MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE MOULOUDE MAMMERI DE TIZI-OUZOU
FACULTE DE GENIE ELECTRIQUE ET D'INFORMATIQUE
DEPARTEMENT D'INFORMATIQUE

En vue de l'obtention du diplôme de MASTER spécialité INFORMATIQUE

MEMOIRE

DE FIN D'ETUDES

Thème

Développement d'une ontologie pour la Sécurité Informatique

Proposé et Dirigé Par :
M^r^ SIMOHAMMED Malik
M^lle^ ILTACHE Samia

Présenté Par :
HAFSI Meriem
GBOBIA Arnaud Koudou

Année Universitaire : 2012 / 2013

Remerciements

Nous tenons à remercier en premier lieu le TOUT PUISSANT DIEU qui nous a donné la volonté nécessaire, la force et la bonne santé, la patience et le courage de mener à bon terme ce mémoire de fin d'études.

Nous exprimons notre gratitude à notre promoteur Monsieur Malik SI MOHAMMED qui nous a fait l'honneur de diriger ce travail. Nous le remercions pour sa disponibilité et ses conseils qui nous ont été d'un apport considérable tant sur l'aspect scientifique que sur la méthode de recherche. Sans son appui et son expérience, rien n'aurait été possible,

Nous exprimons notre gratitude à Mademoiselle Samia ILTACHE, qui n'a cessé de nous soutenir et nous apporter un éclaircissement tout au long de nos recherches. Sa contribution nous a permis d'avoir une connaissance approfondie de notre travail. Nous la remercions d'avoir placé sa confiance et sa disponibilité à notre service,

Nos vifs remerciements vont également aux membres du jury, pour l'intérêt qu'ils ont porté à notre recherche en acceptant d'examiner notre travail et de l'enrichir par leurs propositions.

Enfin, nous remercions tous ceux qui ont contribué de près ou de loin à la réalisation de ce travail.

Dédicace

Je dédie ce présent travail aux prunelles de mes yeux mes chers parents : Gbobia Goba, Feue Diaby Mahoua, Kouakou Antoinette, Diaby Sata, etc. qui n'ont cessé de me soutenir, m'encourager et me protéger depuis ma naissance.

A mes très chères sœurs Eunice, Anna et Prisca qui m'apporte un soutien considérable qu'importe la distance et à toute ma famille.

A mon amie et en temps ma sœur Youcef Toumi Kaouther.

A mes potes : Benzy, Dof, Donatien (chasseur), Bako, Souley, Loceni, Nadège (Madia), Ibrahim, Crisvel, Bémoulance (kady), Dulcie, Mouna, Melaine, Ludo, Jeremias Sarah, Hugo Santos, Tavares Ivandra, Léonnel, Abiba Koffi.

A tous mes nouveaux.

A mon petit Serge Lewis le baron de la finance.

A mon binôme Meriem Hafsi qui a été d'un grand soutien pour moi.

A toute la promotion ISI 2011, qui a été si sympa durant nos deux années d'études.

Gbobia Arnaud Koudou

Dédicace

Je dédie ce modeste travail à toute personne m'ayant aidée et soutenue tout au long de ce parcours et en particulier :

A celle qui m'a donnée la vie, le symbole de la tendresse, qui s'est sacrifiée pour mon bonheur et ma réussite, à ma mère,

A mon père, école de mon enfance, qui a été mon ombre durant toutes les années d'études, et qui a veillé tout au long de ma vie à m'encourager, à me donner l'aide et à me protéger. Que dieu les garde et les protège,

A ma sœur OUARDIA et mon frère DJAMEL qui m'ont toujours soutenu,

A mes grands-parents qui ont toujours été fiers de moi, ainsi qu'à toute ma famille,

Je ne saurai oublier mes chers amis qui m'ont toujours soutenu et aidé dans les moments difficiles, je les remercie pour leur présence et leur aide,

A mon binôme Arnaud Gbobia Koudou,

Et à toute personne ayant contribué à ce travail de près ou de loin, je les remercie du fond du cœur.

HAFSI Meriem

Table des matières

Table des figures

Figure 4.22 : Exemple 3 de l'exécution d'une requête SPARQL sur Protégé.

Figure 4.23 : Exemple 4 de l'exécution d'une requête SPARQL sur Protégé.

Table des Tableaux

Introduction Générale

Introduction générale

Le développement spectaculaire des technologies de l'information et de la communication a permis aux entreprises de passer très rapidement au monde numérique. Cette évolution plus que remarquable leurs permet de mener à bien les activités qu'elles gèrent au travers d'un système d'information.

Les systèmes d'information, moteurs de croissance et de développement des métiers et services, sont importants, voire même indispensables pour le bon fonctionnement de toute entreprise. Cependant, avec les menaces actuelles et l'ouverture des systèmes d'information sur Internet. Il devient nécessaire de garantir la sécurité de l'ensemble des biens constituant tout système d'information contre les menaces auxquelles il est exposé. Ces menaces peuvent cibler différents actifs de l'entreprise. Parmi elles, nous pouvons citer : la menace liée à l'environnement, les erreurs des utilisateurs, les défauts de conception du matériel informatique, l'attaque informatique, etc. **[14]**

Pour comprendre et appréhender le domaine de la sécurité informatique, il est nécessaire voire capital de le représenter par un modèle adéquat. L'utilisation des ontologies pour la sécurité informatique semble une solution avantageuse pour représenter ses différents aspects ; puisque elles permettent de formaliser le domaine étudié dans tous ses aspects en se basant sur une méthodologie.

L'utilisation des ontologies pour la modélisation de la connaissance ne cesse de croitre, depuis que la communauté du W3C en a fait la plaque tournante pour la sémantisation du Web. L'objectif d'une telle démarche est de permettre l'interopérabilité entre la machine et l'humain. Les ontologies sont actuellement utilisées dans divers domaines comme l'E-learning, le Biomédical, la recherche d'information et Le web sémantique. Leur intégration dans les systèmes devient de plus en plus importante suite aux avantages qu'elles apportent.

Nous proposons d'utiliser les ontologies comme solution au problème concernant la modélisation des connaissances du domaine de la sécurité informatique. Notre travail consiste en une étude générale de la sécurité informatique dans un premier temps. Puis, dans un second temps, nous nous intéresserons à un domaine spécifique qui porte sur la gestion des attaques informatiques au sein d'une entreprise. Enfin, nous développerons une ontologie pour ce domaine, en appliquant une méthodologie de construction d'ontologies.

Pour ce faire, nous avons opté pour le plan suivant :

Chapitre 1 : Dans ce chapitre, nous abordons les différents principes de la sécurité informatique au sein d'une entreprise qui a pour mission de sécuriser le système d'information. Nous parlerons ensuite des méthodes et normes de la gestion des risques. Nous étudierons l'une de ces méthodes qui est la méthode MEHARI 2010.

