

les algorithmes de construction d'arbre de décision permettent de créer des arbres de décision avec une taille la plus petite que possible, et ce, de façon à créer des règles de décision simples. Plus un arbre de décision est grand, plus les règles sont complexes. Les algorithmes de construction d'arbres choisissent

les attributs toujours par rapport aux classes.

### 1.5.7 Les avantages et les inconvénients des arbres de décision

#### 1.5.7.1 Les avantages des arbres de décision sont [Mitskos et al ,2010]:

- La compréhension des résultats est facilitée car ils sont exprimés sous forme de conditions explicites.
- La technique des arbres de décision est non-paramétrique, ce qui signifie qu'elle ne suppose pas que les variables explicatives suivent des lois probabilistes particulières.
- Les arbres sont peu perturbés par la présence d'individus hors norme, qui peuvent être isolés

#### 1.5.7.2Les inconvénient arbres de décisions [Mitskos ,2010]:

- Manque de performance s'il y a beaucoup de classes; les arbres peuvent être très complexes et ne sont pas nécessairement optimaux.
- Sensibilité au manque de données; les arbres peuvent trop coller aux données (*data overfitting*) lorsque celles-ci ne sont pas assez représentatives et l'élagage peu ne pas entièrement résoudre le problème.
- Moins bonnes performances pour les prédictions portant sur des valeurs numériques.
- La construction et élagage des arbres de décisions nécessitent généralement beaucoup de temps calcul: le choix du meilleur partitionnement lors de la construction et la comparaison de sous-arbres lors de l'élagage est souvent coûteux.
- La combinaison de différents attributs pour les tests n'est pas toujours bien traitée automatiquement; la plupart des algorithmes ne traitent que d'un attribut à la fois.

## 1.8Conclusion

Dans ce premier chapitre, nous avons présenté les principaux concepts de fouille de données, les processus, les tâches et les méthodes les plus utilisés en data mining ainsi que les avantages et les inconvénients de chaque méthode. et intéressons aux techniques D'ARBRE DE DECISION.

# *Chapitre 2*

# *Ontologies Développées Dans Le Domaine d'Arbre De Décision et Web Sémantique*

## 2. 1 Introduction

L'exploitation de connaissances en informatique a pour objectif de ne plus faire manipuler en aveugle des informations à la machine mais de permettre un dialogue (une coopération) entre le système et les utilisateurs. Alors, le système doit avoir accès non seulement aux termes utilisés par l'être humain mais aussi à la sémantique qui leur est associée, afin qu'une communication efficace soit possible. Actuellement, la connaissance visée par ces ontologies est un sujet de recherche populaire dans diverses communautés (l'ingénierie des connaissances, la recherche d'information, le traitement du langage naturel, les systèmes d'information coopératifs, l'intégration intelligente d'information et la gestion des connaissances). Elles offrent une connaissance partagée sur un domaine qui peut être échangée entre des personnes et des systèmes hétérogènes. Elles ont été définies en intelligence artificielle afin de faciliter le partage des connaissances et leur réutilisation. La définition explicite du concept ontologie soulève un questionnement qui est tout à la fois d'ordre philosophique, épistémologique, cognitif et technique.

Avant de définir notre système, nous allons essayer, dans ce chapitre d'étudier différents systèmes reposant sur l'utilisation de l'ontologie dans le domaine de Data mining.

## 2.Ontologies

### 2.2.1 Définitions de l'ontologie

Ontologie est une branche de la métaphysique qui s'intéresse à l'existence, à

l'être en tant qu'être et aux catégories fondamentales de l'existant. En effet, ce terme est construit à partir des racines grecques « ontos » qui veut dire ce qui existe, l'être, l'existant, et « logos » qui veut dire l'étude, le discours, d'où sa traduction par « l'étude de l'être » et par extension de l'existence.

Dans la philosophie classique, l'ontologie correspond à ce qu'Aristote appelait la Philosophie première (porté philosopha), c'est-à-dire la science de l'être en tant qu'être, par opposition aux philosophies secondes qui s'intéressaient, elles, à l'étude des manifestations de l'être (les existants).

En informatique, plusieurs définitions ont été données à l'ontologie :

- Pour **Gruber1993** :

« *Ontology is an explicit specification of a conceptualization* » [Gruber, 1993]

« Une ontologie est une spécification explicite d'une conceptualisation ». [Bernard ,2010]

- Pour **Guarino 1995 :**

« En Intelligent Artificiel, une ontologie représente un artefact d'ingénierie, constitué par un vocabulaire spécifique utilisé pour décrire une certaine réalité, accompagné d'un ensemble d'hypothèses implicites concernant la signification des mots de ce vocabulaire » [Bernard ,2010]

- Pour **Uschold et Gruninger** 1996:

« *Ontology is a shared understanding of some domain of interest* » [Uschold et al., 1996]

« Une ontologie est une compréhension partagée d'un domaine d'intérêt » [Bernard ,2010].

## 2.2.2Utilisation des ontologies [Bouarroudj ,2010].

Même si le besoin de développer une ontologie est très varié et dépend du

domaine d'application, nous pouvons facilement énumérer un certain nombre d'utilités, notamment:

- **La connaissance du domaine :** Les ontologies permettent la modélisation des connaissances dans un domaine particulier, dans lequel opère le système à développer.
- **La communication:** les ontologies assurent une communication fiable et hétérogène entre personnes et machines (agents logiciels ou organisations) du fait qu'elle permet de mettre en place un langage ou un vocabulaire conceptuel commun.
- **L'interopérabilité :** La représentation explicite des connaissances dans un domaine donné sous forme d'une ontologie, permet à son tour une plus grande réutilisation, un partage plus large et une interopérabilité plus étendue.
- **L'aide à la spécification des systèmes:** La représentation conceptuelle des éléments du domaine, permet aux systèmes de réaliser des raisonnements logiques qu'on appelle inférences, et de sortir avec des conclusions capables d'aider l'utilisateur ou le gestionnaire dans ses décisions.
- **L'indexation et la recherche d'information:** Dans le web sémantique, d'une façon générale, et dans notre application en particulier, les ontologies sont utilisées pour indexer et décrire les ressources utilisées. Cela permet une plus grande précision dans les résultats des recherches ou d'assignation des ressources

### 2.2.3 Les composants d'une ontologie

Une ontologie peut être vue comme un ensemble structurée de concepts et de relations entre ces concepts destinés à représenter les objets du monde sous une forme

compréhensible aussi bien par les hommes que par les machines. Les composants d'une ontologie sont :

**1)Concept: ou classe,**

Définissant un ensemble d'objet, abstrait ou concret, que l'on souhaite modéliser pour un domaine donné. Les connaissances portent sur des objets auxquels on se réfère à travers des concepts. Un concept peut représenter un objet matériel, une notion, une idée. Les concepts dans l'ontologie sont habituellement organisés dans des taxonomies [Bouarroudj ,2010].

Our progressivement un concept se définit par Bachimont à trois niveaux [Bachimont 2004]. Un concept est une signification. Sa place dans un système de significations permet de le comprendre, de le distinguer et de le différencier par rapport à d'autres concepts. Un concept est une construction. Comprendre un concept revient à construire l'objet dont il est le concept. Un concept est une prescription. On le comprend en exécutant l'action qu'il entreprend [Nathalie ,2008]

**2)Les instances: ou individus,**

Constituent la définition extensionnelle de l'ontologie (pour représenter les éléments spécifiques) [Bouarroudj ,2010].

**3)les relations**:

Une relation permet de lier des instances de concepts ou des concepts génériques. Elles sont caractérisées par un terme ou plusieurs, et une signature qui précise le nombre d'instances de concepts que la relation lie, leurs types et l'ordre des concepts, c'est – à – dire la façon dont la relation doit être lue[ Bouarroud ,2010].

Une relation sémantique R représente un type d'interaction entre les concepts d'un domaine $c_1$, $c_2$, .., $c_n$. Elle se définit formellement à partir d'un produit de n concepts : R : $c_1 \times c_2 \times$ …$\times c_n$ ; « subsume », « est un phénomène lié à » sont des exemples de relations binaires.

Les relations les plus courantes dans la littérature sont les relations d'équivalence, taxonomiques, patronymiques, de dépendance, topologique, causale, fonctionnelle, chronologique [Nathalie ,2008]

- **Relation taxonomique (ou subsomption) :**

La notion de subsomption (aussi appelée relation « est un », relation taxonomique ou relation de spécificité/généricité) est une relation binaire particulière qui implique l'engagement sémantique suivant [Guarino 2001] : un concept c1 subsume un concept c2 si toute relation sémantique de c1 est aussi relation sémantique de c2, en d'autres termes si le concept c2 est plus spécifique que le concept c1. Les instances se rapportant au concept c2 seront des instances de c1, par contre une partie seulement des instances de c1 seront des instances de c2. La notion abordée par le concept c2 (intention du concept) sera plus précise que celle abordée par c1. La relation de subsomption permet d'organiser hiérarchiquement un ensemble de concepts.*)*

La relation de subsomption est une relation d'ordre partiel définie à partir des propriétés suivantes:

- ✓ **L'asymétrie :** cette propriété signifie que l'inclusion d'une classe d'individus X dans une classe d'individus Y implique que Y n'est pas incluse dans X. Formellement, cette propriété garantit que : X subsume Y, si et seulement si non (Y subsume X),

- ✓ **La transitivité :** soit une classe d'individus X qui subsume une classe Y, qui elle- même subsume Z, alors X subsume Z.

  Formellement : (X subsume Y) et (Y subsume Z) ⇒ (X subsume Z)
- ✓ **La non réflexivité :** Cette propriété implique qu'un fait décrit par la relation « est un » ne peut pas s'écrire de plusieurs façons.

  Formellement : non (X subsume X)

L'**héritage multiple** est une propriété qui peut être définie sur la relation de subsomption : un concept d'une ontologie peut avoir plusieurs pères par la relation de subsomption. L'héritage multiple implique que le concept hérite des propriétés de tous ses pères.

- **Relation associative**

Les relations « associatives » sont des relations d'interaction entre deux concepts qui ne sont pas la relation de subsomption. La désignation « relation associative » est empruntée aux domaines de la bio-informatique [Zhang 2004], ce domaine ayant une utilisation équivalente des ontologies par l'indexation de publications et de comptes rendus biologiques. Elles correspondent à la notion de rôle en Logique de Description et permettent de typer les concepts reliés.

Ces relations sont soit à des propriétés entre concepts soit à des propriétés d'attribut dans le cas où elles associent un concept à un type de données. La sémantique qui leur est associé est référencée par un label. Elle peut également être précisée à partir de propriétés logiques associées à la relation telles que la transitivité, la symétrie, la fonctionnalité.

**4)Les axiomes :**

Une ontologie est en outre composée d'axiomes qui forment des contraintes sémantiques pour le raisonnement et donnent un acompte d'une conceptualisation. Ils prennent la forme d'une théorie logique [Bouarroud ,2010].

Les axiomes ont pour but de définir dans un langage logique la description des concepts et des relations permettant de représenter leur sémantique. Ils représentent les intentions des concepts et des relations du domaine et, de manière générale, les connaissances n'ayant pas un caractère strictement terminologique. Les axiomes sont des expressions qui sont toujours vraies. Leur inclusion dans une ontologie peut avoir plusieurs objectifs : définir la signification des composants, définir des restrictions sur la valeur des attributs, définir les arguments d'une relation, vérifier la validité des informations spécifiées ou en déduire de nouvelles. [Nathalie ,2008]

### 2.2.5Types d'ontologies :

Un critère pour la classification des ontologies est le contenu de la connaissance qu'elles représentent, c'est-à-dire le sujet de la conceptualisationsSelon Van Heijst [Heijst et al.], Nous listons ci-dessous les différents types d'ontologies les plus utilisées:

#### 2.2.5.1 Les ontologies d'application :

Une ontologie d'application décrit la structure des connaissances nécessaires à la réalisation d'une tâche particulière [Heijst et al, 1997].

Elle permet aux experts du domaine d'utiliser le même langage que celui de l'application. Elle réalise trois objectifs :

1. faciliter le processus d'acquisition des connaissances de l'application
2. permettre l'intégration avec les serveurs existants dans l'environnement de l'application
3. sélectionner les structures de données appropriées au modèle computationnel.

Les ontologies de représentation spécifient un formalisme de description qui fournit une structure de représentation et des primitives pour décrire les concepts des ontologies de domaine et des ontologies génériques.

### 2.2.5.2Les ontologies génériques [Bouarroudj ,2010].

Sont aussi appelée Ontologie de haut niveau ou ontologie Top, elles décrivent des concepts généraux, indépendants d'un domaine ou d'un problème particulier.

Elles permettent par exemple de formaliser les aspects temporels ou spatiaux des objets du monde réel. Cyc14 est un exemple d'une ontologie générique portant sur des concepts de haut niveau. Ces dernières décrivent des notions générales comme les notions d'objet, de propriété, d'état, de valeur, de moment, d'évènement, d'action, de cause et d'effet.

### 2.2.5.3 Les ontologies de tâche

Spécifiques à une tâche générique, telle que la vente, et indépendamment du domaine d'application**.** [Bougchiche 2007]

### 2.2.5.4 Les ontologies de domaine [Bouarroudj ,2010].

Elles sont construites sur un domaine particulier de la connaissance. Les ontologies de domaine fournissent des vocabulaires au sujet des concepts dans un domaine et leurs relations au sujet des activités qui ont lieu dans ce domaine, et au sujet des théories et des principes élémentaires régissant ce domaine. Plusieurs ontologies de domaines existent déjà, telle que MENELAS dans le domaine médical. Entreprise est un autre exemple décrivant le domaine de l'entreprise.

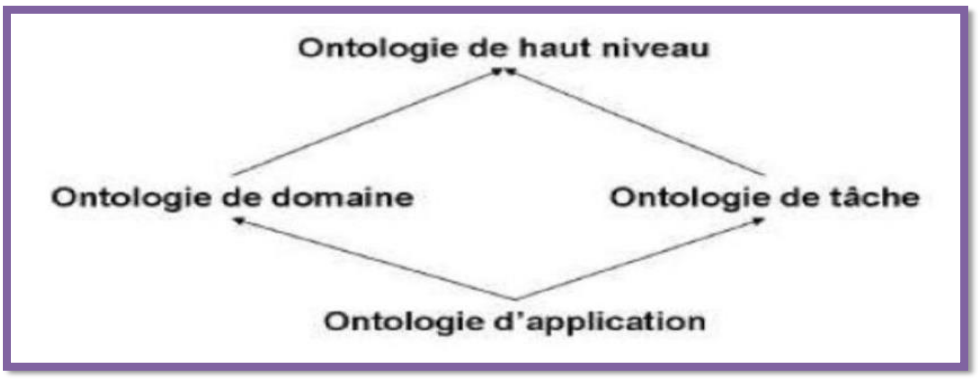



**Figure 27 :** Classification des ontologies selon N. Guarino [Guarino, 1998

### 2.2.5.5Expressivité des Ontologies [Bahia, 2013]

Cette classification est basée sur la force d'expression d'une ontologie, c'est à dire la base d'information que l'ontologie doit exprimer. Nous distinguons les niveaux d'expressivité suivants :

1) **Lexiques contrôlés**

Le lexique est la notion la plus simple possible d'ontologie, qui est une liste finie de termes. Le lexique est un ensemble de sens lexicaux associés à des traits syntaxiques, morphologiques.

2) **Glossaires :**

Ce sont des listes de termes avec leurs significations. Les significations sont le plus souvent exprimées par des énoncés en langue naturelle qui sont principalement destinés à des agents humains**.**

3) **Thesaurus :**

Ils présentent les relations entre les termes, comme dans un dictionnaire synonymie. WordNet6 est un exemple.

4) **Taxonomie informelle :**

Cette catégorie inclut la plupart des ontologies du Web. La taxonomie informelle est une hiérarchie explicite (généralisation et spécialisation), mais pas de relation d'héritage bien définie, c'est à dire l'instance d'une sous-classe n'est pas forcément une instance de la superclasse. Exemple : Taxonomies sur le Web Catégories Yahoo!

5) **Taxonomie formelle :**

Sont des ontologies où les concepts sont organisés selon une 'hiérarchie de sous-classe strictes.

L'héritage est bien défini : chaque instance d'une sous-classe est aussi une instance de la superclasse. Un exemple est UNSPSC8.

### 6) Cadres (Frames) :

Frame (classe) contient un certain nombre de propriétés et ces propriétés sont héritées par les sous-classes et instances. Les Ontologies exprimées en RDFS entrent dans cette catégorie.

## 2.6Web Sémantique

Dans le **Web sémantique**, une **ontologie** est vue comme un ensemble de connaissances, y compris le vocabulaire et les relations **sémantiques**, avec quelques règles simples d'inférence et de logiques relatives à des sujets particuliers

### 2.6.1Définition et Principe du Web Sémantique

Le web sémantique est une extension du web actuel, dont lequel, les informations sont structurées en fonction d'une sémantique bien définie. Le consortium web a défini un cadre général qui permet le partage et la réutilisation des données au travers de différentes applications. Les données du web cachées dans le code HTML manquent d'information sémantique les décrivant. Seul homme peut lire et appréhender le contenu du site web.

Ce contenu n'est pas assigner à la manipulation de manière autonome et intelligente par les programmes informatiques. Ainsi, les ordinateurs jouent un rôle très inactif, quelquefois réduit uniquement à un outil d'affichage du contenu. Ils n'ont pas un réel accès au contenu de la présentation, Ils ne sont pas capables de comprendre la signification de l'information présentée. Le besoin de chercher et retrouver facilement et rapidement les informations pertinentes et complètes est de plus en plus demandé par les utilisateurs du Web.

Les supports actuels construits spécifiquement pour le Web, basés sur la syntaxe des documents, ne satisfont plus ce besoin. Il faut que des agents logiciels aident plus efficacement différentes types d'utilisateurs dans leur accès aux ressources sur le Web. Ainsi, le web devra devenir un espace d'échanges d'informations entre les agents humains et machines.

L'expression Web sémantique, donnée par Tim Berners-Lee au sein du W3C qui définit le web sémantique :

« The semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling computers and peope to work in cooperation»[Berners01]

### 2.6.2L'Objectif Du Web Sémantique

L'objectif primordial du Web sémantique est de permettre aux utilisateurs la manipulation de la totalité du potentiel du Web en partageant, et combinant des informations plus facilement. Pour ce faire, il faut concevoir un environnement en ligne dans lequel on réunit toutes les données de façon logique, ce qui assure la création des liens sémantiques entre ces données pour former une information ultra-pertinente, Seul compte alors le sens des données, et non plus leur place dans un document texte.

Tim Berners-Lee a exposé la hiérarchie du Web sémantique -Basé sur le XML et RDF / RDFS, et en haut de cette construction des ontologies et de règles d'inférence logique, de parfaire le processus da représentation des connaissances basé sur la sémantique et le raisonnement, qui peut être conçue par l'ordinateur et le traitement. Tim Berners-Lee illustre dans un entretien de l'UNESCO en 2000 : "J'ai un double rêve pour le Web. D'une part, je le vois devenir un moyen très puissant de coopération entre les êtres humains. Et dans un second temps, j'aimerais que ce soit les ordinateurs qui coopèrent. [...] Quand mon rêve sera réalisé, le Web sera un univers où la fantaisie de l'être humain et la logique de la machine pourront coexister pour former un mélange idéal et puissant".

### 2.6.3 Les Constituants du Web Sémantique

Pour garantir les sémantiques des informations manipulées par les utilisateurs. Le Web sémantique est composé de :

**1. Méta-données**: Par définition des données sur des données

- Elles complètent donc l'information sur les données à un niveau d'abstraction supérieure.
- Elles peuvent être structurées afin de décrire une ressource quelconque.
- Elles rajoutent un sens aux contenus afin de favoriser leur exploitation par des agents logiciels.

**2. Ontologies**: pour la

- Définitions des concepts.
- Modélisation des connaissances nécessaires à la description et au traitement d'un ensemble de ressources.

**3.Langages**:pour

- Décrire, exploiter et raisonner sur les contenus des ressources.
- Langages de représentation de connaissances afin d'exprimer les ontologies et décrire les annotations.

**4. Des moteurs de raisonnement** :

- Encapsulés dans des systèmes de requêtes et permettant d'inférer sur les annotations d'après les axiomes déclarés dans les ontologies, afin d'interroger le Web et agir sur les réponses obtenues

### 2.6.4 Modèle en couche du Web Sémantique

L'architecture du Web sémantique repose sur une hiérarchie des langages d'assertion et de description d'ontologies ainsi que sur un ensemble de services pour l'accès aux ressources au moyen de leurs références sémantiques, pour gérer l'évolution des

ontologies, pour l'utilisation des moteurs d'inférences capables d'effectuer des raisonnements complexes ainsi que des services pour la vérification de la validité sémantique de ces raisonnements [Oberle04]. [Lekhchine09].

**1. Couche XML** : Base syntaxique

**2. Couche RDF**

- RDF : modèle de triplets pour annoter des ressources.
- RDF Schéma : décrit le vocabulaire (ontologies) utilisé pour ces annotations

**3. Couche Ontologie**

- Langage plus expressif que RDF Schém
- -OWL : Standard courant pour le web : OWL sur une restriction de RDF/S
- OWL Lite / DL / Ful.
- Logiques de description.
- Vérification, classification, identification.

## 2.7conclusion

Nous avons présenté dans ce chapitre quelques concepts de base de domaine d'ontologies et web sémantique .nous proposons de construire une ontologie de domaine pour la description d'arbre de décision. Le chapitre suivant décrit notre Modèle proposée.

# *Chapitre 3*

# *Modèle proposée*

## 3.1 Introduction :

L'ingénierie des connaissances participe à rendre les ontologies interprétables par une machine. Elle est chargée de la « modélisation ontologique » – le fait de définir les primitives de représentation et leur signification qui seront utilisées pour la modélisation formelle des connaissances – étape préalable à la modélisation formelle, c'est-à-dire, à la représentation dans un langage formel des connaissances du domaine.

Comme mentionné dans le chapitre précédant, il existe beaucoup de définitions pour le terme Ontologie, ici nous parlerons d'une ontologie de domaine utilisée dans les arbres de décision. Ce chapitre présente notre contribution à la problématique posée dans ce mémoire, à savoir le développement d'une ontologie d'arbre de décision.

## 3.2 Cycle de vie :

Puisque les ontologies sont destinées à être utilisées comme des composants logiciels dans des systèmes répondant à des objectifs opérationnels différents, leur développement doit s'appuyer sur les mêmes principes que ceux appliqués en génie logiciel. Ainsi, les ontologies doivent être considérées comme des objets techniques évolutifs et possédants un cycle de vie qui nécessite d'être précisé. Dans ce contexte, les activités liées aux ontologies sont, d'une part, des activités de gestion de projet (planification, contrôle, assurance qualité), et d'autre part, des activités de développement (spécification, conceptualisation, formalisation) ; s'y ajoutent des activités transversales de support telles que l'évaluation, la documentation, la gestion de la configuration.

### 3.2.1 Le processus de développement d'ontologie :

La principale phase du cycle de vie est celle de construction qui définit le cycle de développement d'une ontologie et permet de définir les différentes étapes du processus de représentation des connaissances. Il peut être découpé en 3 phases principales: la conceptualisation, l'ontologisation, et l'opérationnalisation.

#### 3.2.1.1 Un cycle de vie inspiré du génie logiciel est proposé dans [Ben Hebireche, 2012]

- **la conceptualisation :**

Elle consiste à identifier précisément, à partir du corpus (ensemble de documents généralement exprimés en langage naturel qui doivent couvrir l'ensemble du domaine de connaissances considéré) et à travers des interviews avec les experts du domaine, les objets conceptuels propres au domaine considéré (concepts, relations et axiomes), certaines connaissances implicitement utilisées dans le domaine ne sont cependant jamais exprimées, ni dans le corpus, ni par les experts, car elles sont acquises par l'expérience sensorielle et accumulées différemment. Un des points les plus délicats de la conceptualisation consiste donc à identifier ces connaissances. La mise en évidence de ces connaissances implicites ne peut a priori se faire que lors de l'utilisation de l'ontologie.

On obtient alors un modèle conceptuel informel (car exprimé en langage naturel) ou une ontologie informelle.

- **L'ontologisation**

L'ontologisation consiste en une formalisation partielle, sans perte d'information, du modèle conceptuel. Il s'agit de transcrire les connaissances exprimées a priori en langage naturel dans un langage ou paradigme de

représentation d'ontologie (le model Frame, le modèle entité relation, le modèle de graphe conceptuel ou réseau sémantique…), afin de respecter les objectifs généraux des ontologies, Gruber propose 5 critères permettant de guider le processus d'ontologisation [Gruber, 1993]:

- la clarté et l'objectivité des définitions, qui doivent être indépendantes de tout choix d'implémentation ;
- la cohérence (consistance logique) des axiomes.
- l'extensibilité d'une ontologie, c'est-à-dire la possibilité de l'étendre sans modification.
- la minimalité des postulats d'encodage, ce qui assure une bonne portabilité.
- la minimalité du vocabulaire, c'est-à-dire l'expressivité maximum de chaque terme.

### L'opérationnalisation

L'opérationnalisation consiste à l'intégration des connaissances dans un système à base de connaissance donc à outiller l'ontologie pour permettre à une machine (via cette ontologie) de manipuler des connaissances du domaine, cette étape consiste ainsi à formaliser complètement l'ontologie obtenue précédemment dans le cadre d'un langage de représentation de connaissances formel et opérationnel.

Dans le cas où le langage d'ontologisation n'est pas opérationnel, il est nécessaire, soit d'outiller ce langage (dans la mesure du possible) soit de transcrire l'ontologie dans un langage opérationnel. Avant d'être livrée aux utilisateurs, l'ontologie doit bien sur être testée par rapport au contexte d'usage pour lequel elle a été bâtie.

- La figure suivante illustre l'enchaînement des trois étapes permettant de passer des données brutes à une ontologie opérationnelle :

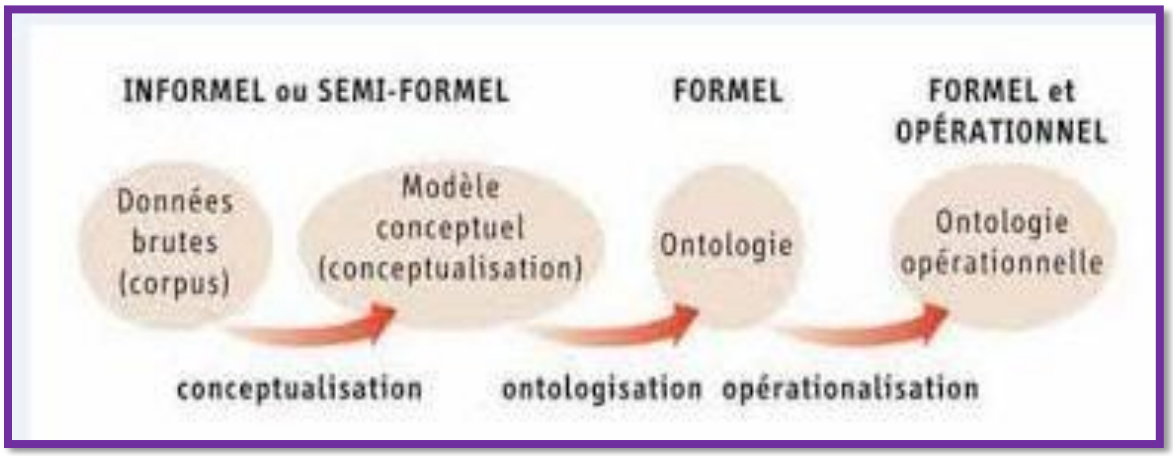



**Figure3. 8 :** Processus de construction d'une ontologie exploitable [Ben Hebireche, 2012].

- **Informelle :** l'ontologie est exprimée en langage naturel (sémantique ouverte). Cela peut permettre de rendre plus compréhensible l'ontologie pour l'utilisateur mais peut rendre plus difficile la vérification de l'absence de redondances ou de contradictions.
- **Semi-informelle :** l'ontologie est exprimée dans une forme restreinte et structurée du langage naturel, cela permet d'augmenter la clarté de l'ontologie tout en réduisant l'ambiguïté.
- **Semi-formelle :** l'ontologie est exprimée dans un langage artificiel défini formellement.
- **Formelle :** l'ontologie est exprimée dans un langage artificiel disposant d'une sémantique formelle ainsi que des théorèmes permettant de prouver ses propriétés,

l'intérêt d'une ontologie formelle est la possibilité d'effectuer des vérifications telles que la complétude, la non-redondance, la consistance, la cohérence, etc.

Un autre cycle est proposé par [Dieng et al. 2001] et il comprend principalement une étape initiale d'évaluation des besoins, une étape de construction, une étape de diffusion, et une étape d'utilisation.

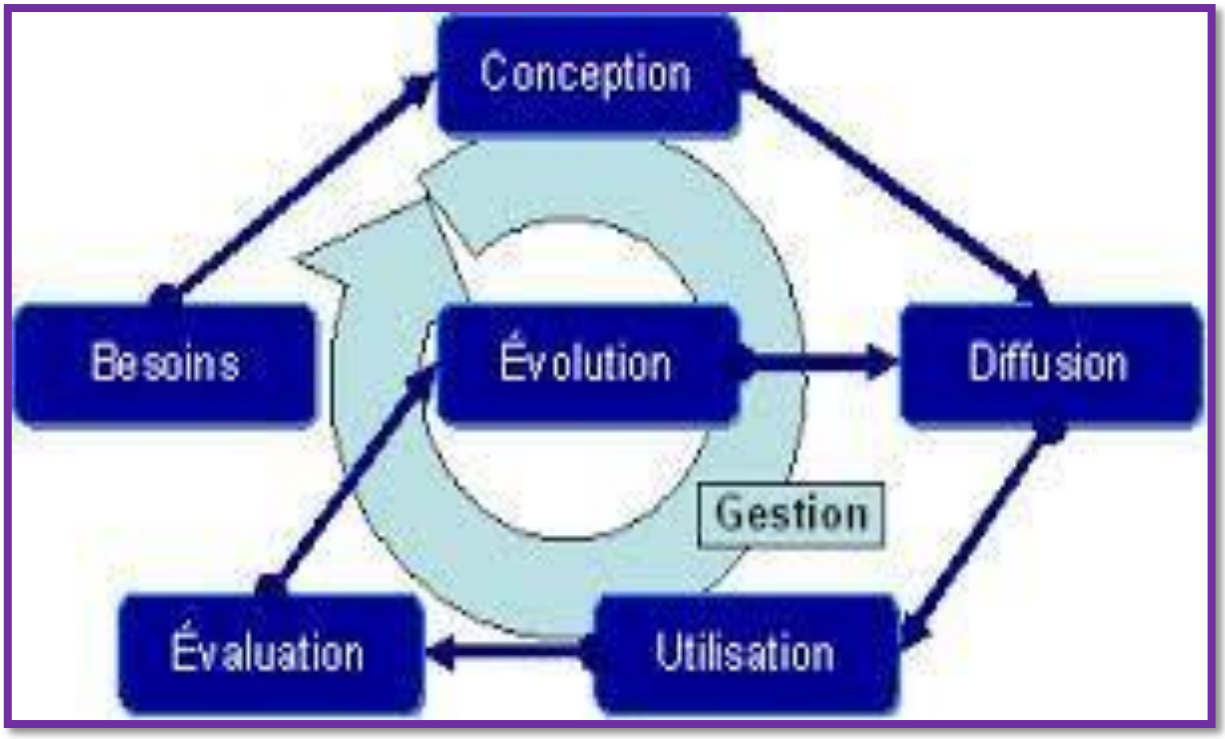



**Figure3. 9** : Le cycle de vie d'une ontologie [Dieng et al. 2001]

La figure 3.10 représente un cycle de vie préposé par Fernandez et ses collègues [Fernandez et al, 1997] qui insistent sur le fait que les activités de documentation et d'évaluation sont nécessaires à l'étape du processus de construction d'ontologie, l'évaluation précoce permettant de limiter la propagation d'erreurs.

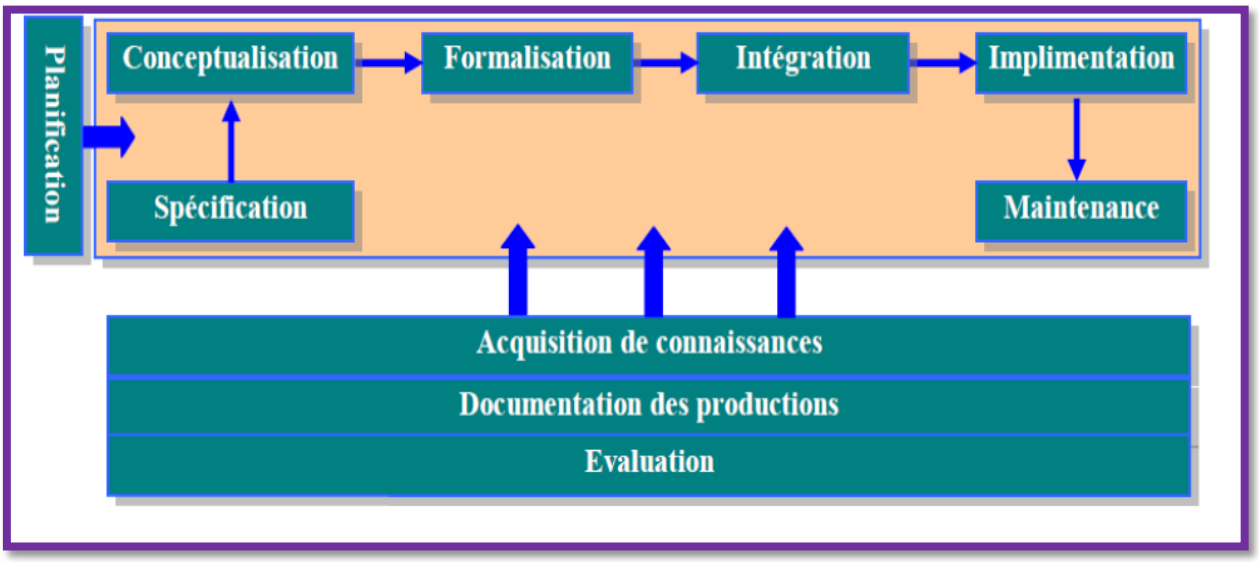



**Figure 3.10** : Cycle de vie de Fernandez & al [Bentahar et al, 2013]

## 3.2 Les méthodes et méthodologies de développement des ontologies

Une ontologie est toujours liée à une méthodologie de construction, à un outil de construction et avec un langage de représentation d'ontologie. A ce niveau, nous présentons les principales méthodologies et méthodes utilisées pour construire les ontologies à partir de zéro

### 3.2.1Uschold et Kings méthode :

Elle est aussi reconnue sous le nom "la méthode ENTERPRISE", c'est le résultat d'une expérience de Mike Uschold et Martin King dans le domaine de construction des ontologies d'entreprise. Et c'est la première méthode proposée pour guider le processus de construction des ontologies [Uschold ,1995].

La méthode ENTERPRISE devise le processus en quatre étapes, qui sont :

- Identifier les objectifs de l'ontologie.
- Construction de l'ontologie : la construction aussi est devisée en trois points:
    - Capturer la connaissance.
    - Coder la connaissance.

- Intégrer des autres ontologies dans l'ontologie en cours de construction.

- Evaluer l'ontologie.
- documenter l'ontologie.

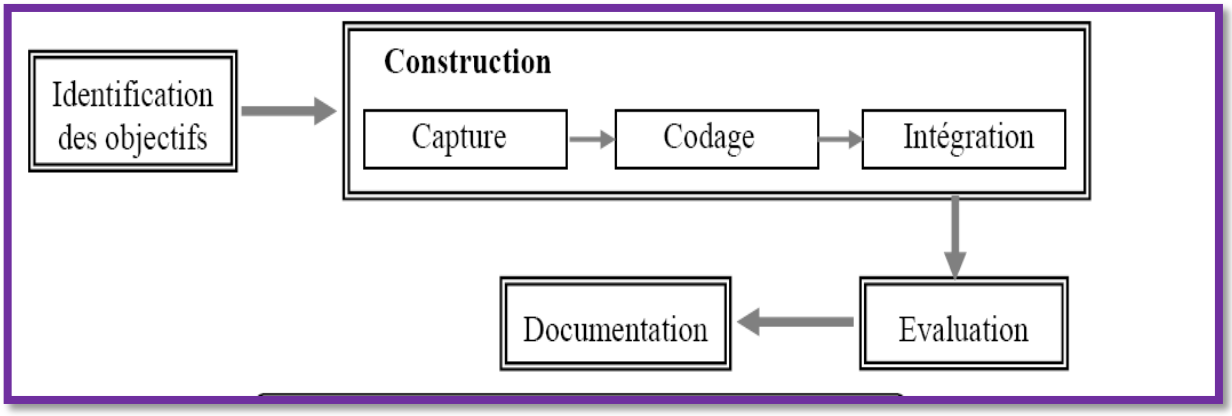



**Figure3. 11** : La méthode Uschold et King [Barakat, 2011].

### 3.2.2 la méthodologie METHONTOLOGY

Le processus de développement d'ontologie défini dans METHONTOLOGY est basé sur les standards de développement des logiciels identifiés par IEEE, ce processus réfère aux activités exécuté durant le développement d'ontologie. Ces activités sont groupées dans trois catégories [Goméez, 2004] :

**1)Les activités de management d'ontologie**

Est devisé en :

- **Planification :** identifier les tâches à exécuter, et identifier pour chaque tâche son arrangement, le temps et les ressources nécessaires que se complètent
- **Contrôle :** assure que les tâches déjà planifiés sont exécutées et terminés de la manière attendue.
- **Assurance de qualité :** assure que chaque sortie de ce processus (ontologie, software et documentation) est satisfaisable.

**2) Les activités orientées au développement d'ontologie**

Il est devisé aussi en trois groupes d'activités :

- **Les activités de pré développement :** contient les activités suivantes :
  - ✓ **Etude de l'environnement:** pour savoir ou l'ontologie sera utilisée (plateforme), et les applications ou l'ontologie sera intégrée.

**Etude de faisabilité :** pour savoir, est ce que il est possible de construire une tel ontologie, et est-il utile de la construire.

- **Les activités de développement :** contient les activités suivantes :
  - ✓ **Spécification :** définir pourquoi l'ontologie doit être construire, quels sont ces cas d'utilisations dans le futur et quels sont leurs futurs utilisateurs.
  - ✓ **Conceptualisation :** construire un modèle conceptuel sur la connaissance de domaine.
  - ✓ **Formalisation :** transformer le modèle conceptuel en modèle formel.
  - ✓ **Implémentation :** coder l'ontologie avec un langage approprié.
- **Les activités post-développement :** contient les activités suivantes :
  - ✓ **Maintenance :** assure-la mis a jour et la correction de l'ontologie.
  - ✓ **Utilisation :** dans cette activité l'ontologie peut être utilisée par des autres ontologies.

**3)Les activités de support d'ontologie**

Contient aussi un ensemble d'activités qui doivent être exécutées pour affiner la construction de l'ontologie

- **Acquisition de connaissance :** acquérir le connaissance de domaine.

- **Evaluation :** évaluer l'ontologie, son environnement et la documentation, et prendre des jugements par rapport à une référence.
- **Intégration :** si nous avons besoin aux autres ontologies pour les utiliser dans l'ontologie en cour de construction.
- **Merging :** union de deux ontologies de même domaine par la fusion de ces derniers dans une seule super ontologie.
- **Alignement :** union de deux ontologies de même domaine par l'établissement des différents liens entre ces ontologies, cette activité préserve les deux ontologies, originaux.
- **Documentation :** produire des documents clairs pour chaque étape et chaque produit.
- **Gestion de configuration :** enregistrer tous les documents et les codes générés pour permettre un contrôle de changement.

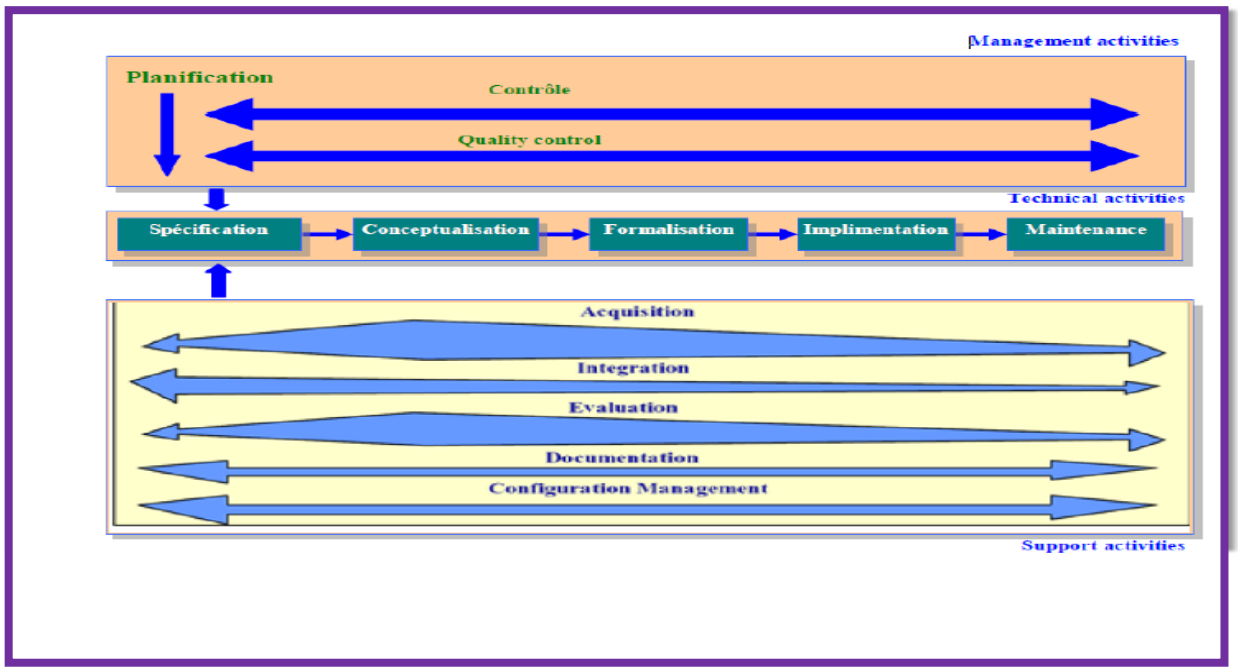



**Figure3. 12** : Le processus de développement d'ontologie de METHONTOLOGY [Bahia,2013]

## 3.3Processus de construction de notre ontologie (ontoDTA) :

### 3.3.1Spécification des besoins

L'application de cette étape dans le cadre de notre travail, nous a conduits à répondre à certaines questions importantes, sur le domaine, le but et la portée de l'ontologie à construire

- **Questions de compétence :**

1. Quelles sont les techniques de Data Mining ?
2. Quelles sont les tâches de AD?
3. Quels sont les concepts de base qui doivent être compris par un débutant en AD?
4. Quels sont les types de données en AD?
5. Quelles sont les composantes d'un arbre en AD?
6. Quels sont les types des AD existants?
7. Comment construire un AD?
8. Quels sont les algorithmes les plus utilisés en AD?
9. Quelles sont les métriques utilisées en AD?
10. Y at-il un logiciel qui permet à un débutant de commencer par AD?
11. Quels sont les domaines d'application de AD?

- **Domaine:** L'arbre de décision
- **Objectif opérationnel (But):** l'ontologie est utilisée comme une référence pour les chercheurs de ce domaine.
- **Utilisateurs :** chercheurs de domaine.
- **Degré de formalisme :** formel
- **Porté (liste des termes importants) :** taxonomie d'arbre de décision
- **Source de connaissances :** Documents techniques relatifs à la méthode des AD.

### 3.3.2 la conceptualisation

La conceptualisation est la plus importante dans le processus de développement de l'ontologie. Elle mérite une attention particulière car elle détermine le reste de la construction de l'ontologie. Cette étape est divisée en deux sous-étapes :

- L'extraction des termes importants à partir d'un corpus.
- Le choix des concepts à partir des termes extraits.

### 3.3.2 .1 Construction d'un glossaire des termes importants

Il contient tous les termes importants du domaine, indifféremment de leurs types (concept, relation, ou propriété).

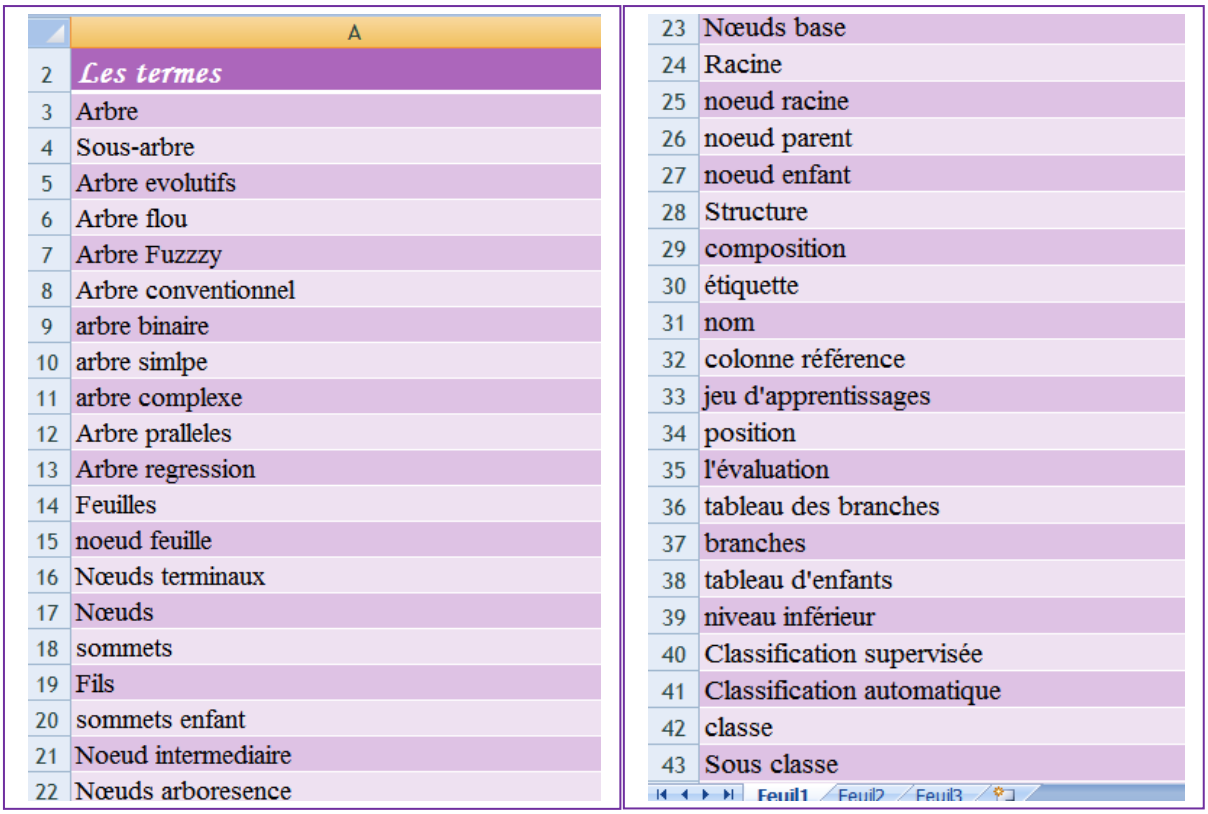
On a fait l'étude dans le domaine d'arbre de décision sur un corpus textuel

[Bénédicte, 2008] , [Brahimi ,2014] , [Aalain ,2007] , [Lachiche 2008] ,[ Schwander ,2009] , [Rahmoun ,2011] , [Frédéric , 2015] , [Stéphane,2014] , [Alouane ,2008] ,

A partir de ces documents, on a extrait manuellement plus de **200** termes (figure 3.14) dont le but est la description des arbres de décision.