Chapitre 2 : Nous consacrerons ce chapitre à la présentation des ontologies. Nous commencerons par la définition de la notion d'ontologie. Nous présenterons ensuite les ses différents composants et types d'une ontologie. Puis nous présenterons les principaux formalismes de représentation des connaissances, à savoir les frames, les graphes conceptuels et les logiques de description. Nous dériverons après les méthodologies de construction d'ontologies et les outils nécessaires à leur développement. Et enfin nous présenterons les différents travaux qui ont été réalisés pour représenter différents domaines de la sécurité informatique en utilisant les ontologies.

Chapitre 3 : Dans ce chapitre, nous allons proposer une solution au problème de la représentation des connaissances du domaine de la sécurité informatique, en développant une ontologie qui représentera les connaissances spécifique à un contexte. Le contexte choisi portera sur les attaques informatiques au sein d'une entreprise. Ce chapitre traite la première partie du développement de notre ontologie. Cette partie est constituée des trois premières étapes de la méthode Methontology, à savoir, la spécification, la conceptualisation et la formalisation.

Chapitre 4 : Nous consacrerons ce dernier chapitre à l'implémentation de l'ontologie formalisée suivant le processus de Methontology en utilisant l'éditeur protégé.

Finalement, nous concluons par un récapitulatif de notre travail, la démarche adoptée, notre contribution et une réflexion sur les extensions à apporter à notre modélisation.

Chapitre I

La sécurité Informatique

1.1. Introduction

Au fil des dernières décennies, l'outil informatique a pris une grande place dans les entreprises. Quel que soit leur secteur d'activité, l'informatique joue un rôle prépondérant devenant peu à peu l'outil de référence. La croissance de son utilisation ainsi que son intégration dans différents domaines, a engendré un nouveau risque lié aux menaces qui guettent l'entreprise et ciblent son système d'information par le biais de l'outil informatique. D'où la nécessité d'élaborer une politique de sécurité qui assure la protection des systèmes d'information contre les incohérences, les interruptions et les intrusions, pouvant atteindre ce système d'information. Ces risques pouvant être causés de l'intérieur, tout comme de l'extérieur, ne doivent en aucun cas être négligés par les responsables, sinon les répercussions seraient couteuses pour l'entreprise, tant au niveau informationnel, financier, matériel, qu'humain. Pour éviter cela, plusieurs normes et méthodes ont été proposées afin de gérer les risques régulièrement et protéger les systèmes d'information de tous types de menaces.

Ce chapitre traite la sécurité des systèmes d'information en général, et se compose de deux parties, la première partie introduit les principes fondamentaux de la sécurité informatique et aborde différents points qui sont:

- Généralités sur les systèmes d'information ;
- Sécurité des systèmes d'information ;
- La sécurité informatique et ses différents aspects, ses enjeux ainsi que les différents risques et menaces pesant sur l'entreprise ;
- Les attaques informatiques ;
- Elaboration d'une politique de sécurité ;
- Les normes et méthodes de gestion des risques.

La deuxième partie sera consacrée à la méthode de gestion des risques MEHARI. Nous présenterons ses objectifs et sa démarche, puis nous étudierons en détail sa dernière version,

1.2. Généralités sur la Sécurité Informatique

Le degré d'utilisation des systèmes d'information et l'environnement des technologies de l'information dans son ensemble, ont évolué de façon extraordinaire depuis 1992 **[11]**. Ces évolutions offrent différents avantages significatifs mais requirent également que les développeurs, le gouvernement, les entreprises, les organisations et les utilisateurs qui développent, possèdent, gèrent et utilisent les systèmes d'information, portent une bien plus grande attention à la sécurité car ces systèmes sont vulnérables, tombent en panne, subissent des erreurs d'utilisation et sont attaqués de

l'intérieur comme de l'extérieur. Une approche globale de la sécurité des systèmes est essentielle pour les protéger et réduire leurs vulnérabilités.

Cette première partie du chapitre aborde les généralités sur la sécurité informatique, en définissant d'abord les systèmes d'information et la sécurité des systèmes d'information. Puis nous nous intéresserons à l'aspect technique qui est la sécurité informatique (logique). Nous présenterons ensuite, les différents risques et menaces auxquels sont exposés les systèmes. Puis nous définirons la politique de sécurité et les différentes normes et méthodes qui ont été développées afin d'apporter des solutions à ce problème. Enfin, nous consacreront la deuxième partie pour une étude partielle de la méthode de gestion des risques MEHARI.

1.2.1. Definition du systèmes d'information

Un système d'information (SI) est un ensemble organisé de ressources (matériels, logiciels, personnel, données et procédures) qui permet de collecter, regrouper, classifier, traiter et diffuser de l'information sur un environnement donné. Le système d'information a pour objectif **[04]** :

- Recueil de l'information (sources internes ou externes)
- Traiter l'information (comparaison, rapport, modification, expertise, …) ;
- Faire véhiculer l'information (envoi de données à un serveur, téléchargement,…) ;
- Mémoriser l'information (sauvegarde, …).

Un système d'information est utilisé pour :

- Le lancement des décisions programmées ;
- Superviser les fonctions permettant les connexions entre toutes les entités d'un organisme ;
- Apporter aux responsables des informations qui leur permettront d'agir au moment opportun ;

1.2.2. La sécurité des systèmes d'information

Un système d'information est un élément absolument vital pour l'entreprise, et tout ce qui le menace est potentiellement dangereux. C'est ainsi que la notion de sécurité fût abordée par les responsables du système d'information. **[09]**

La sécurité des systèmes d'information (SSI) apparaît comme tous moyens humains, matériels, juridiques, techniques et organisationnels mis en œuvre pour assurer la protection de l'information. Cette sécurité peut être assurée par une politique qui vise à garantir, rétablir et préserver le fonctionnement du système d'information **[09].**

La commission européenne définit la sécurité des systèmes d'information comme « la capacité d'un réseau ou d'un système d'information de résister, à un niveau de confiance donné, aux évènements accidentels, ou aux actions malveillantes qui compromettent la disponibilité,

l'authenticité, l'intégrité et la confidentialité des données stockées ou transmises et des services connexes que ces réseaux et systemes offrent ou qu'ils rendent accessibles. »**[10]**

Le fait que la sécurité relève à la fois de mesures techniques de protection des systemes informatiques en meme temps que de mesures juridiques de prévention et de dissuasion des délits, nous mene à dire que la sécurité des systèmes d'information couvre la sécurité informatique (logique) mais aussi la sécurité physique et organisationnelle du système.