Il existe plusieurs type d'extraction des termes en peut citer :

- **Manuellement :** c'est difficile et très lent mais l'extraction manuelle permet d'extraire tous les termes importants (pour des différentes langues).
- **Semi-automatique**.
- **Automatique** : l'extraction à l'aide des outils d'extraction mais le problème c'est quand on a un document qui contient des termes écrits par plusieurs langues.



| | A |
|---|---|
| 2 | *Les termes* |
| 3 | Arbre |
| 4 | Sous-arbre |
| 5 | Arbre evolutifs |
| 6 | Arbre flou |
| 7 | Arbre Fuzzzy |
| 8 | Arbre conventionnel |
| 9 | arbre binaire |
| 10 | arbre simlpe |
| 11 | arbre complexe |
| 12 | Arbre pralleles |
| 13 | Arbre regression |
| 14 | Feuilles |
| 15 | noeud feuille |
| 16 | Nœuds terminaux |
| 17 | Nœuds |
| 18 | sommets |
| 19 | Fils |
| 20 | sommets enfant |
| 21 | Noeud intermediaire |
| 22 | Nœuds arboresence |
| 23 | Nœuds base |
| 24 | Racine |
| 25 | noeud racine |
| 26 | noeud parent |
| 27 | noeud enfant |
| 28 | Structure |
| 29 | composition |
| 30 | étiquette |
| 31 | nom |
| 32 | colonne référence |
| 33 | jeu d'apprentissages |
| 34 | position |
| 35 | l'évaluation |
| 36 | tableau des branches |
| 37 | branches |
| 38 | tableau d'enfants |
| 39 | niveau inférieur |
| 40 | Classification supervisée |
| 41 | Classification automatique |
| 42 | classe |
| 43 | Sous classe |

Feuil1 Feuil2 Feuil3

**Figure3. 13** : La base de données de notre ontologie

Il Ya quelque termes relatifs au domaine qui seront représentés dans l'ontologie finale, par exemple. :.

- **L'apprentissage automatique :** (*machine learning* en anglais), un des champs d'étude de l'intelligence artificielle, est la discipline scientifique concernée par le développement, l'analyse et l'implémentation de méthodes automatisables qui permettent à une machine (au sens large) d'évoluer grâce à un processus d'apprentissage, et ainsi de remplir des tâches qu'il est difficile ou impossible de remplir par des moyens algorithmiques plus classiques.
- **L'apprentissage non supervisé :** (ou classification automatique). Quand le système ou l'opérateur ne disposent que d'exemples, mais non d'étiquettes, et que le nombre de classes et leur nature n'ont pas été prédéterminés,
- **Les forêts d'arbres décisionnels :** (ou forêts aléatoires de l'anglais « *Random decision forest* ») ont été formellement proposées en 2001 par Leo Breiman et

Adèle Cutler. Elles font partie des techniques d'apprentissage automatique.

- **L'entropie de Shannon :** due à Claude Shannon, est une fonction mathématique qui, intuitivement, correspond à la quantité d'information contenue ou délivrée par une source d'information. Cette source peut être un texte écrit dans une langue donnée, un signal électrique ou encore un fichier informatique quelconque (collection d'octets).
- **Le pré-élagage :** La première stratégie utilisable pour éviter un des arbres de décision consiste à proposer des critères d'arrêt lors de la phase d'expansion…
- **Le post-élagage :** la seconde stratégie consiste à construire l'arbre en deux temps.
- **EdiNoS :** est un outil open source de conception et de révision de bases de connaissances opérationnelles s'appuyant sur la notion de situation. Il permet de construire rapidement des arbres de décisions « enrichis » : chaque nœud du graphe contenant les règles permettant de choisir le nœud suivant.

### 3.4Définition des concepts candidats :

Nous avons ensuite, choisi plus de **80** concepts candidats de notre ontologie à partir de corpus textuel.

Ensuite, on fait un choix des concepts candidats qui se base sur une sélection de l'un des termes le plus fréquent.

- **Exemple 1:** Parmi les deux termes Surajustement et Surapprentissage on choisit Surapprentissag

| Terme 1 | Terme 2 | Concept |
|---|---|---|
| Surajustement | Surapprentissage | Surapprentissage |

- **Exemple 2 :** Parmi les deux termes Elegage et prining on a choisit Elegage

| **Terme 1** | **Terme 2** | **Concept** |
|---|---|---|
| Elegage | prining | Elegage |

- **Exemple 3** : le terme EdiNoS c'est un concept

| Terme | Concept |
|---|---|
| EdiNoS | EdiNoS |

| | A | B | C | D |
|---|---|---|---|---|
| 1 | Terme1 | Terme2 | Terme3 | concept |
| 2 | Arbre | classification hiérarchique | / | Arbre |
| 3 | Méthode | procedé | / | Méthode |
| 4 | Nœud | sommet | / | Nœud |
| 5 | Nœud enfant | sommet enfant | fils | fils |
| 6 | nœud intermediaire | sous -arbre | Nœuds arboresence | sous -arbre |
| 7 | Racine | Nœuds base | noeud parent | Racine |
| 8 | Nœuds terminaux | Feuilles | extrimté | Feuilles |
| 9 | structure | composition | / | strcture |
| 10 | Exemple | seri d'attribut | / | Exemple |
| 11 | Méthode d'arbre decison | Méthode de segmentation | Méthode de particionement | Méthode d'arbre decison |
| 12 | Arbre de decision | Arbre de discrimination | / | Arbre de decision |
| 13 | Algorithme | procédure | / | Algorithme |
| 14 | Modèle | Patterns | / | Modèle |
| 15 | Clustering | Segmentation | / | Segmentation |
| 16 | Tableau d'index | tableau de réferance | / | Tableau d'index |
| 17 | Type | variété | sorte | Type |
| 18 | ID3 | Iterative Dichotomiser 3 | / | ID3 |
| 19 | C4.5 | successeurs d'ID3 | / | C4.5 |
| 20 | CHAID | / | / | CHAID |
| 21 | CART | / | / | CART |

**Figure3. 14** : Les concepts candidats

## 3.5 Ontologisation (Formalisation) :

Après la conceptualisation, il faut créer une hiérarchie de classes. Ces classes doivent être regroupés en arbres de classification de concepts, procédé de haut

en bas en commençant par les concepts les plus généraux et en terminant par la spécialisation des concepts. Donc ce travail doit être mené par l'ingénieur de la connaissance.

Les classes plus générales tel que : **Thing, Data Mining, Technique, , Arbre de décision.**

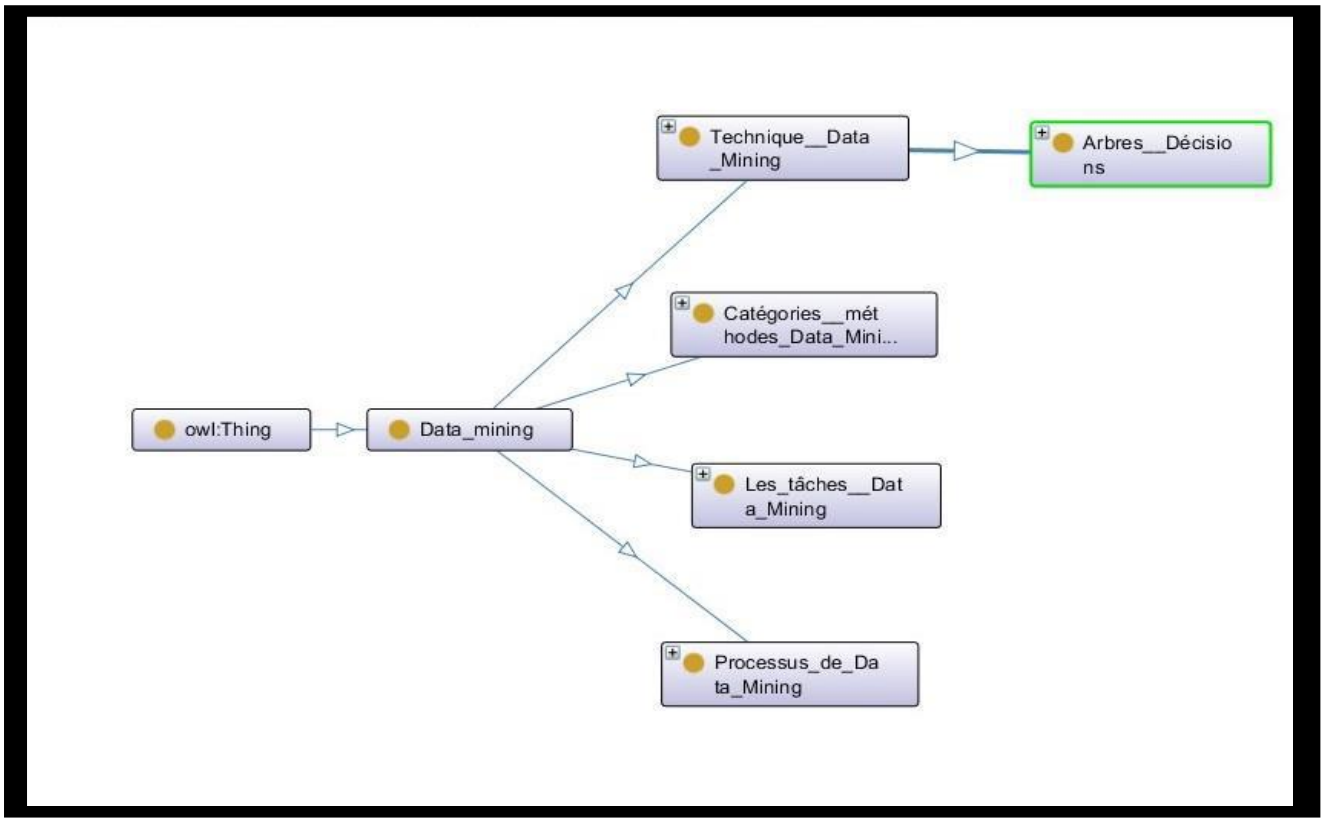



**Figure3. 15** : Les concepts généraux

**Thing** : est une classe principale qui contient la sous classe **Data ming** qui contient **Technique** comme une sous classe. **Arbre de decision :** est une classe qui contient les sous classe suivantes et représenter dans la figure3.17.

- Concepts de base
- Construction
- Contexte d'utilisation
- Position
- Logiciels
- Représentation

- Métrique
- Les taches
- Structure
- Type AD
- Amélioration des performances.

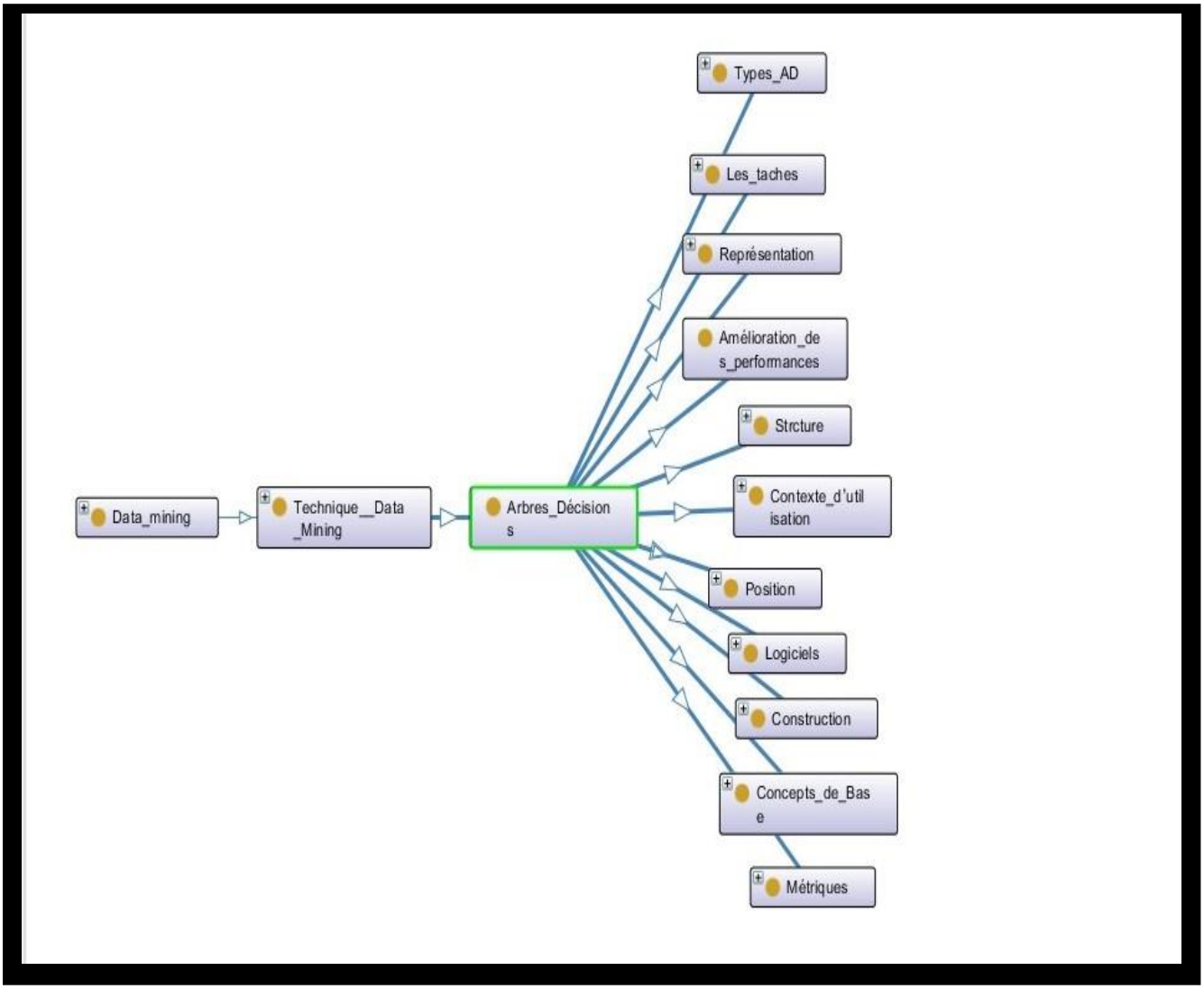



**Figure3. 16** :Les sous classe D'arbre décision.

Ensuite, nous avons affiné chacune de ces classes pour répondre aux questions de Compétence . Par exemple, pour répondre à la question de compétence numéro8 : Quels sont les algorithmes les plus utilisés dans AD ? nous avons fait la recherche sur

les algorithmes les plus utilisés dans AD et nous avons trouvé d'algorithmes:  les (ID3, C5 .0, CART, AID,QUEST,CHAD ,SPRINT ...) .

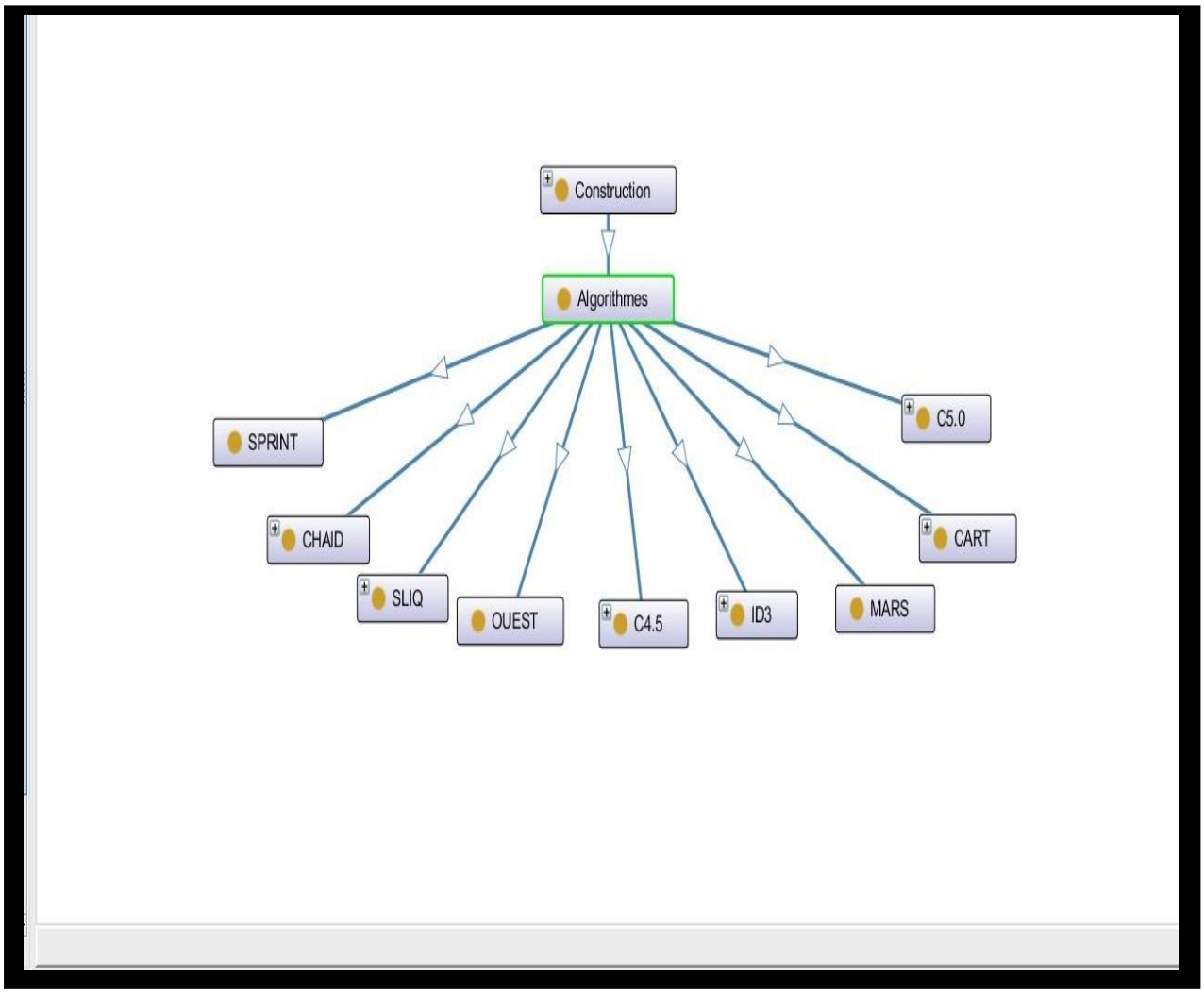



**Figure 3.17** : Les algorithmes les plus utilisées en AD

Et pour répondre à la question de compétence numéro 9 : Quelles sont les métriques utilisées dans AD ? nous avons effectué les recherches sur les métriques les plus utilisées dans AD, et nous avons trouvé les paramètres suivants: Entropy, Gini-Index, Information- Gain, Cost-Complexity, Chi-Square, Tree-Size ...