➢ La sécurité informatique

Un système informatique est la partie automatisée d'un système d'information qui regroupe l'application de gestion et ses éléments d'accompagnement, les logiciels supports et les matériels. Les systèmes informatiques sont au cœur des systèmes d'informations. Ils sont devenus la cible de ceux qui aimeraient accéder aux informations. Assurer la sécurité de l'information implique d'assurer la sécurité des systèmes informatiques. Ainsi, la sécurité informatique peut être définie comme l'ensemble des moyens mis en œuvre pour minimiser la vulnérabilité d'un système informatique contre des menaces accidentelles ou intentionnelles. **[09]**

1.2.3. Objectifs de la Sécurité des systemes d'information

La sécurité des systèmes d'information doit garantir un niveau convenable de **[03]**:

- **Confidentialité :** signifie que l'information ne doit être accessible qu'aux personnes autorisées.
- **Disponibilité :** les informations doivent être mises à disposition des utilisateurs en temps voulu. Dans la sécurité informatique, la disponibilité vise à ce qu'un système soit capable d'assurer ses fonctions sans interruption ou dégradation, au moment même où la sollicitation en est faite.
- **Intégrité :** garantir que les informations sont bien celles que l'on croit être.
- **Non-répudiation :** interdire à une entité de pouvoir nier avoir pris part à une action.

Le problème que la sécurité doit résoudre, réside dans le fait que les ressources (matérielle, logicielle, humaine, etc.) d'une entreprise soient confrontés aux menaces. Qu'elles soient internes ou externes, elles peuvent exploiter leurs vulnérabilités et engendrer la réalisation d'un risque qui aura des conséquences sur l'entreprise.

Un risque est un évènement pouvant se produire à tout moment et que l'on craint. Il peut être vu comme la probabilité d'échec dans l'atteinte d'objectifs. Le risque est une fonction de la menace (Cause potentielle d'incident), la vulnérabilité (faiblesse d'une personne, d'un système informatique, d'une ressource) ainsi que la conséquence qui peut être le résultat de la réalisation du risque. **[13]**

RISQUE = VULNERABILITE x MENACE x CONSEQUENCE

Sécuriser un système d'information s'avère très capital pour une entreprise. Dès lors, des moyens sont employés pour répondre aux besoins de l'entreprise. Et pour cela, il est nécessaire de gérer ses risques en utilisant une méthode de gestion des risques et en élaborant une politique de sécurité**.**

1.2.4. La gestion des risques dans les entreprises

Pour permettre la gestion des risques, La norme ISO 27005 propose un processus qui commence par l'identification des risques au sein d'une entreprise. Ce processus se fait en identifiant les actifs, les menaces ou attaques, les contre-mesures, les vulnérabilités et les conséquences. **[01]**

Les actifs : Une entreprise possède plusieurs actifs. L'actif est tout ce qui peut représenter une valeur ou un enjeu pour l'entité. (Tout élément ayant de la valeur pour l'organisme et nécessitant une protection). Il existe deux types d'actifs au niveau d'une entreprise : actif primaire et l'actif secondaire.

- **Actif Primaire :** C'est la raison d'être de l'entreprise, ce type d'actifs englobe les processus métiers et les informations de l'entreprise.
- **Actif Secondaire :** Tous les autres actifs que possède une entreprise, il existe cinq catégories qui sont : le matériel, le logiciel, le réseau, le personnel et le site.

Les Menaces : c'est l'ensemble des actions de l'environnement d'un système pouvant entraîner des pertes financières. Dans le domaine de la sécurité informatique, elle représente une infraction potentielle de la sécurité, qui peut être active, passive, intentionnelle (attaque) ou accidentelle. Exemple : vol de documents, erreur d'utilisation, attaque informatique, etc.

Les Attaques : activité malveillante qui exploite la vulnérabilité d'un système informatique à des fins non connues par les responsables du système et généralement préjudiciables.

Les contre-mesures : c'est l'ensemble des actions mises en œuvre en prévention de la menace.

Les vulnérabilités : la vulnérabilité est la caractéristique d'un système, d'un objet, ou d'un actif constituant un point d'application potentiel de menace. Exemple : Absence de stratégie de sécurité, utilisation de mot de passe en clair, etc. **[14]**

Les conséquences : la conséquence est le résultat de la réalisation **du risque**, elle peut être une dégradation, une destruction, un dommage, l'atteinte à un des objectifs de la sécurité, ou la perte d'un actif.

1.2.4.a. Le risque informatique

Améliorer la sécurité informatique, c'est gérer les risques liés à l'informatique. En effet, Le risque informatique est lié principalement aux attaques qui ciblent les actifs de l'entreprise en utilisant l'outil informatique. Pour réduire ce risque, il est nécessaire d'identifier les menaces qui pèsent sur les actifs à travers ces attaques. Ces attaques représentent pour une entreprise un souci majeur. Elles font

l'objet de l'exploitation d'une faille d'un système informatique à des fins non connues par l'utilisateur du système. Elles peuvent être destructrices et amener une entreprise à déposer le bilan. Afin de contrer ces attaques, il est indispensable de connaitre les principaux types d'attaques afin de mettre en œuvre des dispositions préventives. Les systèmes informatiques mettent en œuvre différentes composantes, allant de l'électricité pour alimenter les machines au logiciel exécuté via le système d'exploitation et utilisant le réseau. Les attaques peuvent intervenir à chaque maillon de cette chaîne, pour peu qu'il existe une vulnérabilité exploitable.

1.2.4.b. Les attaques

Il existe plusieurs types d'attaques informatiques, elles se différencient les unes des autres selon le type d'actif visé, la méthode exploitée, et l'objectif fixé. **[02]**

- **Les attaques applicatives**

Ce type attaques exploite les vulnérabilités des logiciels et applications. **[19]** Parmi elles nous pouvons citer :

- **Le dépassement de tampon (Buffer Overflow)** qui consiste à mettre en mémoire plus d'informations que celle-ci n'est disposée à en recevoir.
- **L'exploitation d'une porte dérobée (Backdoor)** qui est une faille présentée par un système, exemple : port ouvert inutilement.
- **Injection de Code** consiste à injecter du code supplémentaire dans un programme afin d'exécuter des taches supplémentaires, exemples : l'injection SQL et la faille XSS.