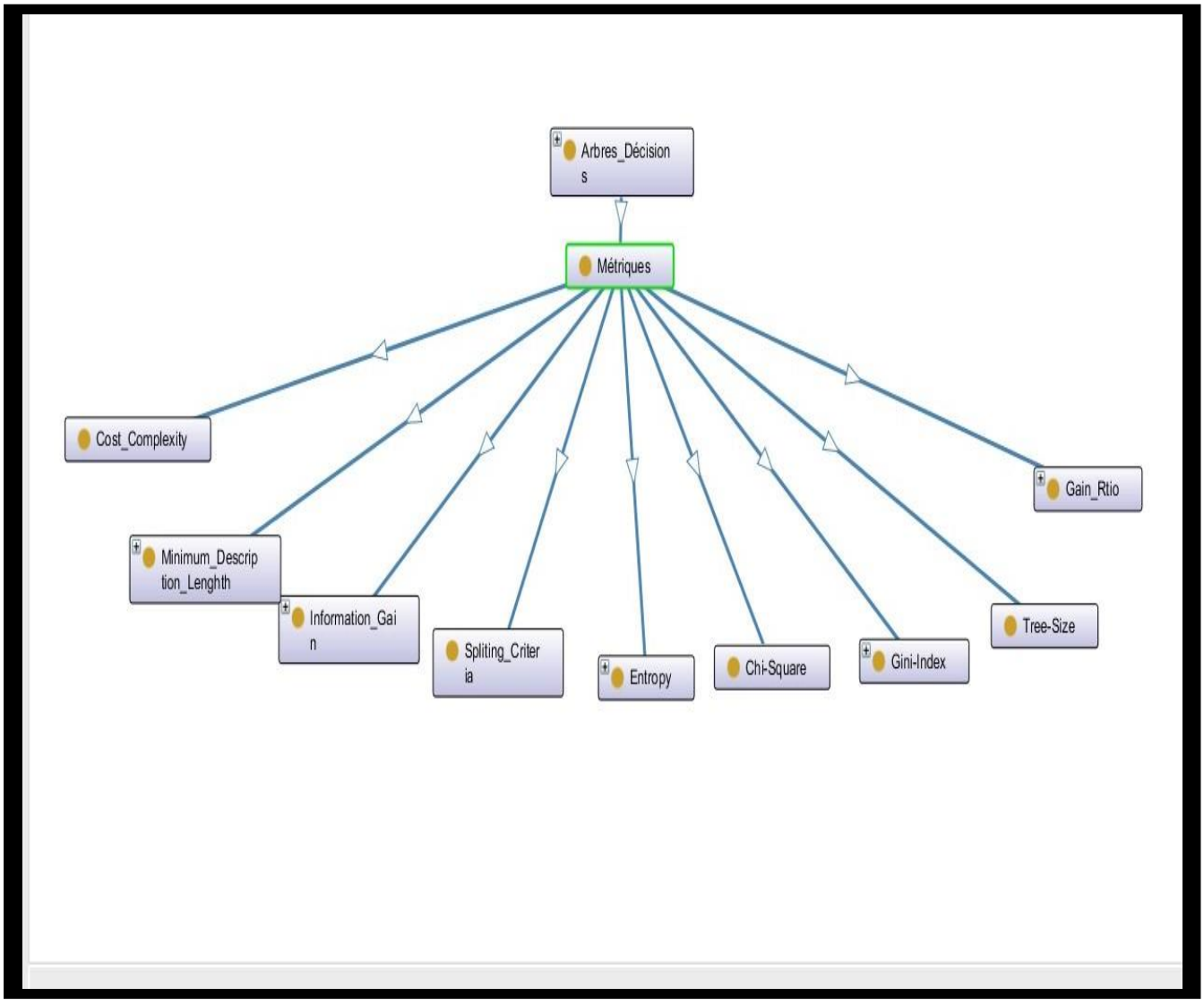



**Figure 3.18** : Les métriques les plus utilisées en AD

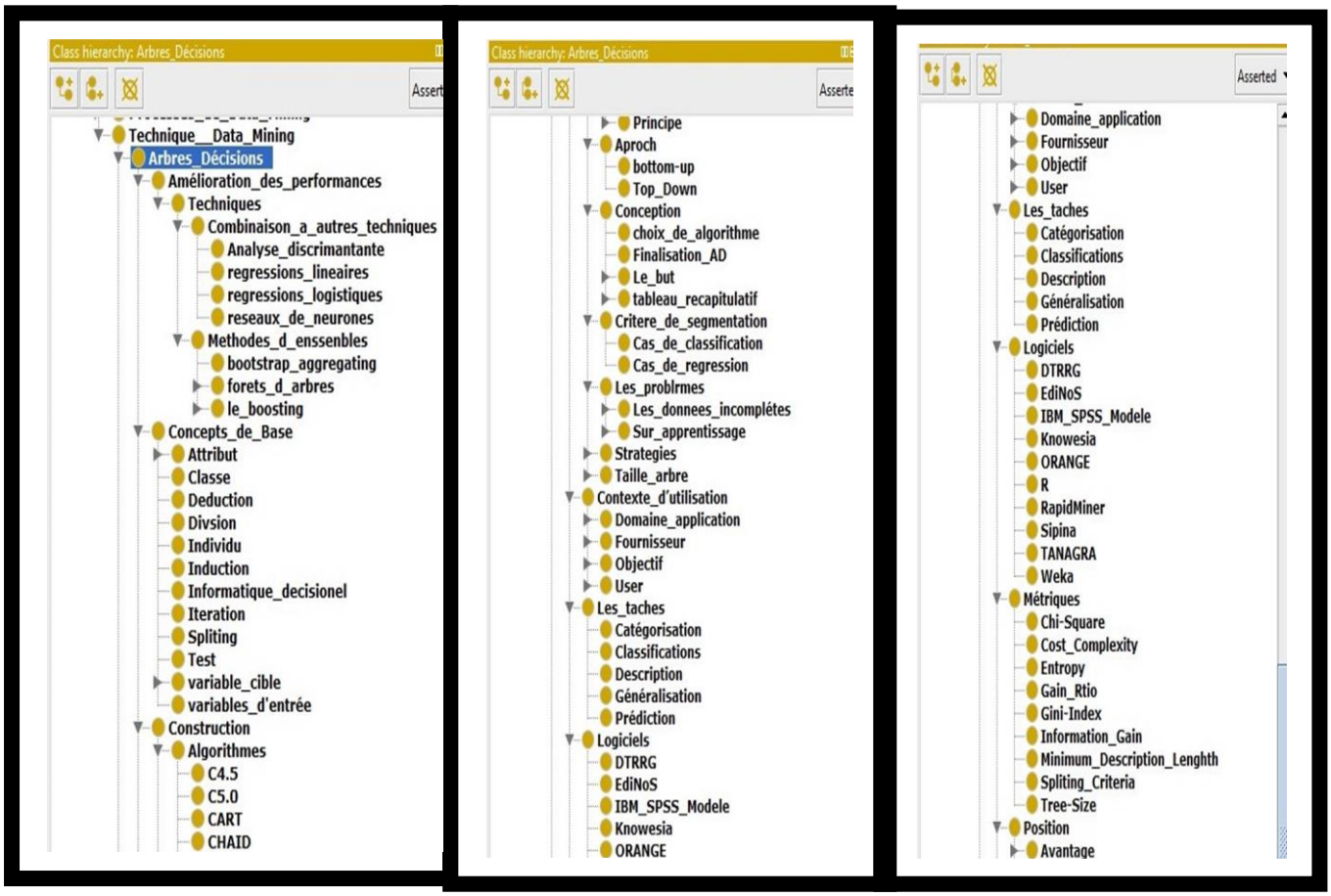



**Figure3. 19** : Représente la hiérarchie générale des classes de notre ontologie OntoDTA

## 3.6 Intégration

Nous envisageons de réutiliser des ontologies déjà existantes lors du développement de notre ontologie. Plusieurs ontologies ont été analysées.

Pour le développement de notre ontologies, nous sommes inspirés l' ontologies

1. L'ontologie DMOP ([Hilario et al., 2011], [Keet et al., 2013] et [Keet et al., 2015]). Bien que d'ontologies n'a été développée dans le but de décrire les concepts de base d'une technique spécifique de Data Mining, mais l' ontologies (DMOP) soutiennent notre sujet et utilisent des concepts que nous pouvons les intégrer dans notre ontologies

 les concepts **"Algorithm", "Process", "Data"**, **"Task"** et **"Metrics"** de l'ontologie**DMOP**.

## 3.7Conclusion

Ce chapitre est la base de notre recherche, là où nous avons présenté la conception de notre ontologie, nous avons commencé de corpus jusqu'à l'extraction des concepts (conceptualisation) et en passant par l'étape de l'ontologisation, donc il nous reste les étapes d'implémentation et de validation qui sont le but de chapitre suivant.

# *Chapitre4*

# *Implémentation De Notre Ontologie OntoDTA*

L'objectif de ce chapitre est de présenter notre contribution au problème posé par ce mémoire à savoir le développement d'une ontologie d'arbre de décision.

## 4.1Logiciels utilisés :

### 4.1.1 Langages de représentation des ontologies

L'une des décisions importantes a prendre dans le processus de développement des ontologies, est de sélectionner le langage dans lequel l'ontologie sera implémentée, dans les dernières années, plusieurs langages de représentations de connaissances ont été créées, et plusieurs langages de représentations de connaissances ont été utilises aussi ; Généralement, la sélection d'un langage n'est pas basée sur la méthode de représentation de connaissance ou sur le mécanisme d'inférence utilise par l'application qui va utiliser l'ontologie, mais plutôt sur les préférences individuels du développeur, malgré qu'un mauvaise choix du langage peut engendrer des problèmes une fois que l'on veut utiliser l'ontologie dans l'application.

#### 4.1.1.1OWL : [Loraine, 2008]

Doit permettre de représenter des ontologies, en particulier sur le Web. Il est fondé sur la syntaxe RDF/XML et est dédié totalement à la représentation des ontologies. OWL est destiné à être utilisé lorsque les informations contenues dans les documents doivent être traitées par des applications logicielles, c'est-à-dire lorsqu'elles ne sont pas simplement montrées à l'utilisateur. Une ontologie OWL est composée d'un en-tête (métadonnées), d'axiomes et de faits. Les axiomes concernent la définition complète ou partielle de concepts et de relations, la spécification de propriétés sur les relations (propriétés algébriques)

et la définition d'axiomes sur les classes et les relations (équivalences, expression booléenne).

Parmi les relations, on distingue celles dont le domaine de valeur sera de type primitif (attribut) de celles dont le domaine de valeur sera un autre concept (relation). Les faits concernent des individus pour lesquels on donne des valeurs aux propriétés des classes dont ils sont les instances.

#### 4.1.1.2XML [Brad, 2001]

XML (eXtensible Markup Language) est un métalangage utilisé pour définir des langages de marquage comme XHTML qui permettent la structuration des documents du Web, non pas sur la base d'une structure figée (statique) comme le permettait HTML, mais en laissant la possibilité au concepteur de distinguer les données selon leur sens et leur contenu.

Un document XML se présente sous la forme de données taguées par un ensemble de balises, chacune pouvant comporter des attributs et des valeurs.

#### 4.1.1.3 RDF : [Loraine, 2008]

Est un modèle de représentation sémantique des informations du Web qui utilise la syntaxe XML. Il permet la mise en place de descriptions simples sur les ressources du Web comme les auteurs de pages Web, leur date de création, etc. Les ressources du Web sont l'élément de base de RDF. Chaque ressource est pourvue d'un identifiant uniforme de ressource (URI*).* Initialement recommandé parle W3C dans le but de standardiser les définitions et les usages des métadonnées, RDF est également utile à la représentation de données elles-mêmes.

### 4.2Logiciels de validation :

Le W3C propose plusieurs outils de tests et de validation de documents en ligne (au format HTML, XHTML, RSS, RDF...). Ils sont destinés aux développeurs web et aux webmestres. Selon les outils il est possible de tester un document avec son URL, en le téléchargeant (upload), ou par copier-coller de code source.

- **Les outils proposés :**

  - **CSS Validato**r : permet de valider les feuilles de style CSS ou les documents contenant des CSS

  - **Feed Validation Service** : permet de valider des flux au format RSS et ATOM.

  - **Link Checker** : permet de vérifier la validité des liens hypertextes d'une page web ou de tout un site.

  - **W3C mobileOK Checker :** permet de connaître le niveau de compatibilité d'un document web avec un terminal mobile. C'est notamment utile si l'on souhaite créer des pages web plus facilement accessibles aux téléphones mobiles.

  - **Semantic Extractor** : extrait des informations d'un document au format HTML en analysant le code sémantique. Cela permet par exemple de vérifier qu'avec une structure de document riche du point de vue sémantique, on peut générer facilement un sommaire ou un index d'abréviations.

  - **RDF Validator** : permet de valider et de visualiser des documents au format RDF. Il est notamment possible d'avoir une représentation graphique des triples. Valider ses documents et son

code permet d'assurer une meilleure interopérabilité et pérennité aux contenus mis en ligne.

### 4.3L'Editeur d'Ontologies Protégé :

Protégé est un éditeur qui permet de construire une ontologie pour un domaine donné, de définir des formulaires d'entrée de données, et d'acquérir des données à l'aide de ces formulaires sous forme d'instances de cette ontologie. Protégé est également une librairie Java qui peut être étendue pour créer de véritables applications à bases de connaissances en utilisant un moteur d'inférence pour raisonner et déduire de nouveaux faits par application de règles d'inférence aux instances de l'ontologie et à l'ontologie elle-même (métaraisonnement). Développé par le Stanford Medical Informatics SMI de l'université de médecine de Stanford. Il est adapté à la construction d'ontologies depuis la version PROTÉGÉ 2000.(Bahia ,2013) L'implémentation de notre ontologie – OntoDTA- c'est effectuée à travers l'éditeur d'ontologies **Protégé_5.1.0** plusieurs raisons ont motivé notre choix :

- Protégé est un éditeur d'ontologies open source et gratuit.
- Il possède une interface modulaire, ce qui permet son enrichissement par des modules additionnels (plugins).
- Il permet l'édition et la visualisation graphique d'ontologies.
- Il permet de contrôler la cohérence de l'ontologie par la vérification des contraintes.
- Protégé est fourni une API écrite en JAVA, qui permet de développer des applications pouvant accéder aux ontologies de Protégé et de les manipuler.
- Il permet d'importer et d'exporter des ontologies dans les différents langages de spécification d'ontologies (RDF-Schéma, OWL, DAML, OIL,...etc.)

- Exécuter des raisonneurs.

## 4 .3.1 Interface de PROTÉGÉ OWL

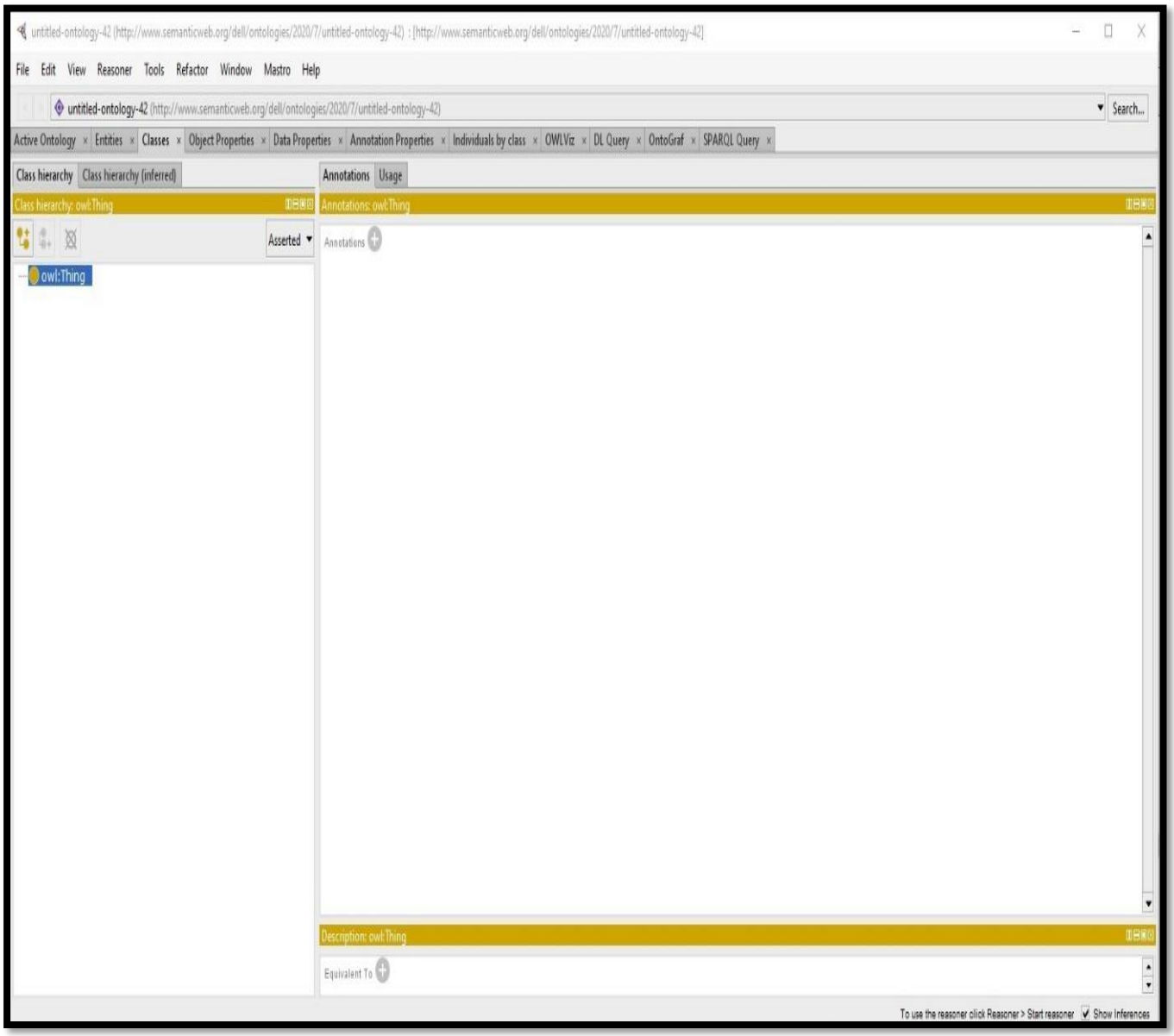



**Figure4. 20** : Interface de PROTÉGÉ OWL.

Comme illustré dans la Figure 4.21, l'interface utilisateur de PROTÉGÉ-OWL plugin fournit un ensemble d'onglets :

- L'onglet «**Active ontologie**» permet de visualiser les informations relatives au projet telle que l'URI de l'ontologie et les espaces de nom associés et d'ajouter des métadonnées par le biais d'annotations.
- L'onglet «**Entities**» permet de définir les concepts de l'ontologie et leurs conditions/restrictions.
- L'onglet «**Object Properties**» permet de définir les relations entre les différents concepts.
- L'onglet «**Data Properties**» permet de définir les attributs de concepts. Leurs conditions/restrictions.
- L'onglet «**Individuals**» permet de peupler l'ontologie.
- L'onglet «**OntoGraf**» permet de visualiser le graphe de l'ontologie.
- **OWLViz** est conçu pour être utilisé avec l'éditeur Protégé OWL plugin. Cet outil permet de visualiser la hiérarchie des classes dans une ontologie OWL
- D'autre onglet **DL Query** peuvent être ajouté par l'intermédiaire de plugins supplémentaires.

### 4.4 Présentation de notre l'ontologie OntoDTA :

L'objectif premier d'une ontologie est de modéliser un ensemble de connaissances dans un domaine donné, pour notre projet nous avons pu construire une ontologie d'arbre de decision.

### 4.4.1 Création des classes et la hiérarchie des classes.

Nous avons commencé par éditer toutes les classes de l'ontologie spécifiées dans l'étape de conceptualisation en utilisant l'onglet *OWL Class*.

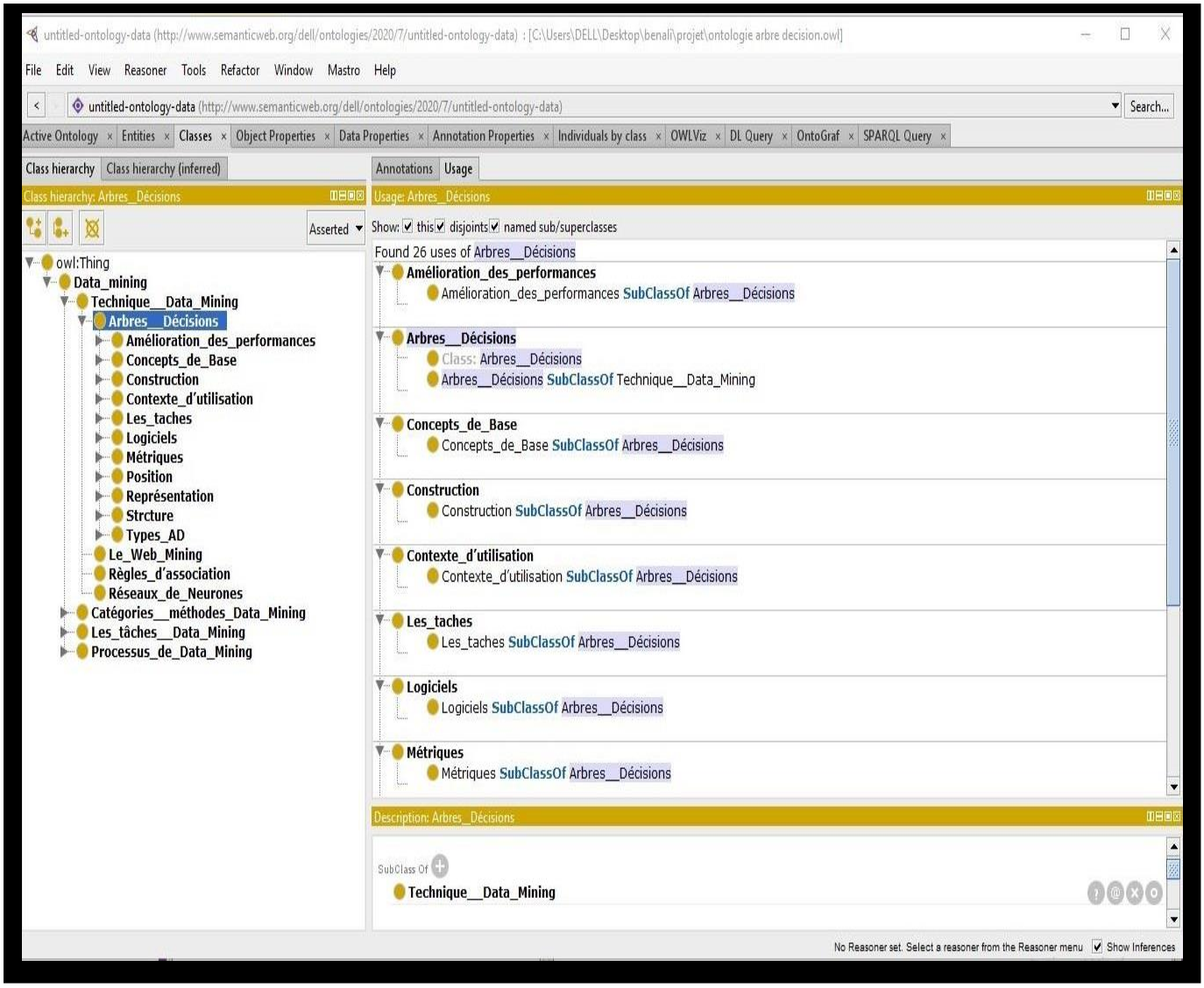



**Figure 4.21** : Création des classes et hiérarchie des classes

Parmi les tâches de la méthode des arbres de décision nous trouvons : la catégorisation, la classification, la description, la généralisation et la prédiction (Figure 4.23).

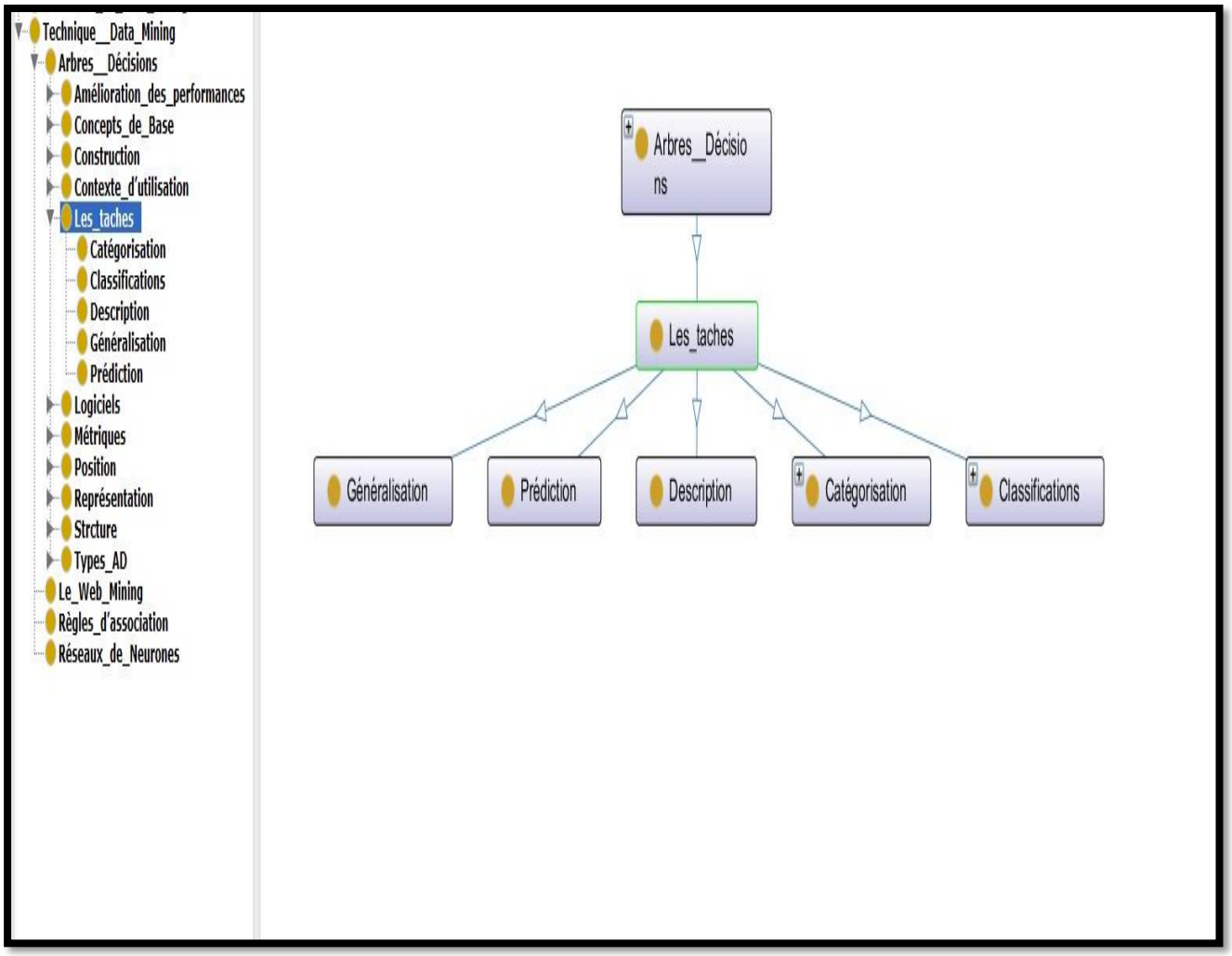



**Figure 4.22** : **Les tâches de la méthode des arbres de décision**.

La Figure 4.24 montre les algorithmes de la méthode des arbres de décision tel que CART, SPRIT, CHAID, SLIQ, ID3, C4.5, MARS, C5.0…etc.

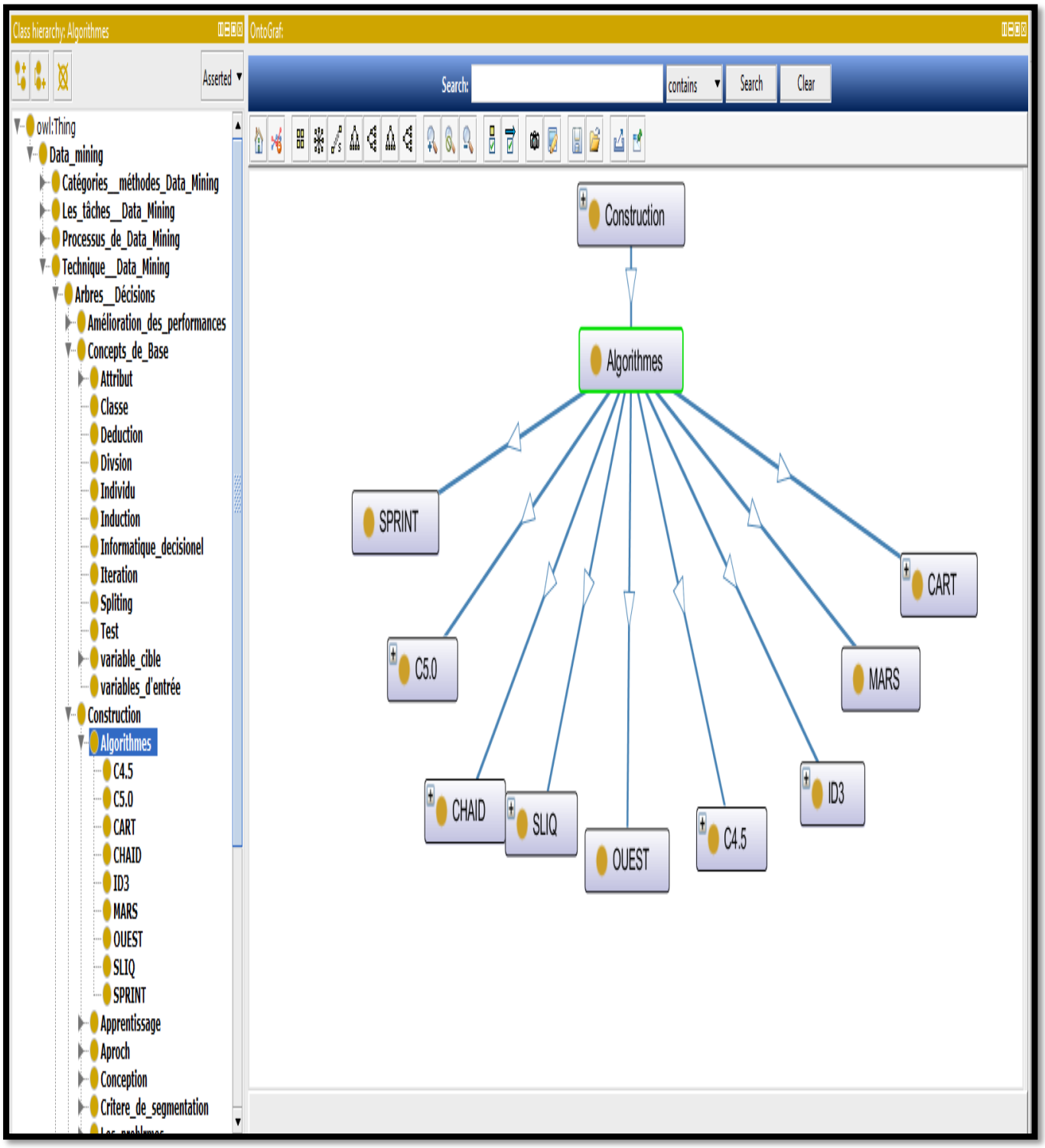



**Figure 4.23** : Les algorithmes de la méthode des arbres de décision.

### 4.4.2 Les relations sémantiques

Par exemple :la relation sémantique “ **représente la valeur de** ” entre les concepts le nœud feuille et variable cible

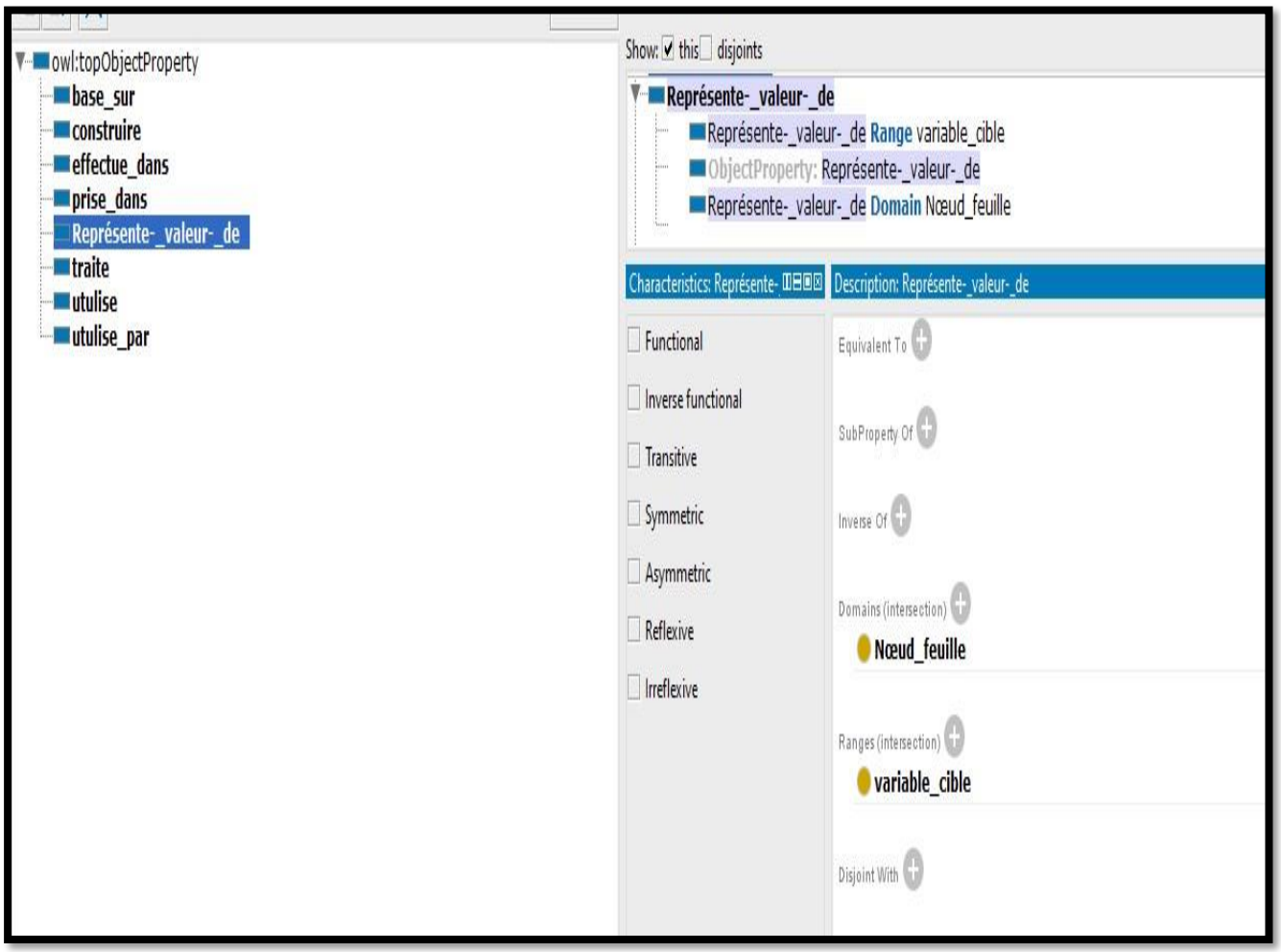



**Figure 4.24** :Une partie des relations sémantiques de l’ontologie OntoDTA.

### 4.4.3Création des propriétés

Après avoir construit les classes, nous créons maintenant les propriétés (sont appelées rôles en logique de description) pour chacune d’elles en utilisant l’onglet *Properties*. Il existe deux types de propriétés, les attributs ***‘Data type Properties’*** et les relations ***‘Object Properties’***. Les propriétés d’une classe sont les propriétés héritées de sa superclasse, plus ses propres propriétés privées.

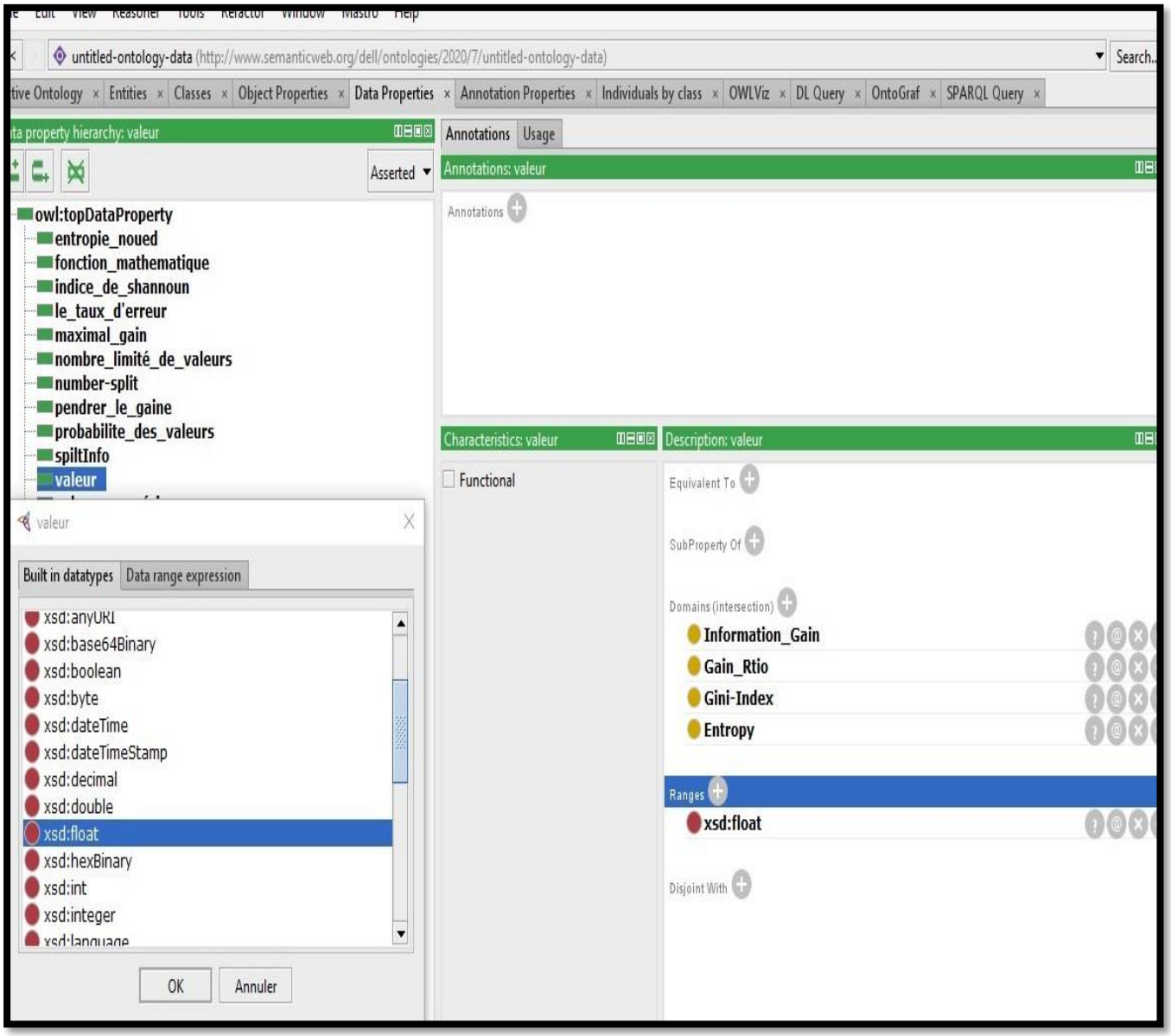



**Figure4. 25** : Les Data type Propertés' de l'ontologie.

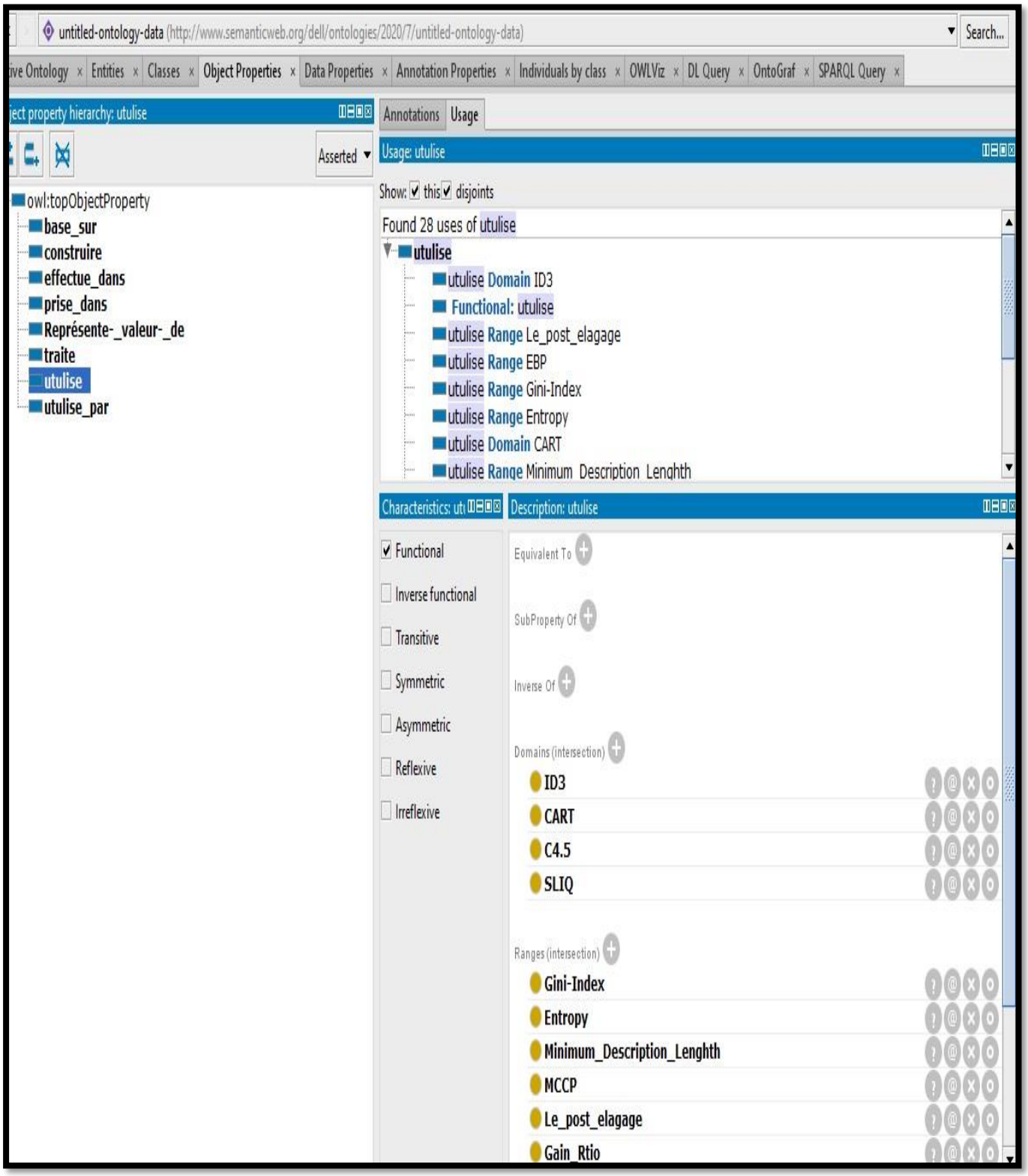



**Figure 4.26** :Les Object Properties' de l'ontologie.

### 4.4.3 Définition des instances :

Après la création des classes, des sous- classes et leurs propriétés, nous entamons maintenant une autre phase de l'implémentation de notre ontologie c'est l'instanciation des concepts.

L'onglet « Individuals »permet de créer des instances et de leur affecter des propriétés, conformément à la définition des classes et des propriétés effectuée dans l'onglet « Classes ». Sur l'écran présenté **(Figure 4 .28).** Il est par exemple possible d'éditer les informations concernant :**TANAGRA**

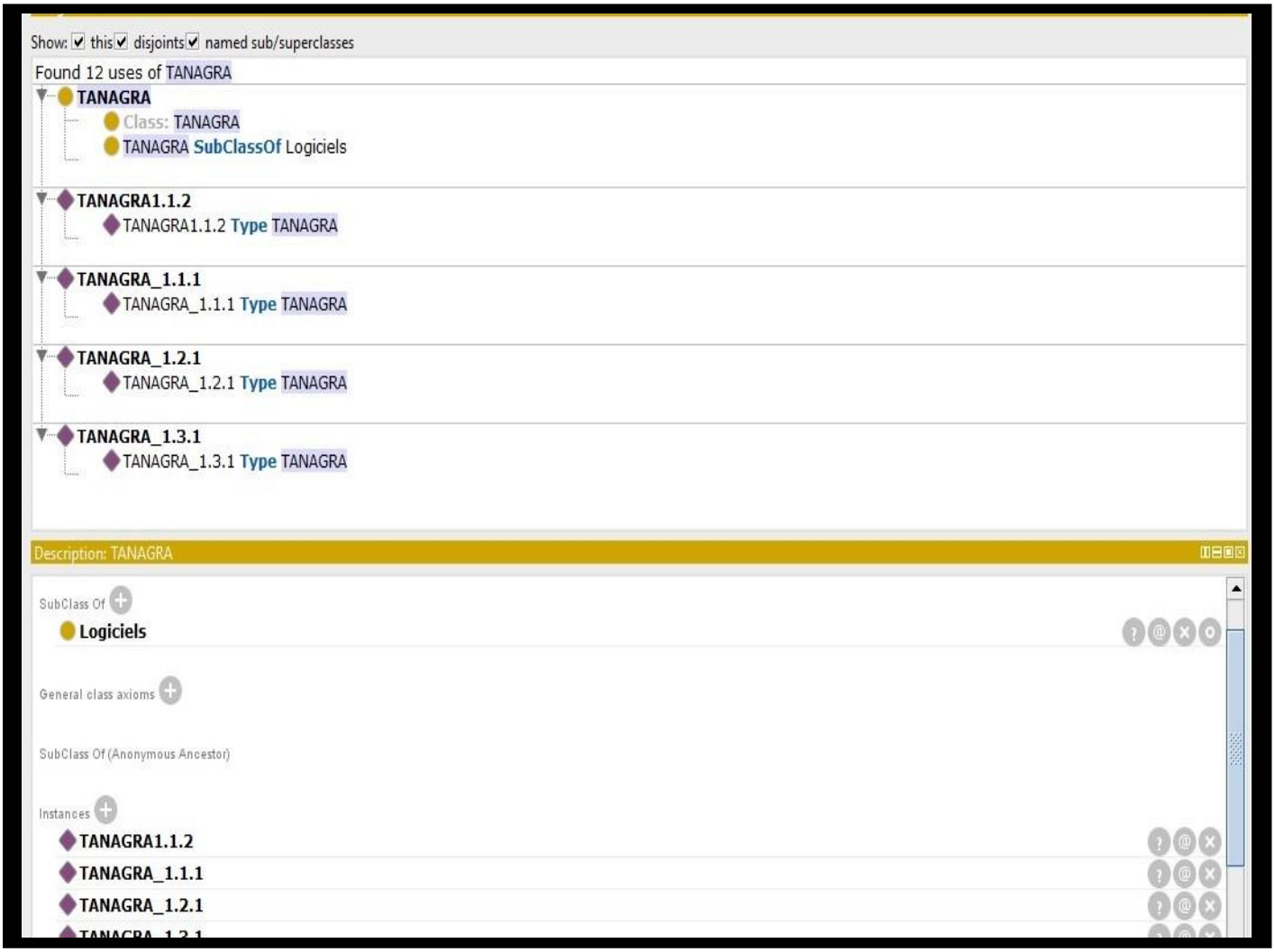



**Figure4. 27** :Les instances.

### 4.4.4 Les Type Des Arc :

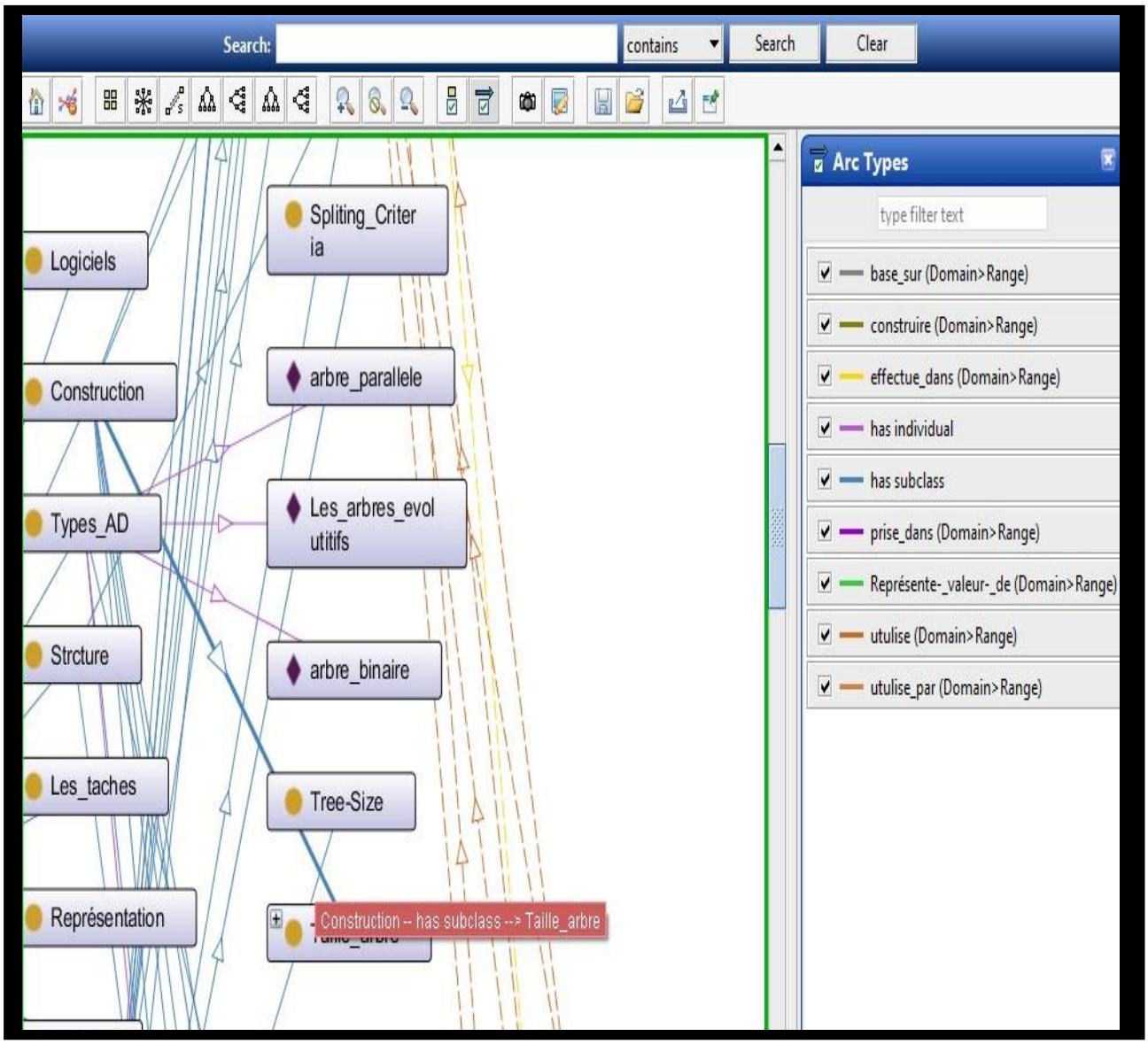



**Figure 4.28** :Les Type Des Arc .

Par exemple la construction **has subclass** de taille-arbre

### 4.4.5Création des axiomes

Par exemple  La propriété “number-split" est une propriété de la classe Splitting, de type: integer ,voir la figure 4.30.