- **Les attaques par déni de service**

C'est le type d'attaques le plus répandu et aussi le plus difficile à éliminer. Le DoS consiste à saturer les ressources d'un système ou bien la charge du processeur d'une machine, la quantité de mémoire utilisée par un processus ou encore la bande passante disponible pour atteindre une machine. Pour plus d'efficacité, l'attaque peut être redistribuée. Cependant, la facilité de mise en œuvre des attaques DoS et leurs dégâts potentiellement très graves retiennent toute l'attention des administrateurs de la sécurité. Nous pouvons citer comme exemples d'attaque par déni de service parmi tant d'autres :

- **Déni de service distribué (DDoS) :** Le déni de service distribué consiste à attaquer une ressource en utilisant un botnet qui est un ensemble de machines contrôlées par une seule personne. **[20]**
- **SYN Attaque (ou TCP SYN flooding) :** exploite les trois étapes de connexion du protocole TCP. Son rôle est de rendre indisponible un service TCP offert sur une machine en envoyant un grand nombre de requêtes au serveur ciblé. Le principe de cette attaque est de créer des connexions semi-ouvertes sur la machine cible afin de remplir la file d'attente ou sont enregistrées les requêtes d'ouvertures de connexions. **[02]**

➢ L'attaque de l'homme du milieu

L'attaque de l'homme du milieu (Man In The Middle) a pour but d'intercepter les communications entre deux parties, sans que ni l'une ni l'autre ne puisse se douter que le canal de communication entre elles a été écouté ou compromis**. [02]**

➢ Le Phishing (Hameçonnage)

Cette méthode consiste à entrainer l'internaute à divulguer des informations confidentielles en usant d'un hameçon fait de mensonge et de contrefaçon électronique. Le moyen le plus répandu est celui d'un mail usurpant l'identité d'une structure et contenant un lien vers un faux site où l'on demandera de confirmer le numéro de compte sous un prétexte plus ou moins vraisemblable. Le courriel non sollicité (spam) en est un exemple. **[02]**

➢ Le Sniffing (Analyse du réseau)

Cette technique permet d'écouter le réseau en utilisant un logiciel appelé sniffer (analyseur de trames ou sniffer). C'est un dispositif permettant d'écouter le trafic d'un réseau, c'est-à-dire de capturer les informations qui y circulent. En effet, dans un réseau non commuté, les données sont envoyées à toutes les machines du réseau. Toutefois, dans une utilisation normale les machines ignorent les paquets qui ne leur sont pas destinés, Ainsi, en utilisant l'interface réseau dans un mode spécifique, il est possible d'écouter tout le trafic passant par un adaptateur réseau (une carte réseau Ethernet, une carte réseau sans fil). **[19]**

➢ L'usurpation d'identité

C'est une technique qui consiste à se faire passer pour une autre personne en falsifiant son adresse IP. Plusieurs techniques sont utilisées pour ce faire parmi elles, nous pouvons citer :

- **ARP Spoofing :** Cette attaque redirige le trafic réseau de plusieurs machines vers la machine du pirate. Elle s'effectue sur le réseau physique des victimes et corrompt le cache de la machine victime. Le pirate recevra sur sa machine tout le trafic à destination de la passerelle, ce qui lui permettra d'écouter le trafic et le modifier, puis router les paquets vers la véritable destination.
- **DNS Spoofing :** Le pirate utilise les faiblesses du protocole DNS et de son implémentation sur les serveurs de noms de domaine pour rediriger des internautes vers des faux sites. Son objectif est de faire correspondre l'adresse IP d'une machine qui est sous son contrôle à l'URL réelle d'une machine publique. **[20]**

➢ Les attaques virales

Les pirates utilisent certains logiciels malveillants (malwares) qui se répandent en général sur le réseau. Ces logiciels sont des programmes ou une partie de programme visant à la neutralisation ou la destruction des éléments logiciels nécessaires au bon déroulement d'un système. Ces personnes

réalisent leurs attaques soit par accès direct à l'ordinateur attaqué, soit caché dans un courriel ou sur un site Web attrayant. **[19]**

- **Le Keylogger (Enregistreur de frappes)** : c'est un programme permettant d'enregistrer les frappes de touches du clavier (telles que le mot de passe, numéro de compte, mail confidentiel, login, etc.) et de les sauvegarder, à l'insu de l'utilisateur dans un fichier (.log). Ces enregistreurs sont capables d'enregistrer les URL visités, les courriels, fichiers ouverts, etc.
- **Le Virus :** c'est un petit programme qui se reproduit en s'incorporant à un autre programme (du système ou d'une application). Une fois dans le système, il peut maintenant se propager. Il peut avoir comme effets de nuire en perturbant le fonctionnement de l'ordinateur infecté en apportant les changements indésirables. **[20]**
- **Le Ver (Worm) :** est un programme autonome se reproduisant et se propageant à l'insu des utilisateurs. Contrairement aux virus, un ver n'a pas besoin de logiciels hôte pour se reproduire.
- **Le Cheval de Troie (Trojan)** : c'est programme caché dans un autre programme qui s'exécute au démarrage du programme hôte.
- **Bombe logique :** C'est une partie d'un programme malveillant qui est installé et reste sur un système jusqu'à ce qu'un instant ou un évènement survienne, ou encore que certaines conditions soient réunies, pour provoquer des effets dévastateur en son sein. Ce mode exploite principalement des informations comme la date système, le lancement d'une procédure, l'entrée d'une chaîne de caractères. **[20]**

1.2.5. politique de sécurité

Le système d'information est essentiel à l'activité de l'organisation, son utilisation inappropriée ou son mal fonctionnement peut menacer l'existence de l'organisation. En analysant et définissant les risques, on peut construire une politique de sécurité du système d'information, et pour cela, il est necessaire de définir le cadre d'utilisation des moyens informatiques.

➢ Definition de la politique de sécurité

Une politique de sécurité est un énoncé général dicté par les cadres supérieurs décrivant le rôle de la sécurité au sein de l'entreprise afin d'assurer les objectifs d'affaire. Elle définit un certain nombre de règles, de procédures et de bonnes pratiques permettant d'assurer un niveau de sécurité conforme aux besoins de l'organisation. Un tel document doit nécessairement être conduit comme un véritable projet associant des représentants des utilisateurs et conduit au plus haut niveau de la hiérarchie, afin qu'il soit accepté par tous. Lorsque la rédaction de la politique de sécurité est terminée, les clauses concernant le personnel doivent leur être communiquées, afin de donner à la politique de sécurité le maximum d'impact.