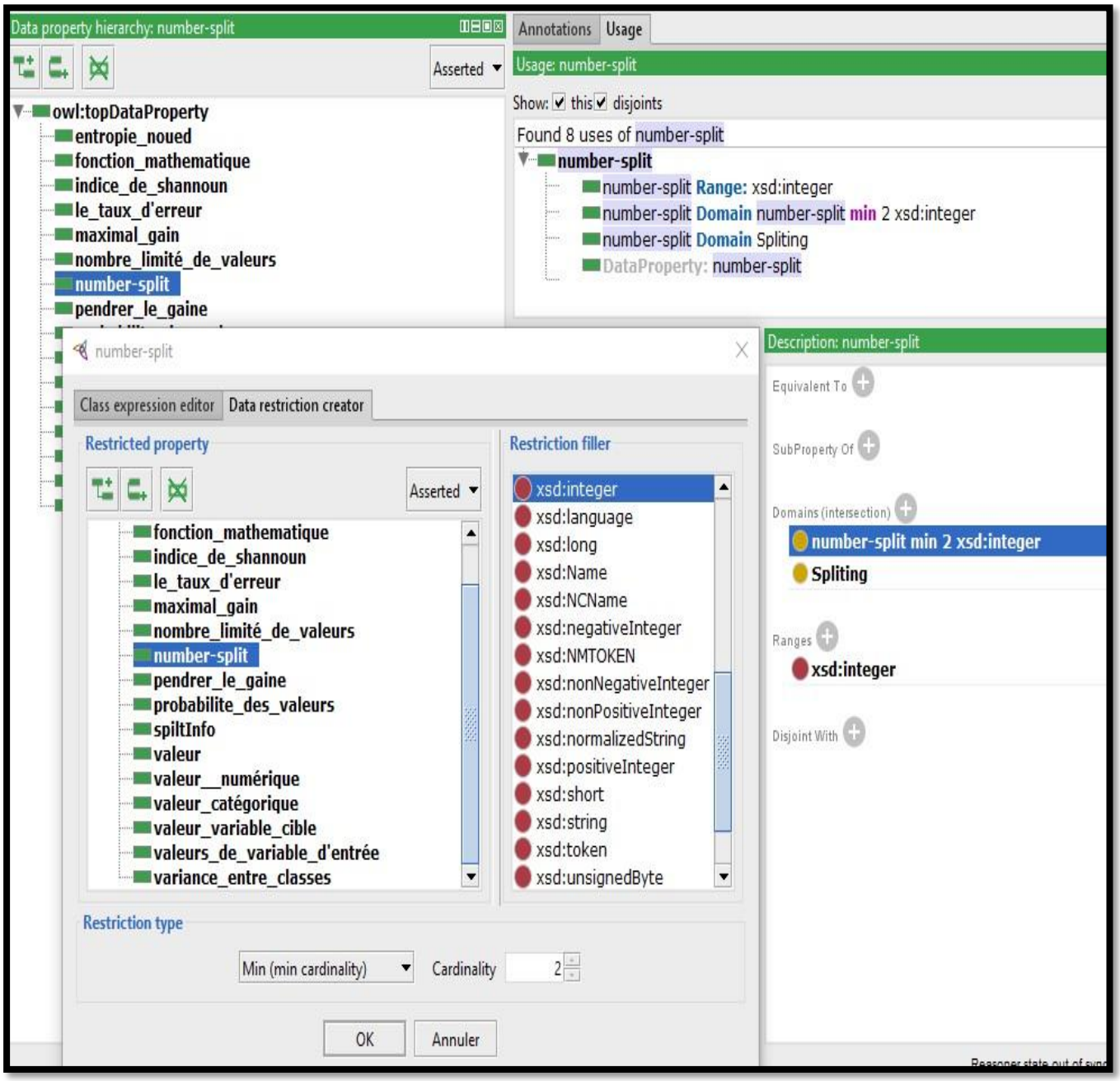



**Figure 4.29 : Les axiomes en OntoDTA.owl.**

## 4.5 Présentation du prototype

L'ontologie que nous avons obtenue à ce stade est une ontologie formelle, il nous reste d'opérationnaliser cette ontologie pour qu'elle soit concrètement manipulable dans un système informatique. Par conséquent, elle doit être spécifiée dans un langage de représentation de connaissance doté de capacités d'inférences.

Après l'étude effectuée dans le chapitre 3 nous avons construit notre ontologie OntoDTA en OWL. Thing est une classe prédéfinie. Toute classe OWL est une sous-classe **d'owl:Thing.**

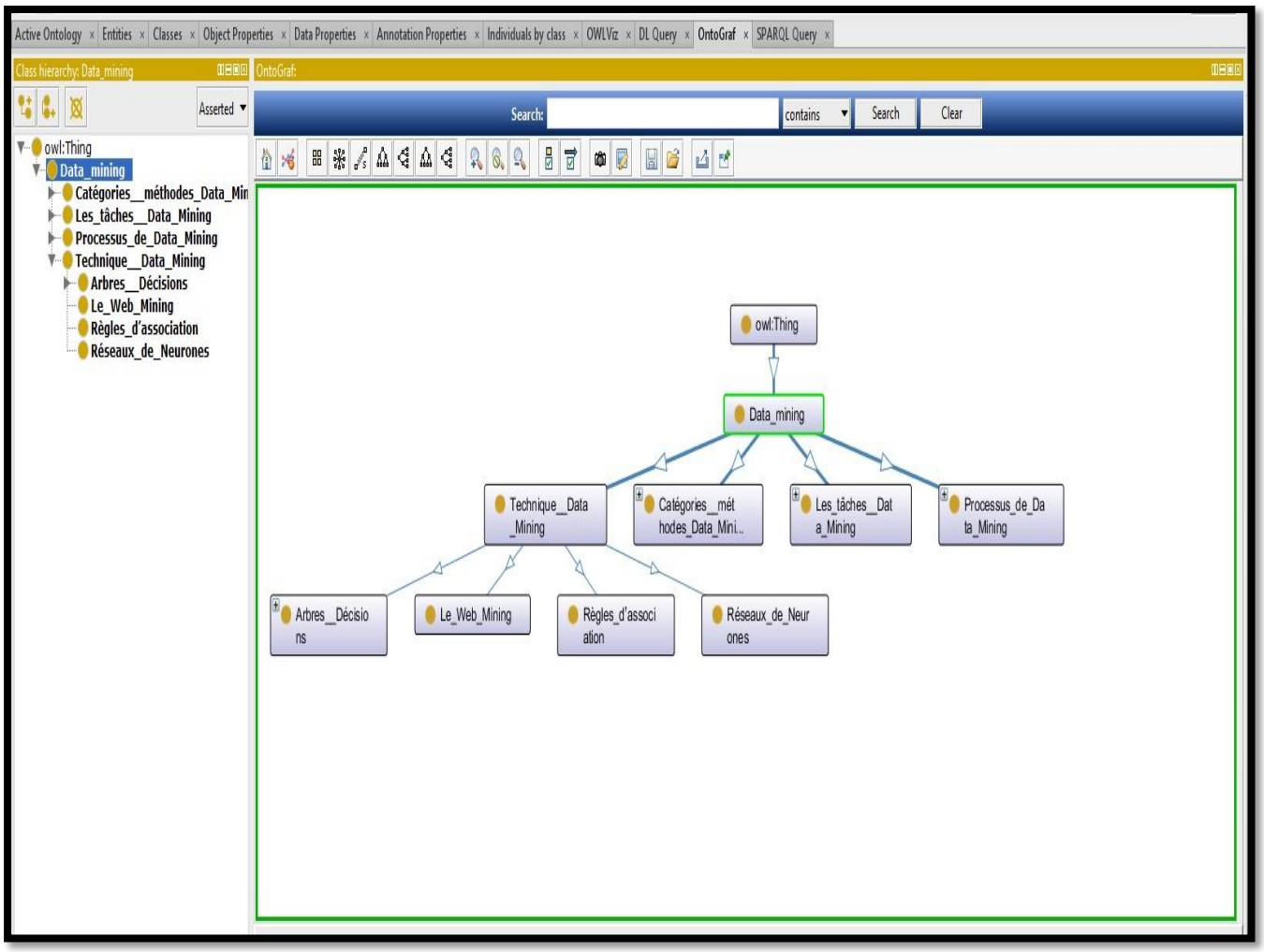



**Figure 4.30** : Hiérarchie de concept Data Mining**.**

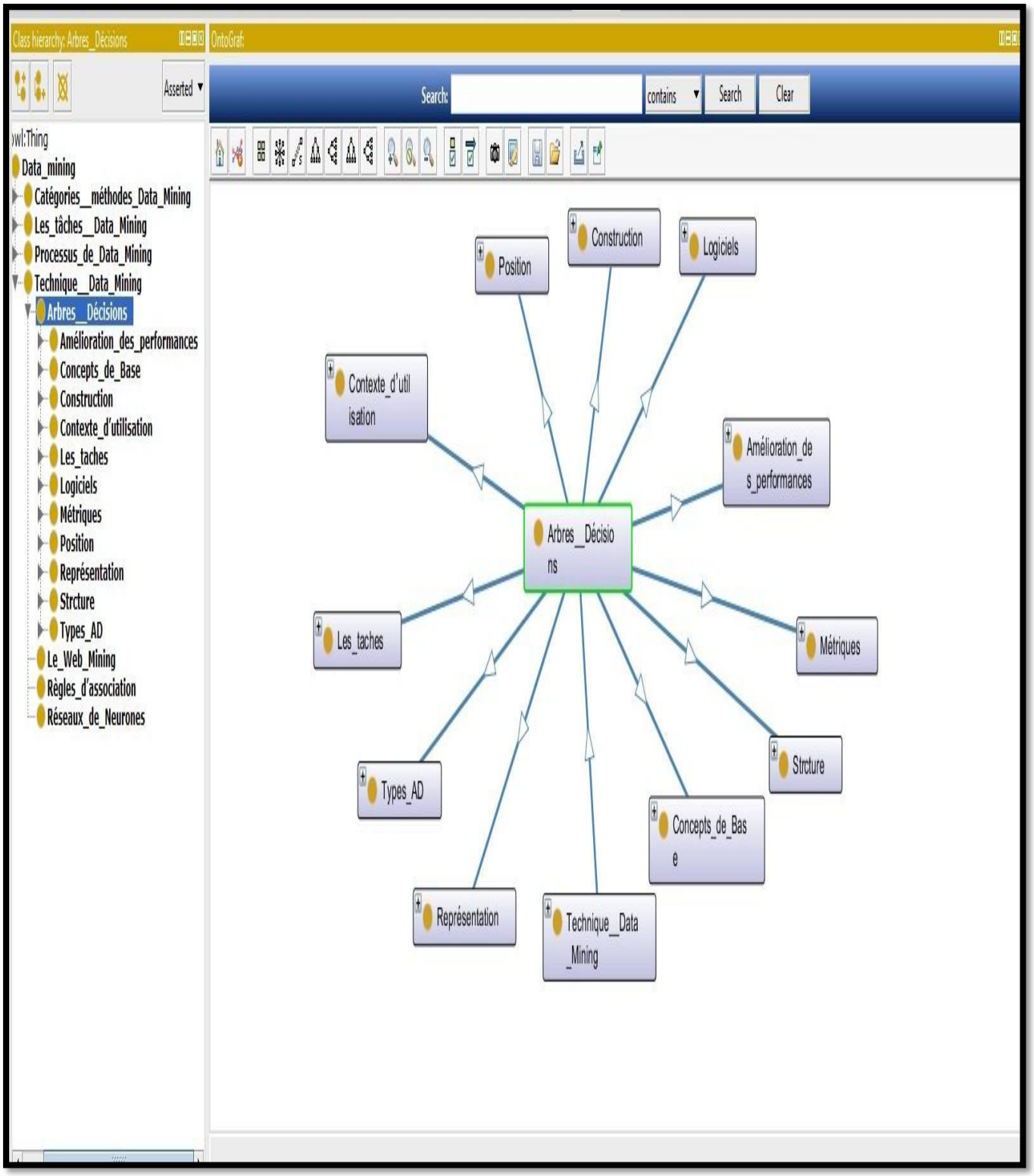



**Figure1. 32 :**Présente Quelque sous classe de la classe Arbre de décision.

La figure suivante présente les sous classes de la classe concepts de base tels que :