Définition selon RFC 2196 *« une politique de sécurité est une déclaration formelle des règles auxquelles doivent se conformer les personnes recevant un droit d'accès au capital technologique et informatif d'une entreprise »* **[18]**

- **La mise en place d'une politique de sécurité**

La sécurité des systèmes informatiques se cantonne généralement à garantir les droits d'accès aux données et ressources d'un système en mettant en place des mécanismes d'authentification et de contrôle permettant d'assurer que les utilisateurs des dites ressources possèdent uniquement les droits qui leur ont été octroyés. La mise en œuvre d'une politique de sécurité se fait selon quatre étapes :

- Identifier les besoins en termes de sécurité, les risques informatiques pesant sur l'entreprise et leurs éventuelles conséquences,
- Elaborer des règles et des procédures à mettre en œuvre dans les différents services de l'organisation pour les risques identifiés,
- Surveiller et détecter les vulnérabilités du système d'information et se tenir informé des failles sur les applications et matériels utilisés,
- Définir les actions à entreprendre et les personnes à contacter en cas de détection d'une menace.

La sécurité informatique de l'entreprise repose sur une bonne connaissance des règles par les employés, grâce à des actions de formation et de sensibilisation auprès des utilisateurs, mais elle doit aller au-delà et notamment couvrir les champs suivants :

- Un dispositif de sécurité physique et logique, adapté aux besoins de l'entreprise et aux usages des utilisateurs.
- Une stratégie de sauvegarde correctement planifiée.
- Une procédure de management des mises à jour.
- Une stratégie de sauvegarde correctement planifiée.
- Un plan de reprise après incident.
- Un système documenté à jour.

1.2.6. Les normes et méthodes

Avec l'avenue des technologies et de l'informatique, les entreprises ont ressenti le besoin d'établir des référentiels de sécurité pour établir des relations de confiance. Ces référentiels ont évolué vers des normes internationales ISO de sécurité. Afin d'assurer une bonne démarche de sécurisation des SI des méthodes de sécurité ont été introduites. Ces méthodes sont basées sur les normes ISO/IEC qui sont destinées à protéger l'information et à assurer la sécurité du système d'information.

1.2.6.a. Les normes

Une norme est un document de référence basé sur un consensus couvrant un large intérêt industriel ou économique et établi par un processus volontaire.

L'organisation internationale de normalisation ISO (International Organization for Standardization) : c'est un organisme de normalisation international composé de représentants d'organisations nationales de normalisation de 164 pays. Cette organisation créée en 1947 a pour but de produire des normes internationales dans les domaines industriels et commerciaux appelées normes ISO. Elles sont utiles aux organisations industrielles et économiques de tout type, aux gouvernements, aux instances de réglementation, aux dirigeants de l'économie, aux professionnels de l'évaluation de la conformité, dans les secteurs tant publics que privés et, en fin de compte, elles servent les intérêts du public en général.

Plusieurs normes ont été établies par l'ISO, parmi elles, nous pouvons citer la famille ISO/IEC 27000 pour les systèmes de gestion de la sécurité de l'information et la sécurité des systèmes d'information. Les normes de cette famille constituent un ensemble de méthodes, mesures et bonnes pratiques reconnues au niveau international dans le domaine de la sécurité de l'information. Elles sont destinées à tout type de société, quelle que soit sa taille, son secteur d'activité ou son pays d'origine. Elles ont pour but de décrire les objectifs à atteindre en matière de sécurité informatique, et non la manière concrète d'y arriver. Celle-ci dépend généralement du contexte propre à toute organisation.

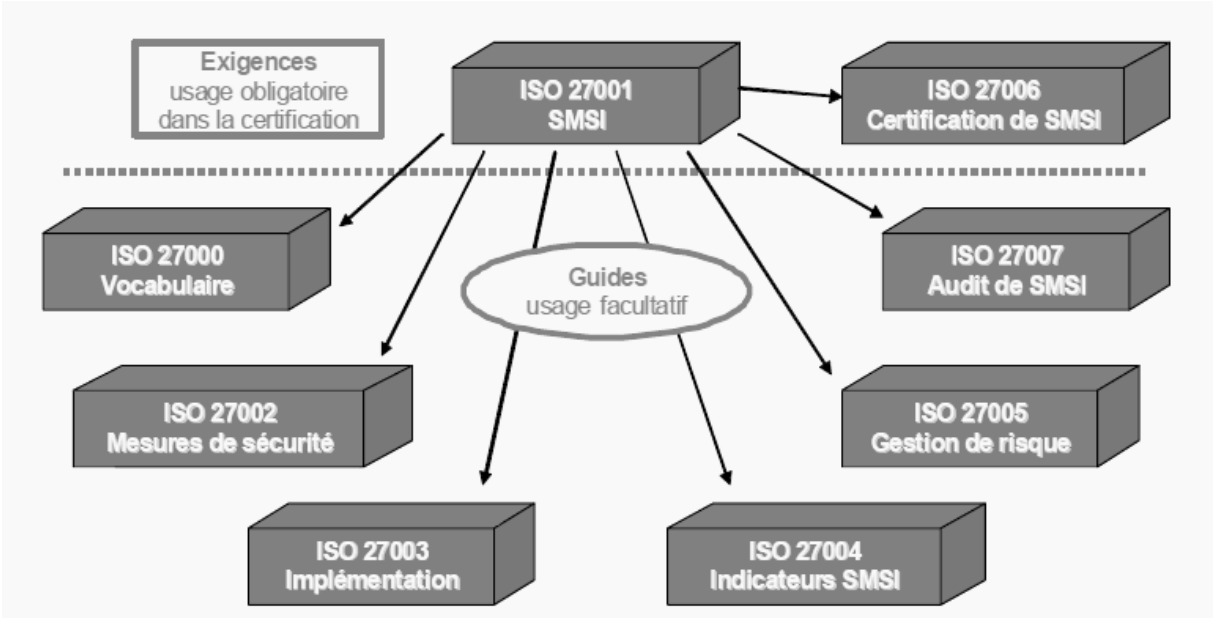


Figure 1.1 : *La famille des normes ISO/IEC 27000* **[07]**

- **ISO/IEC 27000 :** cette norme fournit une vue d'ensemble de la famille de normes du Systèmes de Management de la sécurité de l'information (SMSI). Les objectifs de l'ISO/IEC 27000 sont la fourniture de termes et définitions, et une introduction à la famille des normes SMSI.