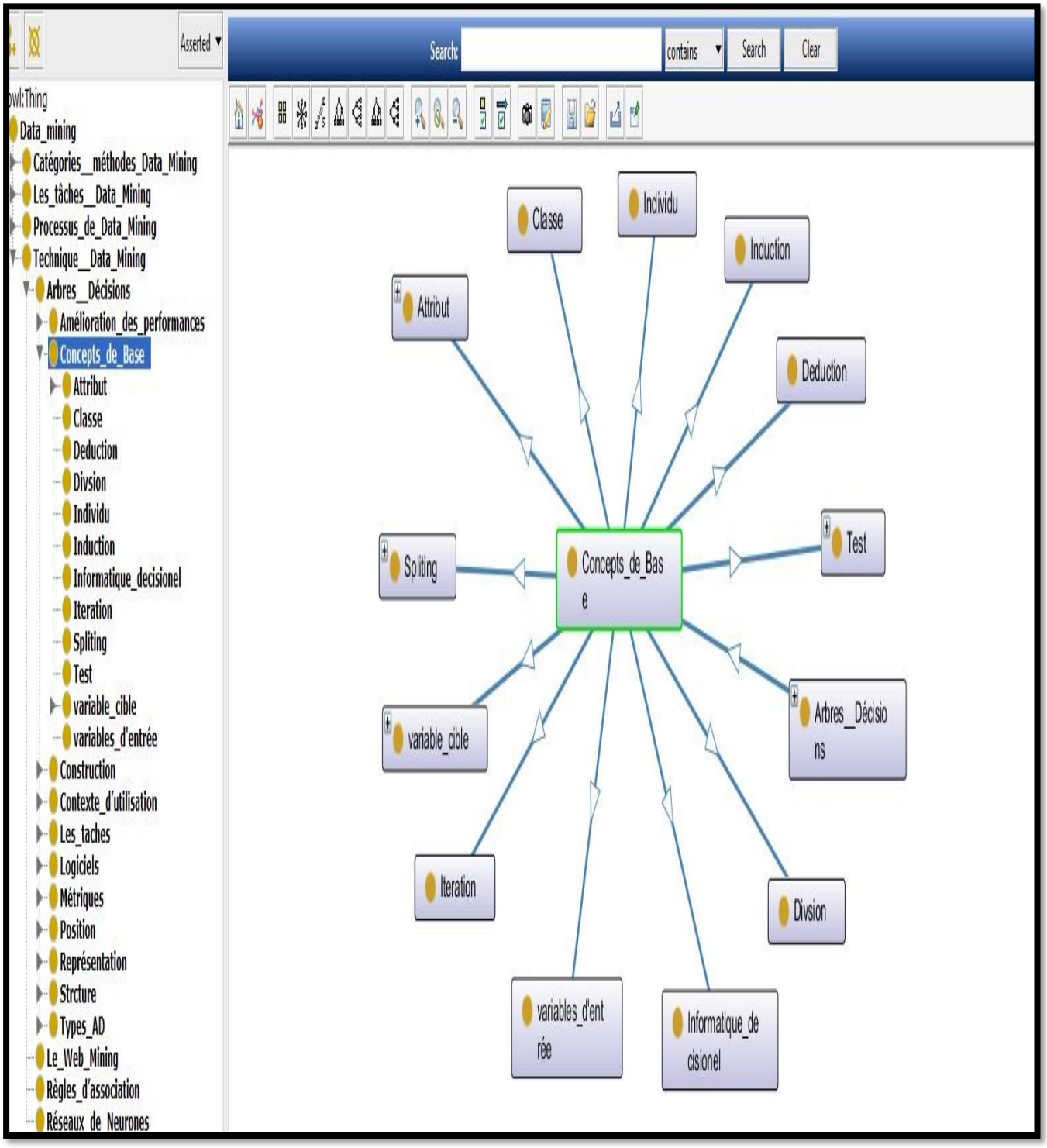



**Figure4.33** : Concepts de base.

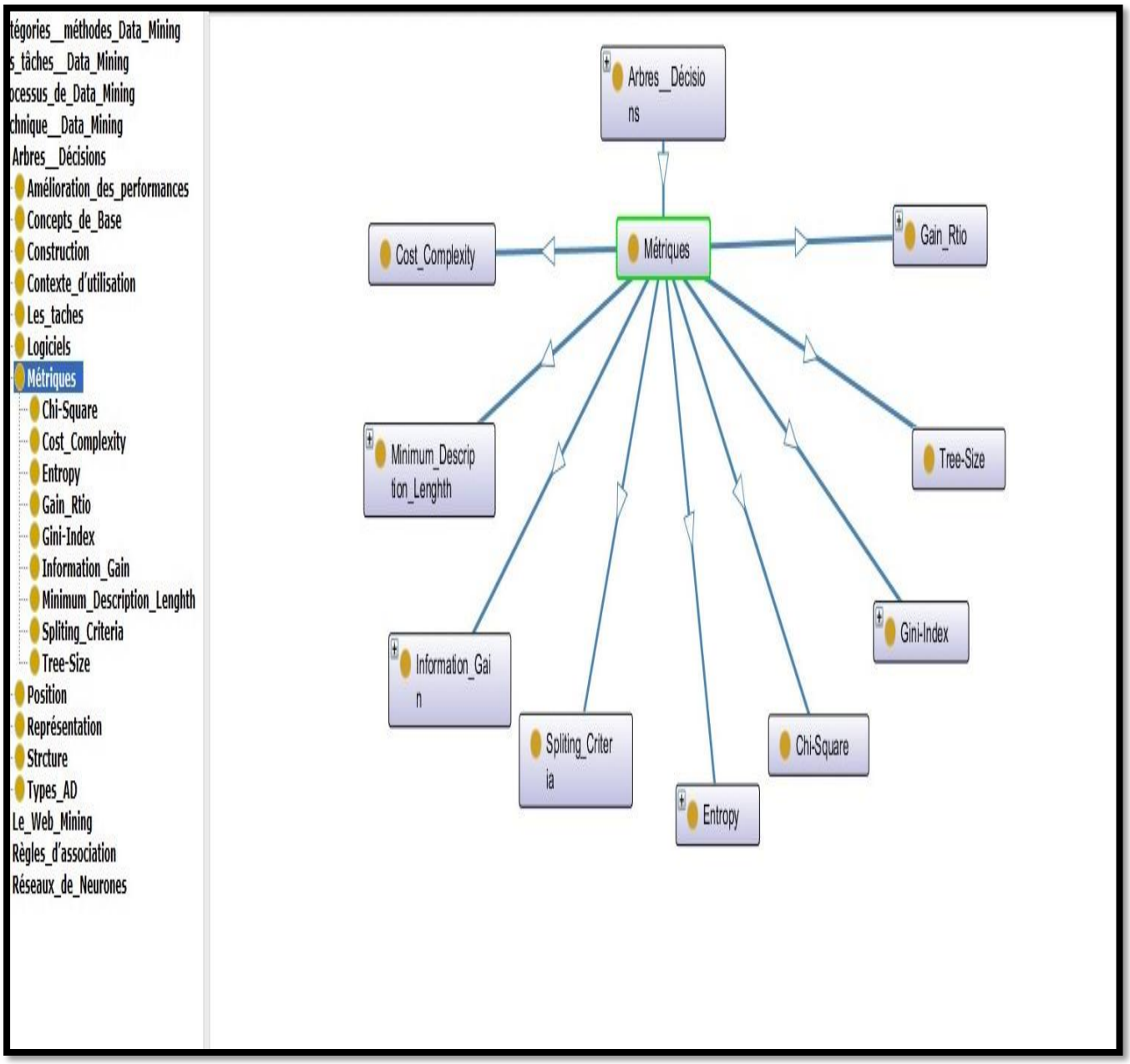



**Figure 4.34** : Présente toutes les sous classe de la classe Métrique.

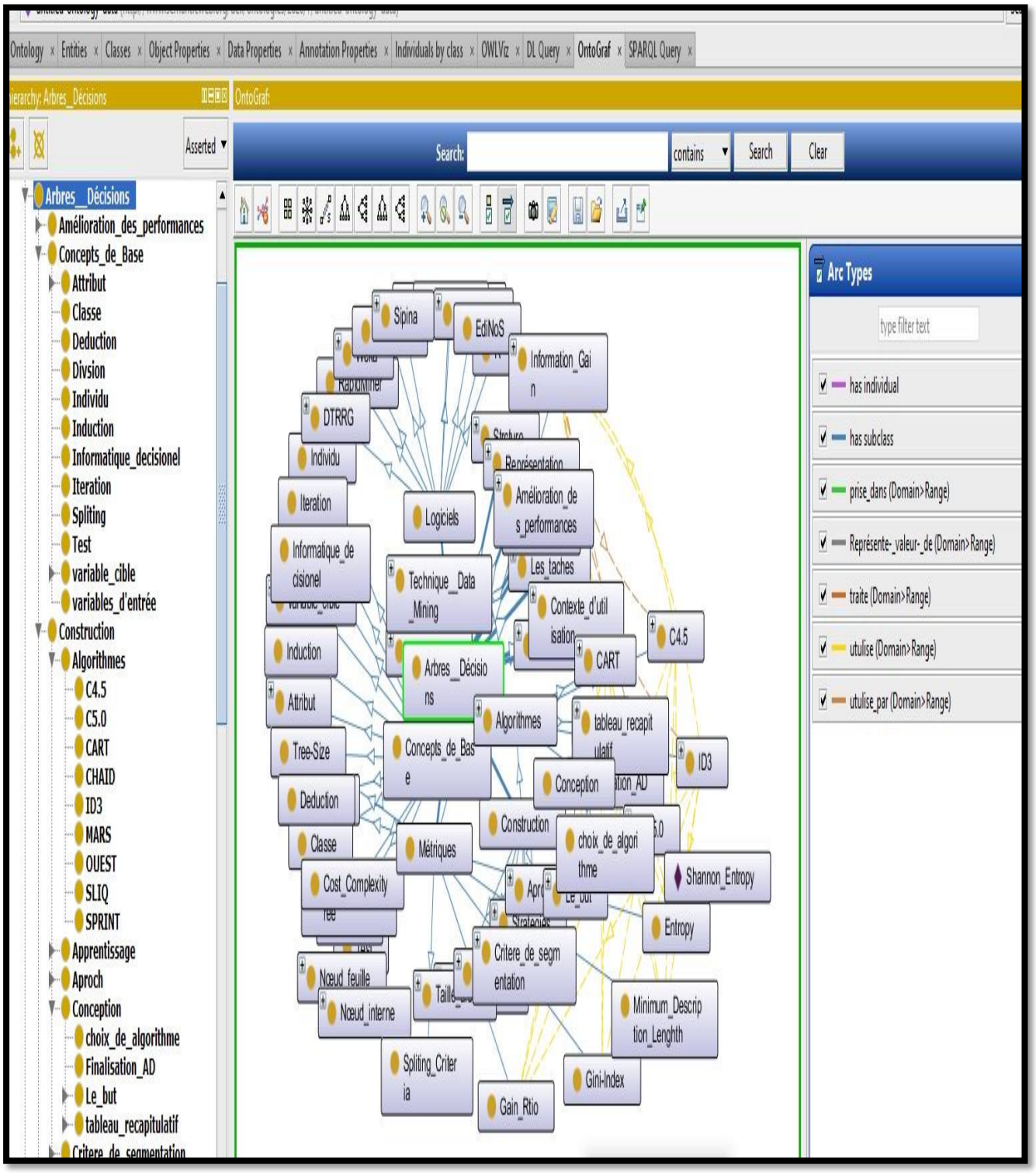



**Figure 4.31 :** Diagramme d'OntoDTA.owl.

## 4.6Génération du code OWL:

L'outil Protégé a été conçu pour dégager et libérer le développeur de la complexité du codage, même pour implémenter une petite ontologie, celle–ci va prendre plusieurs lignes du codes et nécessite un grand effort, ce que nous pouvons constater après la génération du code de notre ontologie, une partie de ce code de l'ontologie OntoDT.owl sera présentée dans l'Annexe.

## 4.7 Résultats expérimentaux

Au niveau de cette section, nous discutons les évaluations expérimentales, ainsi que les tests appliqués à notre ontologie. Les tests d'évaluation sont réalisés sur une machine avec un processeur Intel® Core™ i3, 2,30GHz, une mémoire de 4 Go sous Windows 7(64 bit). Nous avons utilisé l'éditeur d'ontologie **Protégé 5.1.0** (avec le raisonneur FaCT++ ).

### 4.7.1 Contribution au domaine d'arbre de decision :

Avec notre ontologie **OntoDTA.owl**, on a proposé une architecture unifiée et standard basée sur les ontologies, pour les termes les plus utilisés en Arbre de decision, là où il sera facile pour un chercheur d'arbre de decision d'utiliser et de comprendre ces termes.

## 4.8 Cohérence du modèle d'ontologie avec Fact++ et RDF Validator :

### 4.8.1 : Validation par FaCT ++ :

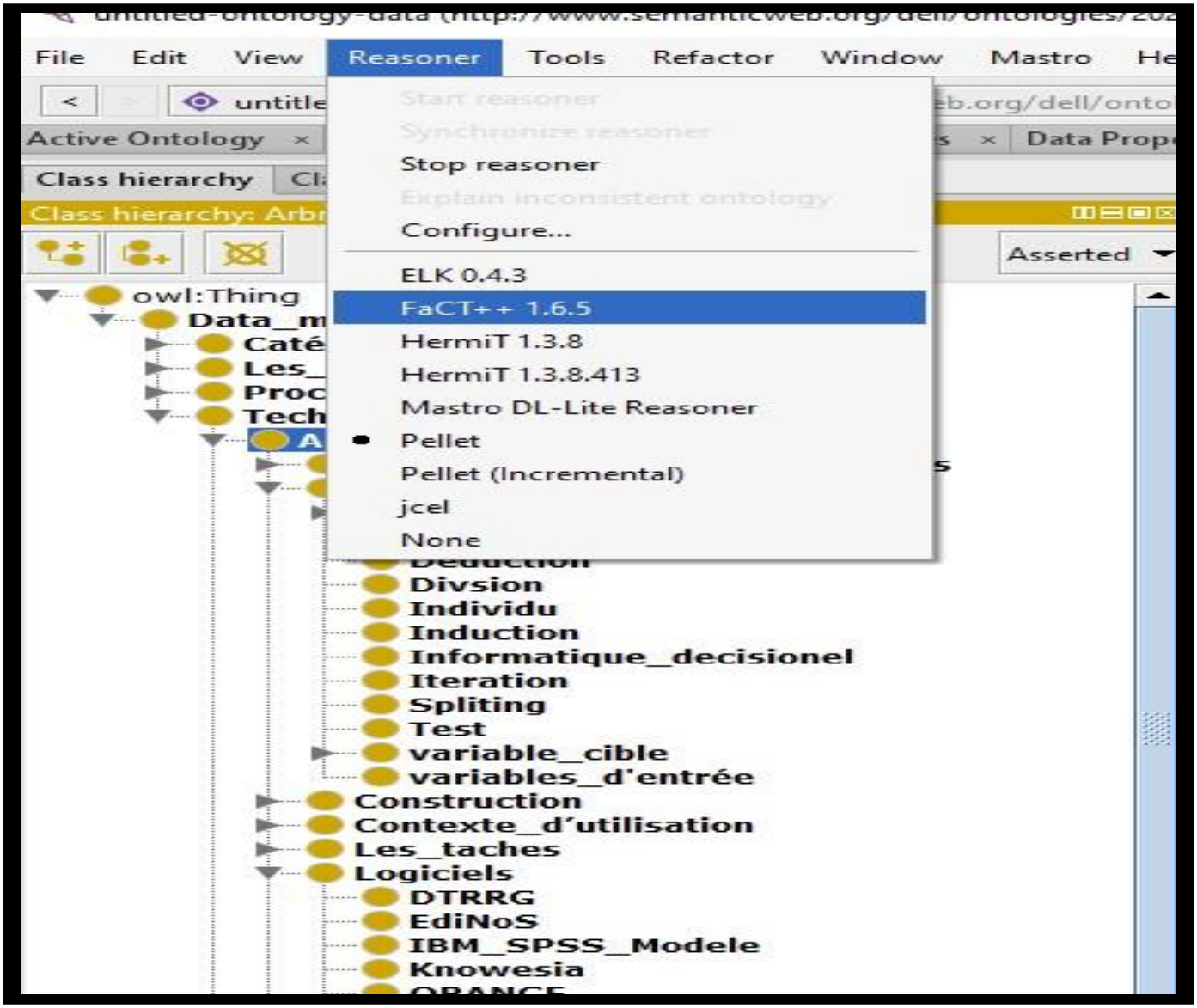



**Figure 4.32** : Début de validation par Fact ++

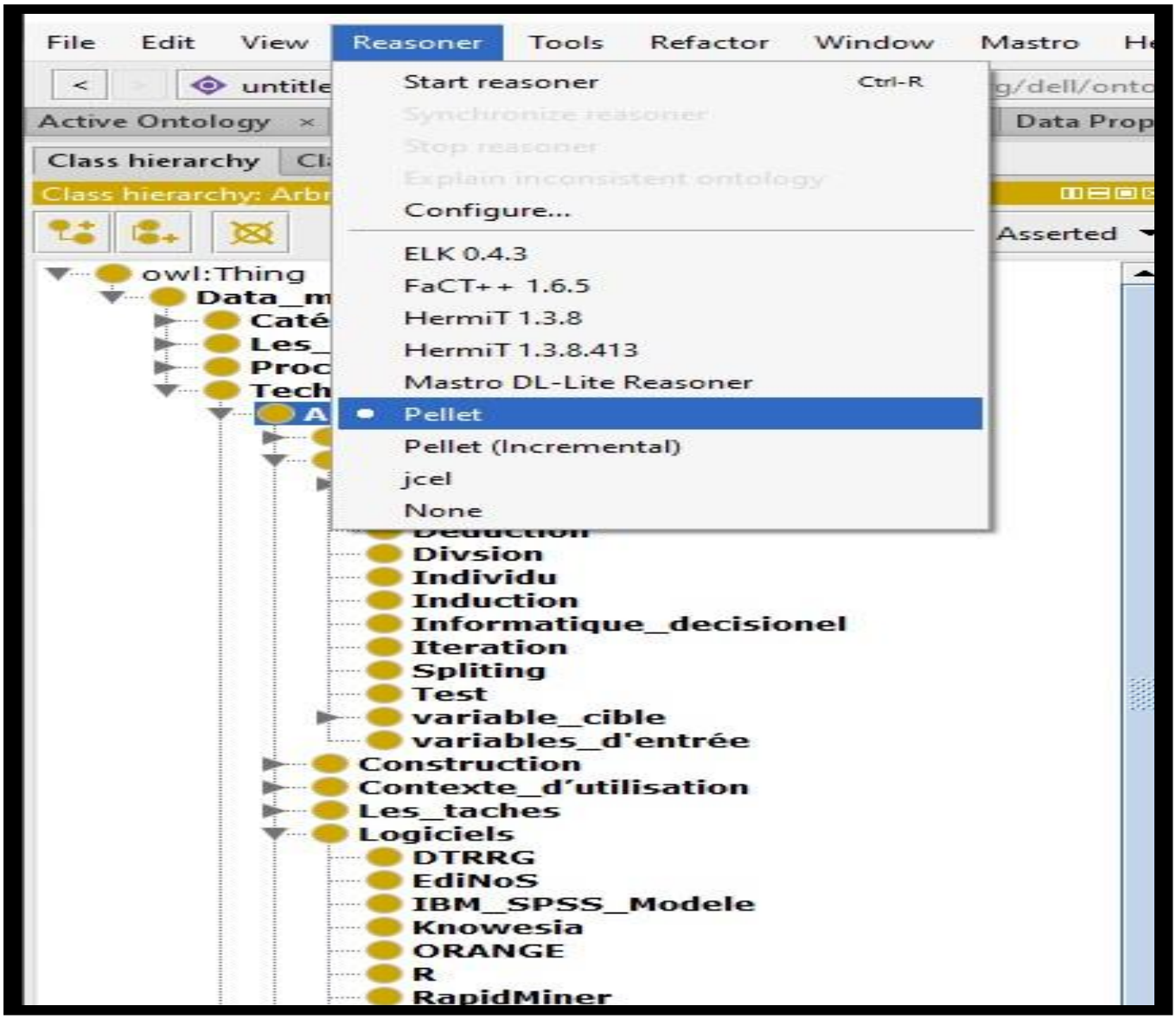



**Figure4. 33** : : Lancement de raisonneur Pellet

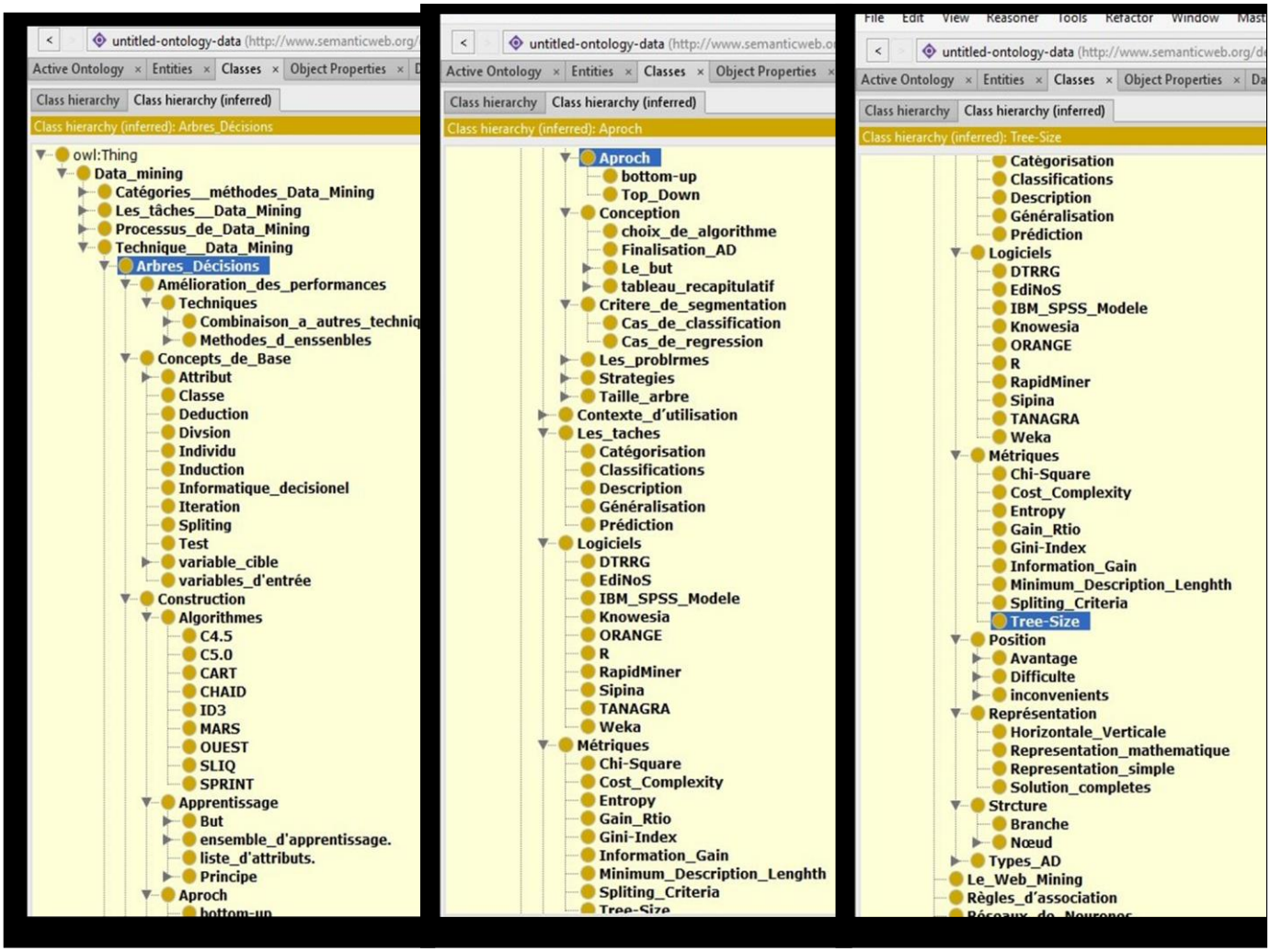



**Figure 4.34** : Classes vérifiées  par le moteur d'inférence Pellet

### 4.8.2 Validation par RDF Validator

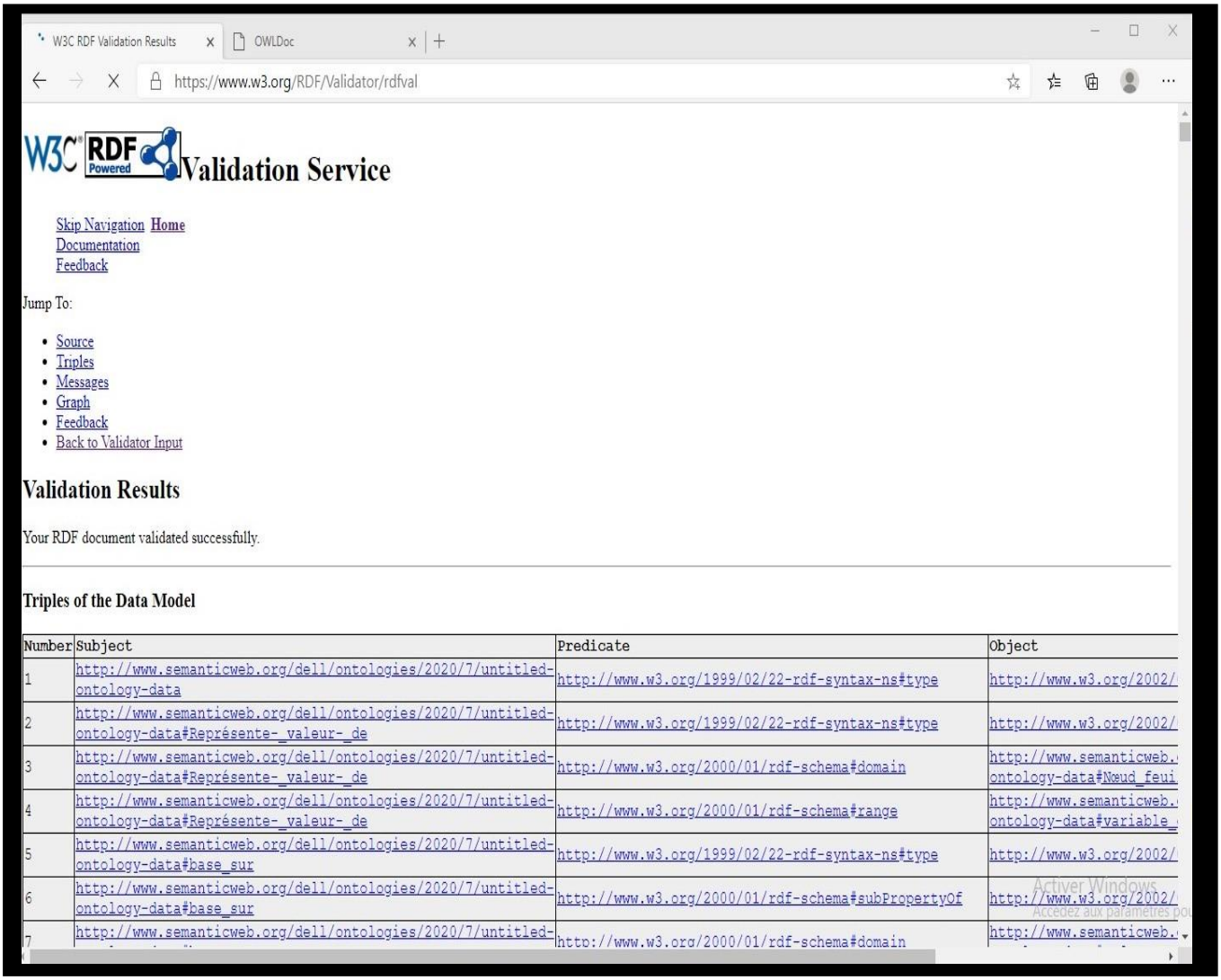



**Figure 4.35** : Validation de notre ontologie « OntoDTA.owl » par le validateur RDF du W3C

## 4.9 Evaluation Métriques

Dans notre cas, nous utilisons les paramètres statistiques d'ontologie du logiciel Protégé (tels que le nombre de classes, axiomes, propriétés, relations sémantiques et individus ...). Les valeurs de ces paramètres statistiques pour les ontologies OntoDTA est le suivant :

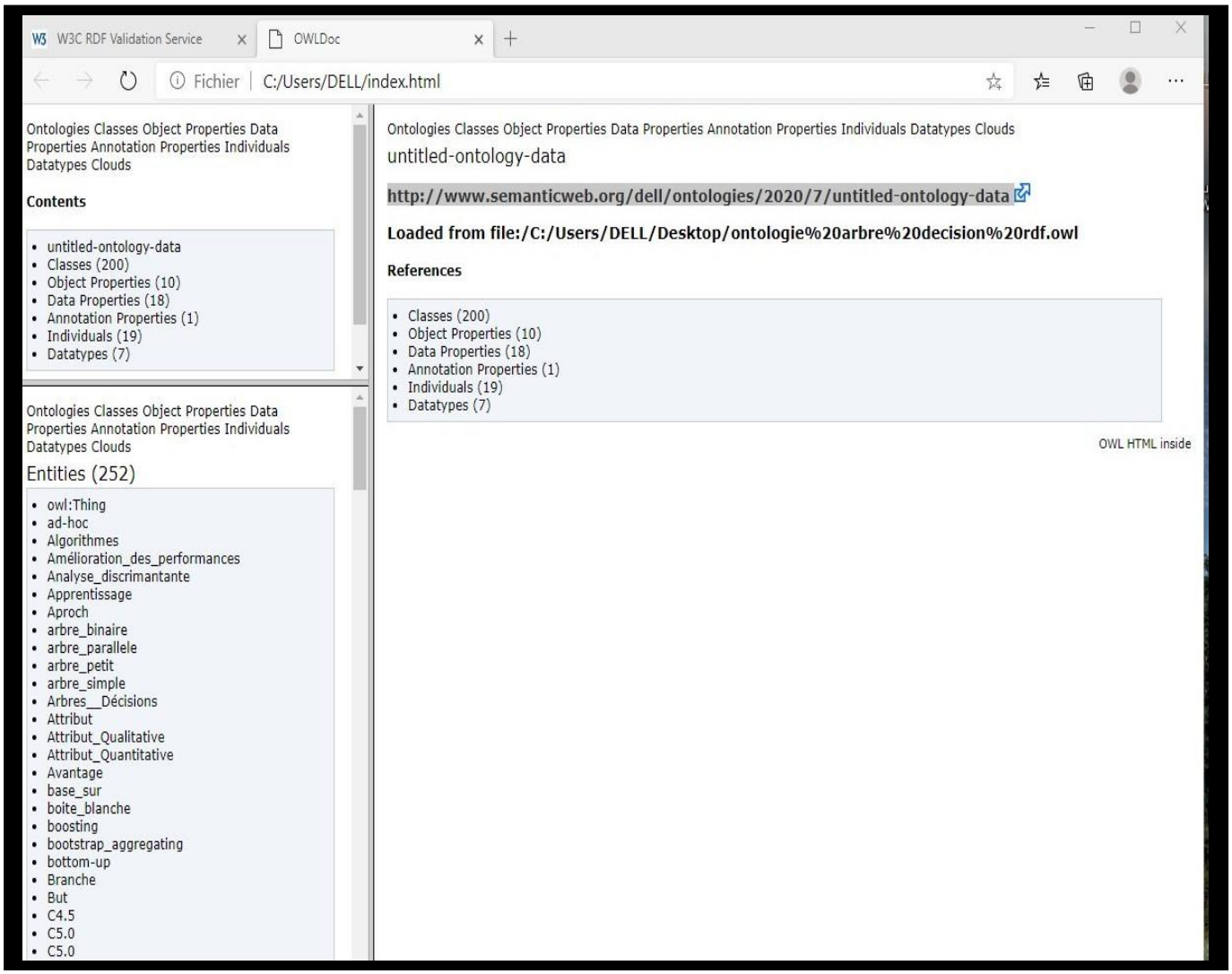



**Figure4. 36** : Métriques statistique pour l'ontologie OntoDTA.owl

## 4.10 Evaluation Sémantique

la validation sémantique garantit que notre ontologies représente les domaines de connaissances définis dans la phase de spécification (chapitre3). Afin de vérifier et de valider notre  ontologies en ce qui concerne les différentes questions de compétence posées dans la phase de spécification de besoin.