- **ISO/IEC 27001 :** c'est une norme internationale de système de gestion de la sécurité de l'information publiée en Octobre 2005 par l'ISO sous le titre « Technologies de l'information et techniques de sécurité » pour la gestion de sécurité de l'information. Son objectif est de protéger les informations de toute perte, vol ou altération, et les systèmes informatiques de toute intrusion. Elle décrit les exigences pour la mise en place d'un SMSI, le SMSI est destiné à choisir les mesures de sécurité afin d'assurer la protection des biens sensibles. **[06]**
- **ISO/IEC 27002 :** la norme ISO/IEC 27002 concerne la sécurité de l'information, publiée en 2005 par l'ISO, dont le titre est « Code de bonnes pratiques pour la gestion de la sécurité de l'information ». c'est un code de pratique, qu'une véritable norme ou qu'une spécification formelle telle que l'ISO/IEC 27001, elle présente une série de contrôles qui suggèrent de tenir compte des risques de sécurité des informations relatives à la confidentialité, l'intégrité et les aspects de disponibilité. Les entreprises qui adoptent l'ISO/IEC 27002 doivent évaluer leurs propres risques de sécurité de l'information et appliquer les contrôles appropriés, en utilisant la norme pour orienter l'entreprise.
- **ISO/IEC 27003 :** diffusée au début de 2009, cette norme déclare fournir un guide de préparation et d'implémentation de la phase de planification d'un SMSI conforme aux exigences de la norme ISO 27001. Elle insiste sur l'approbation du projet par la direction de l'organisation, l'attribution de rôles et de responsabilités dans le cadre du projet et la préparation des points importants de cette planification.
- **ISO/IEC 27004 :** Cette norme couvre les informations mesure de gestion du système de sécurité et les mesures suggérées, y compris ISO 27002 contrôles alignés.
- **ISO/IEC 27005 :** publiée le 4 juin 2008, c'est la première norme de gestion des risques de la sécurité des systèmes d'information, cette norme est un standard international qui décrit le Système de management des risques liés à la sécurité de l'information. Elle explique en détail comment conduire l'appréciation des risques et le traitement des risques, dans le cadre de la sécurité de l'information. Elle propose une méthodologie de gestion des risques en matière d'information dans l'entreprise conforme à la norme ISO/CEI 27001. **[08]**
- **ISO/IEC 27006 :** C'est un standard de sécurité de l'information publié par l'ISO et la Commission électrotechnique Internationale (CEI) en 2010. Son objectif est de fournir les prérequis pour les organismes d'audit et de certification à la norme ISO 27001 pour les SMSI.
- **ISO/IEC 27007 :** publiée en 2010, elle constitue un guide pour l'audit de Systèmes de Management de la Sécurité de l'Information (SMSI).

1.2.6.b. Les méthodes

Une méthode est un moyen d'arriver efficacement à un résultat souhaité précis, une méthode n'intègre pas la notion de document de référence, ni la notion de consensus.

Plusieurs méthodes ont été créés pour la gestion des risques, nous pouvons citer : MARION, MELISA, MEHARI, EBIOS, OCTAVE, etc.

- **MARION (Méthode d'Analyse des Risques Informatique Orientée par Niveaux)**

La méthode MARION a été développée par le CLUSIF (CLub de la Sécurité de l'Information Français) et l'APSAD (Assemblée Plénière des Sociétés d'Assurances Dommage) dans les années 1984, elle permet d'évaluer le niveau de sécurité d'une entreprise au travers d'un questionnaire comportant 27 chapitres répartis en plusieurs grands thèmes qui sont :

- Sécurité organisationnelle et sécurité physique
- Continuité et organisation informatique
- Sécurité logique et exploitation
- Sécurité des applications

La méthode MARION a été délaissée avec le temps, car elle n'évolue plus et a pris du retard par rapport aux nouvelles méthodes qui ont été développées et qui ont réussi à gagner la confiance des utilisateurs. Sa dernière mise à jour a été faite en 1998, elle n'est actuellement plus maintenue.

- **MELISA**

MELISA (Méthode d'évaluation de la vulnérabilité résiduelle des systèmes d'information) fut inventée par Albert Harari au sein de la direction Générale de l'Armement (DGA/DGN) en 1985 en France. MELISA est une méthode assez lourde basée sur un thésaurus de questions. Elle est destinée à être utilisée par de grandes entreprises. MELISA a été abandonnée par ses propriétaires bien qu'elle fut largement utilisée.

- **MEHARI**

MEHARI a été élaborée en 1992 par la Commission Méthodes du CLUSIF (Club de la Sécurité des Systèmes d'Information Français) à partir des méthodes MARION et MELISA. Son objectif est de proposer un plan de sécurité qui se traduit par un ensemble cohérent de mesures permettant de pallier au mieux les failles constatées et d'atteindre le niveau de sécurité répondant aux exigences des objectifs fixés. Méhari propose un guide qui comprend la description de la démarche, les techniques, ainsi que la base de connaissances.

- **EBIOS**

EBIOS (Expression des Besoins et Identification des Objectifs de Sécurité) est une méthode utilisée par l'administration française DCSSI depuis 1995. Elle présente un guide qui comprend la description de la démarche, les techniques, la base de connaissances, et des classes de fonctionnalités et répertoires des fiches. Contrairement à méhari, EBIOS n'utilise pas de questionnaire pour faire l'étude de la vulnérabilité**. [06]**

- **OCTAVE**

OCTAVE est une méthode d'auto-évaluation des risques développée par Carnegie Mellon en 1990, elle présente un processus composé de trois phases principales qui sont :

- Phase 1 : Identification des actifs critiques, des menaces pesant sur ces actifs et des vulnérabilités existantes.
- Phase 2 : Identification des composants technologiques critiques.
- Phase 3 : Evaluation des risques vis-à-vis des actifs critiques, et élaboration du plan de réduction des risques.

Les systèmes d'information sont tous les jours confrontés à différentes menaces qui risquent à tout moment de remettre en question la raison d'être de l'entreprise, d'où la nécessité de mettre en place une politique de sécurité, afin de préserver l'image de cette entreprise, et lui permettre de continuer à exercer son métier en toute sécurité. Dans cette partie nous avons abordé les généralités sur la sécurité des systèmes d'information et les risques qui les guettent, puis nous avons défini la notion de politique de sécurité, enfin, nous avons énumérer les normes et méthodes de gestion des risques les plus connues.

Nous allons consacrer la deuxième partie de ce chapitre, à l'une de ces méthodes, la méthode MEHARI (Méthode Harmonisée d'Analyse des Risques). Nous allons la définir et présenter son cycle de vie, puis étudierons sa dernière version MEHARI 2010. Enfin, nous donnerons un aperçu de sa base de connaissances.

1.3. La méthode de gestion de risques MEHARI

La sécurité du système d'information d'une entreprise est un requis important pour la poursuite de ses activités. Qu'il s'agisse de la dégradation de son image de marque, du vol de ses secrets de fabrication ou de la perte de ses données, une catastrophe informatique a toujours des conséquences fâcheuses pouvant aller jusqu'au dépôt du bilan. **[05]**

Organiser cette sécurité n'est pas chose facile, c'est pourquoi des méthodes ont été développées afin d'aider les responsables informatiques à mettre en place une bonne politique de sécurité et à procéder aux audits permettant d'en vérifier l'efficacité. Parmi ces méthodes nous pouvons citer EBIOS, MEHARI, MARION, MELISA, OCTAVE, etc.