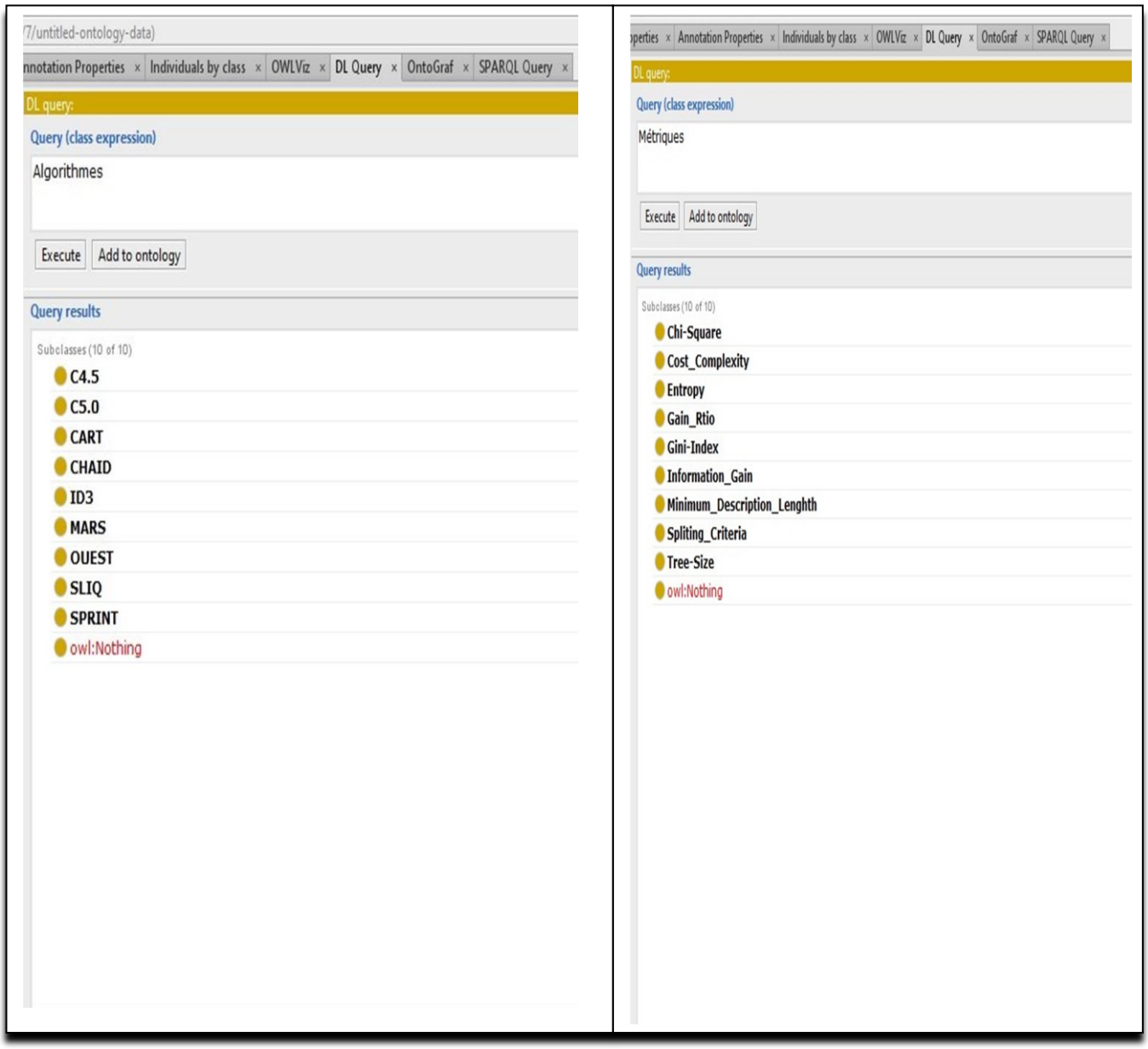



**Figure4. 37** : Réponses des requêtes DL aux questions de compétence 8 et 9dans le domaine des arbres de décision.

On a testé notre ontologie selon les critères de Gruber [Gruber et al., 1993] par le raisonneur Fact++ installé dans Protégé 5 et par le validateur RDF du W3C qui nous permettent donc également de s'assurer que notre document OWL respecte la syntaxe de RDF, ce qui donne déjà une première indication de la validité de notre ontologie qui nous permette de valider la cohérence du modèle associé à l'ontologie. Donc, notre ontologie est caractérisée par:

- La clarté et l'objectivité des définitions, qui doivent être indépendantes de tout choix d'implémentation.
- L'extensibilité.
- Pas de cycle (c'est-à-dire de définition en boucle).
- Pas de redondance de concepts ou de relations.
- Chaque hiérarchie est bien connexe.
- Pas de concepts ou de relations isolées des autres (sans aucun sens).
- Vocabulaire minimum : expressivité maximum de chaque terme.

**4 .11 Conclusion :**

Dans ce chapitre nous avons présenté l'implémentation de notre ontologie. Nous avons tout d'abord présenté l'environnement de développement ainsi que les différents outils utilisés, et nous avons donné une description détaillée de notre ontologie à travers des fenêtres de capture qui représentent les interfaces de ce dernier, qui sont conçues de manière à être conviviales et simples d'utilisation. Cette étape nous a aussi permis de nous familiariser avec les outils utilisés pour le développement d'ontologie.

# *Conclusion Générale et Perspective*

## Conclusion Générale

Dans le cadre de ce mémoire se voulant essentiellement une contribution à la construction d'une ontologie d'arbre de décision.

Pour cela, nous avons d'abord, effectué une étude bibliographique, sur data Mining ,l'arbre de décision, les ontologies et les outils dédient à la représentation des formats XML, RDF, OWL.

Notre contribution réside à la construction d'une ontologie d'arbre de décision. Pour ce faire, nous avons eu recours à un processus basé sur certains travaux intéressants, trouvés dans la littérature, notamment la méthodologie METHONTOLOGY. Bien évidemment, nous étions guidés dans notre travail par plusieurs principes largement acceptés par la communauté des ontologistes. Une fois l'ontologie conceptuelle mise au propre, nous avons passé à sa formalisation en nous appuyant sur un raisonneur Fact++ et son opérationnalisation avec l'outil PROTÉGÉ qui nous a permis de générer automatiquement le code OWL d'ontologie. Enfin, des tests de consistance et de classification ont été appliqués sur l'ontologie opérationnelle par le biais du raisonneur Fact++ et par validateur RDF du W3C.

À travers tous ce que nous avons réalisés dans ce mémoire, nous pouvons dire que le processus suivi pour la construction de notre ontologie, nous a permis de réussir finalement à construire une ontologie pour le domaine d'arbre de décision qu'on a baptisée **OntoDTA.owl**

## Perspectives

Finalement, nous envisageons comme perspectives du travail réalisé dans ce mémoire :

- L’évaluation de l’ontologie d’arbre de décision est limitée par le manque d’un vrai expert de domaine. Son point de vue concernant le contenu de l’ontologie a un impact très important pour vérifier sa complétude.

Il est peu probable qu’**OntoDTA** soit suffisante pour représenter toutes les connaissances de ce domaine qui évolue sans cesse. Il s’agit donc d’étendre cette ontologie et suivre son évolution lors de la configuration des services qui doivent s'adapter à l'évolution des besoins des chercheurs , ou encore de nouvelles logiciels des nouveaux algorithmes  qui peuvent intervenir dans le domaine d’arbre de décision.

.

# Bibliographie

**[Zig01]** D. A. Zighed, Y. Kodratoff, and A Napoli, "Extraction de connaissance à partir d'une base de donnée," Bulletin AFIA'01, 2001.

**[Zig00]** D. A. Zighed, G. Duru, and J. P. Auray, "Sipina, méthode et logiciel," Lacassagne, 2000.

**[Kod98]** Y. Kodratoff, "techniques et outils de l'extraction de connaissances à partir des données," Signaux, vol. 92, pp 38–43, Mars 1998.

**[Fay96]** U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "From data mining to knowledge discovery in databases," Dans aimag KDD overview, pp 1–34, 1996.

**[Lie07]** J. Lieber, Fouille de données : notes de cours, 2007.

**[Bentaher,2015]** Thèses Master .2015 université Bechar , encadrer par Dr Benali Khale

**[Agr93]** R. Agrawal, T. Imielinski, and A. Swami, "Mining association rules between sets of items in large database," Proceedings of the ACM SIGMOD International Conference on Management of Data, Washington, DC, pp 207–216, May 26–28, 1993.

**[Kan03]** M. Kantardzic, "Data Mining–Concepts, Models, Methods,and Algorithms," IEEE Press, Piscataway, NJ, USA, 2003.

**[Tuf05]** S. Tufféry, Data mining et statistique décisionnelle, l'intelligence dans les bases de données, Groupe bancaire Français, Universités de Rennes 1 et paris-Dauphine, 2005.

**[Tuf02]** S. Tufféry, Data mining et scoring, Bases de données et gestion de la relation client, Groupe bancaire Français, Universités de Rennes 1 et paris-Dauphine, 2002.

**[Bac99]** V. Bachelet, Métaheuristiques parallèles hybrides : Application au QAP. PhD thesis, USTL LIFL France, 1999.

**[Alain, 2007]** : Alain Girard, Exploration D'un Algorithme Génétique Et D'un Arbre De Décision À Des Fins De Catégorisation, Université Du Québec, AVRIL 2007.

**[Bachimont et al, 2004]** : B. Bachimont. Art et sciences du numérique : ingénierie des connaissances et critique de la raison computationnelle. Mémoire d'habilitation à diriger des recherches, Université Technologique de Compiègne, Janvier 2004.

**[Bahia, 2013]** : Bahia Zineb, Conception et utilisation d'une ontologie pour la description de Data Mining, Université de Béchar, Licence en Informatique (LMD), Option: Systèmes d'Informations Avancés (SIA), Juin 2013.

**[Barakat, 2011]** : Barkat Abdelbasset, Une approche basée agent pour le processus génération d'ontologie de domaine, Université Mohamed Khider - Biskra, Magister en Informatique Option: Intelligence Artificielle et Systèmes Distribués, 2011.

**[Brad 01]** : N. Bradley, The {XML} Companion, Addison-Wesley Professional Publisher, 2001.

**[Ben Hebireche, 2012]** : Ben Hebireche Halima, Proposition d'une ontologie formelle pour la modélisation et la simulation intelligente, thèse de magister en informatique : Technologie de l'Information et de Communication (TIC), Université Kasdi Merbah Ouargla, soutenu publiquement le 28/06/2012

**[Bentahar et al, 2013]** : Bentahar Aicha et Mebrouki Fatima Zahra, Construction d'une ontologie médicale -Application au domaine de la pédiatrie -, Université de Béchar, Licence en Informatique (LMD), Juin 2013.

**[Bénédicte, 2008]** : Bénédicte BRIAND, Construction D'arbres De Discrimination Pour Expliquer Les Niveaux De Contamination Radioactive Des Végétaux, Universite Montpellier II, 25 Avril 2008.

**[Brahimi ,2014]** : BRAHAMI Menaouer, Conception et Expérimentation d'une nouvelle méthode booléenne de cartographie des connaissances guidée par data mining, Thèse De Doctorat En Informatique Option : Informatique & Automatique, Université d'Oran, 15/05/2014.

**[Berry ,1997]** : M. J. BERRY, G. S. LINOFF, Data Mining Techniques Second Edition, 1997

**[Bernard, 2010]** : Bernard Espinasse, Introduction aux Ontologies, l'Université d'Aix-Marseille, 2010.

**[Bertrand, 2008]** : Bertrand Liaudet, Cours De Data Mining, EPF – 4/ 5ème année - Option Ingénierie d'Affaires et de Projets – Finance, 2008.

**[Bouarroudj ,2010]** : Samia Bouarroudj, Raisonnement Sur Une Ontologie Enrichie Par Des Règles SWRL Pour La Recherche Sémantique D'images Annotées, Mémoire De Magistère, Universite 20 Aout 1955 Skikda ,2010.

**[Bougchiche 2007]** : Bougchiche Lilia, Vers une ontologie pour le dispositif d'interaction, 2007. Mémoire de Magister en informatique Ecole Nationale Supérieure d'Informatique E.S.I Oued- Smar, Alger ,2007

**[Bouza et al, 2008]** : Amancio Bouza, Gerald Reif, Abraham Bernstein, Harald Gall, SemTree: Ontology-Based Decision Tree Algorithm for Recommender Systems, Department of Informatics University of Zurich, 2008.

**[Chami ,2010]** : Chami Djazia, Une plate forme orientée agent pour le data mining, Magister en informatique Spécialité : Sciences et Technologies de l'Information et de la Communication (STIC), Université HADJ LAKHDAR – BATNA, 2010.

**[Charbel et al, 2004]** : Georges El Helou et Charbel Abou khalil, Data Mining Techniques d'extraction des connaissances, Université de Panthéon-Assas Paris II , Projet soutenu le 16 février 2004

**[Dieng et al., 2001]** : Rose Dieng-Kuntz, Olivier Corby, Fabien Gandon, Alain Giboin, Joanna Golebiowska, Nada Matta, Myriam Ribière, "Méthodes et outils pour la gestion des connaissances : une approche pluridisciplinaire du knowledge management". Dunod Edition Informatiques, Séries Systèmes d'Information, 2001.

**[Fernandez et al ,1997]** : Fernandez M., Gomez-Perez A. Juristo N. METHONTOLOGY, " from Ontological art towards Ontological engineering", in Proceedings of the Springs Symposium Series on Ontological Engineering (AAAI'97), AAAI Press , 1997.

**[Frédéric, 2015]** : Frédéric Santos, Arbres de décision, CNRS, UMR 5199 PACEA, 27 mars 2015.

**[Heijst et al., 1997]** : G. van Heijst, A. Th. Schreiber and B. J. Wielinga, "Using explicit ontology in KBS development". International Journal of Human-Computer Studies, Vol. 46 (No. 2/3), 1997.

**[Gajderowicz et al, 2010]** : Bart Gajderowicz, Alireza Sadeghian, Ontology Granulation Through Inductive Decision Trees, Ryerson University, Computer Science Department, 250 Victoria Street, Toronto, Ontario, Canada ,2010.

**[Goméez, 2004]** :A Goméez-Pérez, Mariano Ferndndez- Lpéez and Oscar Corcho "Ontological Engineering with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web" Springer 2004.

**[Gruber, 1993]** : GRUBER T., " A translation approch to portable Ontology specifications". Knowledge Acquisition, 5(2), 1993.

**[Guarino et al, 1995]** : Guarino, Poli, Formal ontology in conceptual analysis and Knowledge Representation, Special issue of the International Journal of Human and Computer Studies ,1995.

**[Guarino 1998]** : N. Guarino. "Formal ontology and information systems . In N. Guarino, editor, Proceedings of FOIS'98, IOS Press, Amsterdam, 1998.

**[Lachiche 2008]** : N. Lachiche, Apprentissage automatique Arbres de décision, 2008.
[Loraine,2008] : Loraine Marcheix , Conception d'une ontologie a partir d'un thesaurus spécialise dans le domaine de l'archéologie et des sciences de l'antiquité, thèse de le Master II professionnel de Gestion de l'Information et du Document : Gestion des Connaissances,

Université Vincennes – Saint-Denis, Paris 8, soutenu 2008

**[Mitskos et al ,2010]:** Mitskos Christina et Spinel Jean-Denis, Data-Mining, Université Catholique De Louvain, 2010.

**[Nathalie ,2008]** : Nathalie Hernandez, Ontologies De Domaine Pour La Modélisation Du Contexte En Recherche D'information, L'Université Paul Sabatier de Toulouse, Doctorat de l'Université Paul Sabatier Spécialité Informatique ,2008.

**[Piatestky ,2005]** : G. Piatetsky-Shapiro, Data mining and Knowledge Discovery 1996 to 2005: overcoming the hype and moving from «university» to «business» and «analytics», Data mining and Knowledge Discovery, 15(1), 99-105, 2005.

**[Preux ,2011]** : Ph. Preux, cours : Fouille de données, Université de Lille 3, 26 mai 2011.

**[Uschold et al, 1996**] : Uschold et Gruninger, Malik, Michael, Ontologies: Principles, Methods and Applications, Knowledge Engineering Review, vol.11, n°2, p. 93-136,1996.

**[Uschold ,1995]** : M. USCHOLD et M. KING, "Towards a Methodology for Building Ontologies". In Proceedings of the Workshop on Basic Ontological Issues in Knowledge Sharing at. the International Joint Conference on Artificial Intelligence (IJCAI'1995), 1995

**[Rahmoun ,2011]** : Somia Rahmoun, Méthodes d'apprentissage pour améliorer la QoS d'une flotte de logiciels embarqués, Master en Informatique Option: Modèle Intelligent et Décision(M.I.D), Université Abou Bakr Belkaid– Tlemcen, Présenté le 15 Septembre 2011

**[Sowa 2000]** : Sowa J., "Ontology, metadata and semiotics". In 8th International Conference on Conceptual Structures (ICCS'2000), Springer-Verlag LNCS.2000.

**[Schwander ,2009]** : Olivier Schwander, Etude de critères de séparation pour les arbres de decision, Master 2 Recherche en Informatique, Ecole Normale Supérieure de Cachan, 30 juin 2009.

**[Stéphane, 2014]** : Stéphane Tuffery, Cours De Data Mining, Master 2 Ingénierie économique et financière, Université Rennes 1, 7 février 2014.

**[Taylor et al., 1997]** : Taylor, M., Stoffel, K., and Hendler, J. (1997). Ontology-based Induction of High Level Classification Rules. In SIGMOD Data Mining and Knowledge Discovery workshop proceedings. Tuscon, Arizona, 1997

**[Zhang 2004]** : Shichao Zhang .Charles X.Ling, Qiang Yang, Jianning Wang, Decision Tree with Minimal Costs, Appearing Proceeding of 21 th International Conference on Machine Learning (ICML), Banff, Canada 2004

**[Zhang et al. 2002]** : Jun Zhang, Adrian Silvescu, Vasant Honavar, Ontology-Driven Induction of Decision Trees at Multiple Levels of Abstraction, Artificial Intelligence Research Laboratory,

Department of Computer Science, Iowa State University Ames, Iowa 50011-1040 USA .2002.

# *Annexe*

**Annexe : Une partie de déclaration des class et sous class de notre ontologie de domaine OntoDTA .XML**

<?xml version="1.0"?>

<Ontology xmlns="http://www.w3.org/2002/07/owl#"

xml:base="http://www.semanticweb.org/dell/ontologies/2020/7/untitled-ontology-data"

xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"

xmlns:xml="http://www.w3.org/XML/1998/namespace"

xmlns:xsd="http://www.w3.org/2001/XMLSchema#"

xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"

ontologyIRI="http://www.semanticweb.org/dell/ontologies/2020/7/untitled-ontology-data">

<Prefix name="" IRI="http://www.semanticweb.org/dell/ontologies/2020/7/untitled-ontology-data"/>

<Prefix name="owl" IRI="http://www.w3.org/2002/07/owl#"/>

<Prefix name="rdf" IRI="http://www.w3.org/1999/02/22-rdf-syntax-ns#"/>

<Prefix name="xml" IRI="http://www.w3.org/XML/1998/namespace"/>

<Prefix name="xsd" IRI="http://www.w3.org/2001/XMLSchema#"/>

<Prefix name="rdfs" IRI="http://www.w3.org/2000/01/rdf-schema#"/>

<Declaration>

```
<Declaration>
    <Class IRI="#choix_de_algorithme"/>
</Declaration>
<Declaration>
    <Class IRI="#Les_arbres_de_classification"/>
</Declaration>
    <Class IRI="#Strategies"/>
</Declaration>
<Declaration>
    <Class IRI="#boite_blanche"/>
</Declaration>
<Declaration>
    <Class IRI="#Information_Gain"/>
</Declaration>
<Declaration>
    <NamedIndividual IRI="#arbre_binaire"/>
</Declaration>
<Declaration>
    <Class IRI="#Méthode_de_la_valeur_critique"/>
</Declaration>
<Declaration>
    <Class IRI="#Attribut_Quantitative"/>
</Declaration>
<Declaration>
```

```
        <Class IRI="#Technique__Data_Mining"/>
    </Declaration>

<Declaration>
        <Class IRI="#Les_problemes"/>
    </Declaration>
    <SubClassOf>
        <Class IRI="#Algorithmes"/>
        <Class IRI="#Construction"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#Amélioration_des_performances"/>
        <Class IRI="#Arbres__Décisions"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#Analyse_discrimantante"/>
        <Class IRI="#Combinaison_a_autres_techniques"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#Apprentissage"/>
        <Class IRI="#Construction"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#Aproch"/>
```

```
        <Class IRI="#Construction"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#Arbres__Décisions"/>
        <Class IRI="#Technique__Data_Mining"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#Attribut"/>
        <Class IRI="#Concepts_de_Base"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#Attribut_Qualitative"/>
        <Class IRI="#Attribut"/>
    </SubClassOf>
    <SubClassOf>
        <Class IRI="#Attribut_Quantitative"/>
    </SubClassOf>
    <SubClassOf>
            <Class IRI="#Position"/>
    </SubClassOf>
</xml>
```