Dans ce qui suit, nous allons nous intéresser à l'une de ces méthodes, qui est la méthode MEHARI. Notre choix s'est fait sur MEHARI pour plusieurs raisons :

- MEHARI est dérivée des méthodes MELISA et MARION,
- MEHARI est utilisée par de nombreuses entreprises publiques et privées, en France et au Québec,

- ❖ MEHARI est en constant développement, actuellement arrivée à la 6eme version,
- ❖ MEHARI est disponible en Français et en Anglais,
- ❖ MEHARI est gratuite.

1.3.2. Présentation de MEHARI

MEHARI est une méthode complète d'évaluation et de management des risques liés à l'information, ses traitements et les ressources mis en œuvre. Développée et maintenue en France par le CLUSIF (Club de La Sécurité des Systèmes d'Information Français) depuis 1995, MEHARI est disponible en Français et en Anglais, c'est la dérivée des méthodes MARION et MELISA (qui elles n'évoluent plus depuis plusieurs années).

MEHARI demeure l'une des méthodes d'analyses des risques les plus utilisées actuellement, elle se présente comme une véritable boite à outils de la sécurité des systèmes d'information, permettant d'appréhender le risque de différentes manières au sein d'une organisation. [05]

- ➢ **Le CLUSIF (Club de la Sécurité de l'Information Français)**

Le CLUSIF est une association française d'entreprises et de collectivités fondée en 1984, dans le but de traiter et d'échanger des informations concernant plusieurs domaine de la sécurité de l'information. Nous pouvons citer : la gestion des risques, les politiques de sécurité, la cybercriminalité, etc. Le résultat des travaux du CLUSIF est disponible sur son site http://www.clusif.asso.fr/. Parmi ces travaux, nous pouvons trouver la méthode de gestion des risques MEHARI.

1.3.3. Objectifs de MEHARI

L'objectif premier de MEHARI est de fournir une méthode d'analyse et de gestion des risques et plus particulièrement, pour le domaine de la sécurité de l'information, avec l'ensemble des outils et moyens requis pour sa mise en œuvre. A cet objectif s'ajoutent deux objectifs complémentaires :

- ➢ Permettre une analyse directe et individualisée de situations de risque décrites par des scénarios de risque,
- ➢ Fournir une gamme complète d'outils adaptée à la gestion à court, moyen et long terme, de la sécurité, quelle que soit la maturité de l'organisme en matière de sécurité et quelques soient les types d'action envisagés.

Compte tenu de ces objectifs, MEHARI propose un ensemble méthodologique cohérent faisant appel à des bases de connaissance adaptées, et capables d'accompagner les responsables d'entreprise ou d'organisme, et les responsables de la sécurité dans leurs différentes démarches et actions, ainsi que les acteurs impliqués dans la gestion des risques. **[12]**

1.3.4. Principes de la méthode MEHARI

Réduire les risques impose de connaitre les enjeux et les processus majeurs de l'organisation afin d'appliquer les mesures organisationnelles et techniques de manière à optimiser les investissements. Cette démarche implique donc d'utiliser les pratiques et solutions à la hauteur des enjeux et des types de menaces pesant sur l'information, sous toutes ses formes, et les processus comme les éléments qui la gèrent et la traitent. **[12]**

MEHARI apporte une aide efficace pour manager et sécuriser l'information de toutes sortes d'organisations, et pour cela, elle fournit un cadre méthodologique, des outils et des bases de connaissances pour :

- Analyser les enjeux majeurs,
- Etudier les vulnérabilités,
- Réduire la gravité des risques,
- Piloter la sécurité de l'information.

MEHARI se compose de plusieurs modules qui peuvent être combinés, en fonction de choix d'orientation ou de politiques d'entreprise, pour bâtir des plans d'action ou pour aider la prise de décision concernant la sécurité de l'information. La figure suivante montre les étapes de la méthode en mettant en évidence ces plans d'action résultants.

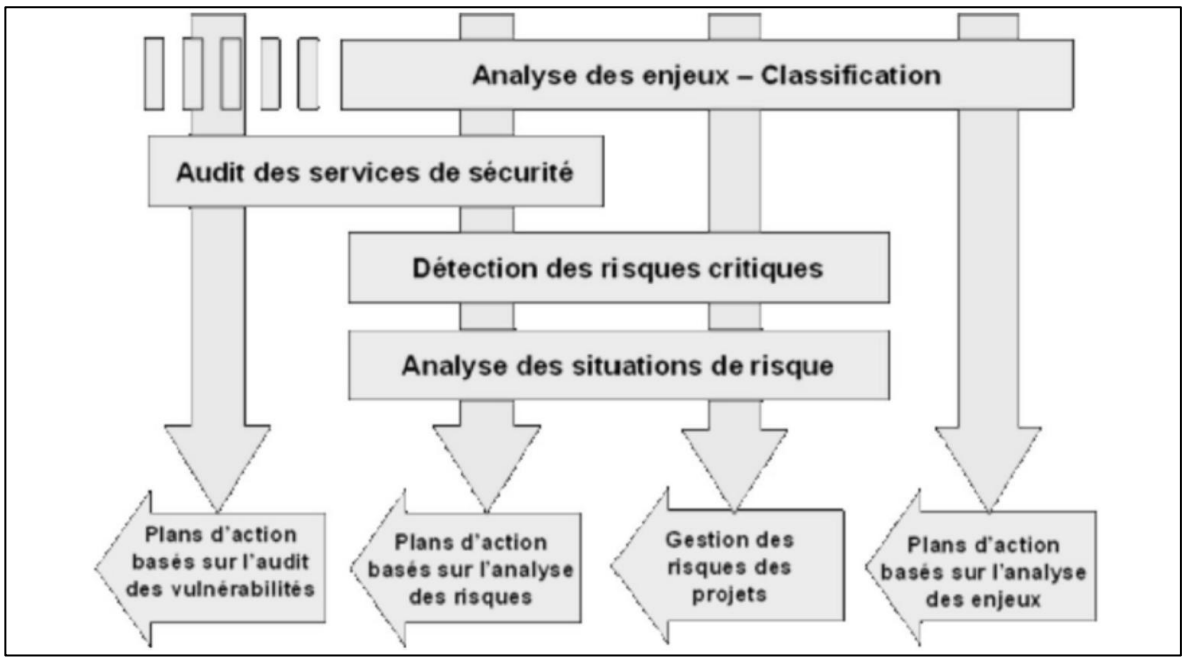


Figure 1.2 : Les étapes de la méthode MEHARI. [06]

La première étape à effectuer est l'analyse des enjeux, elle permet d'obtenir les premiers plans d'action, puis vient la deuxième étape qui est l'audit des services de sécurité, elle donne comme

résultat les plans d'action basés sur l'audit des vulnérabilités, et permet de lancer la troisième étape qui est la détection des risques critiques, une fois terminée, une analyse des situations de risques est effectuée, cette dernière donne fournit résultat deux types de plans d'action : les plans de gestion des risques des projets et les plans d'action basés sur l'analyse des risque. Dans ce qui suit, nous allons étudier en détail la démarche de MEHARI, et nous expliquerons chacune de ses étapes.

1.3.5. Démarche de MEHARI

MEHARI propose une démarche qui se fait en quatre étapes. D'abord, une analyse des enjeux majeurs doit être faite avant d'étudier les vulnérabilités, une fois ces deux choses faites, nous serons en mesure de réduire la gravité des risques et enfin piloter la sécurité de l'information. **[17]**

ETAPE 1 : L'Analyse des enjeux

Cette étape consiste en l'identification des dysfonctionnements potentiels pouvant être causés ou favorisés par un défaut de sécurité, et l'évaluation de la gravité de ces dysfonctionnements. Il s'agit d'une analyse totaliement focalisée sur les objectifs et attentes des metiers de l'entreprise, elle met à contribution les décideurs et le haut management de l'entreprise ou de l'organisme dans lequel elle est menée. Elle vise à etre sélectif dans les moyens à mettre en œuvre pour la sécurité de l'information, à définir les priorités et à éviter des contraites inutiles aux utilisteurs. Il ne s'agit en aucun cas d'un audit des dysfonctionnements réels qui pourrait etre constatés, mais d'une réfléxion sur les risques majeurs auxquels l'entité est exposée et sur le niveau de gravité de leurs conséquences éventuelles. Cette analyse se traduit par :

- Une échelle de valeurs des dysfonctionnements potentiels, document de référence centré sur les impacts « business » ;
- Une classification formelle des informations et ressources du système d'information ;
- Processus d'assistance pour la réalisation de cette classification ;
- Etablissement de liens directs vers l'étude détaillée des risques concernés.

Une fois que les enjeux sont spécifiés, la prochaine étape à réaliser et l'analyse des différentes vulnérabilités qui peuvent exister, et cela en utilisant les résultats fournis par cette première étape.

ETAPE 2 : L'Analyse des vulnérabilités

L'analyse des vulnérabilités revient à identifier les faiblesses et les défauts des mesures de sécurité. Il s'agit d'une évaluation quantitative de la qualité de mesures de sécurité. Et pour cela, des services de sécurité sont proposés par MEHARI, ils sont décrits et documentés dans une base de connaissances développée et maintenue par le CLUSIF, elle est structurée par domaines et par services ayant des finalités précises de réduction de potentialité ou d'impact des situations de risques. Cette analyse des vulnérabilités permet de :

- Corriger les points faibles inacceptables par des plans d'action immédiats.
- Evaluer l'efficacité des mesures mises en place et garantir leur efficience.
- Préparer l'analyse des risques induits par les faiblesses mises en évidence.
- Se comparer à l'état de l'art ou aux normes en usage.

Apres cette étape, il faut procéder à l'analyse des risques pouvant être causés par les vulnérabilités recensées lors de cette étape.

ETAPE 3 : L'Analyse des risques

Cette étape consiste en l'identification des situations susceptibles de remettre en cause un des résultats attendus de l'entreprise ou de l'organisme ou d'une entité en son sein, l'évaluation de la probabilité de telles situations, de leurs conséquences possibles, et de la mise en évidence des mesures susceptibles de ramener chaque risque à un niveau acceptable.

Cette analyse des risques vise à définir les mesures les mieux adaptées au contexte et aux enjeux, à mettre en place un management des risques et garantir que toutes les situations de risques critiques ont été identifiées et prises en compte. Dans cette étape, MEHARI apporte :

- Un modèle de risque et des métriques associées.
- Des automatismes de calcul du niveau de gravité des risques.
- Une démarche structurée et des guides de mise en œuvre.
- Des bases de connaissances.
- Une consolidation des analyses de risque sous forme de plan d'action : module de consolidation des besoins et optimisation des mesures à mettre en œuvre.

ETAPE 4 : Le pilotage de la sécurité

Cette étape demande un cadre structurant pour définir les objectifs annuels ou les étapes de plans d'action et des indicateurs permettant de comparer les résultats obtenus aux objectifs. **[17]** Dans cette étape, MEHARI apporte :

- Un cadre adapté à différentes démarches et différentes sortes de management de la sécurité,
- Une variété d'indicateurs et de synthèses, comme les niveaux de vulnérabilités et de risques, les centres d'intérêt de sécurité, ainsi qu'un tableau de bord des risques critiques.

Dans cette partie, nous avons expliqué la démarche de base de la méthode MEHARI, nous tenons à préciser que ce processus change d'une version à une autre, mais le principe est toujours le même. Dans ce qui suit, nous allons parler de l'évolution de MEHARI depuis sa création à nos jours.

1.3.6. L'évolution de MEHARI

Développée sur la base des travaux de Jean-Philippe Jouas et Albert Harari, elle hérite des connaissances acquises après plus de quinze ans d'analyse des résultats de la méthode MARION et MELISA, de l'expertise de leurs concepteurs et utilisateurs**. [08]** Elle a connue plusieurs versions :

- De 1996 à 2003 : apparition des trois premières versions, V1, V2 et V2.5.
- En 2004 : apparition de la version V3 en Français.
- En Février 2006 : toujours la version V3 en Anglais.
- Fin 2006 à 2007 : apparition de la V4 sous le nom MEHARI 2007.
- Mars 2010 : sortie de la version MEHARI 2010.

Dans ce qui suit, nous allons nous intéresser à la dernière version : MEHARI 2010, et pour cela, nous allons la définir et présenter sa base de connaissances.

1.3.7. Présentation de MEHARI 2010

En Mars 2010, le CLUSIF a mis en ligne la version MEHARI 2010, cette version apporte aux responsables :

- La capacité directe de déterminer les plans de sécurité adaptés pour réduire les risques.
- Un outillage répondant aux exigences de la norme ISO/IEC 27005:2008.
- Un moyen de contrôler la mise en place du management de la sécurité tel que défini par ISO/IEC 27001 et d'évaluer la qualité des mesures de sécurité au regard des pratiques préconisées par ISO/IEC 27002.
- Les scénarios de risque intègrent tous les paramètres permettant de les envisager : actif, menace (acteur, moyens et circonstances), vulnérabilité. **[15]**

1.3.7.a. Processus de la méthode MEHARI 2010

Cette version présente un nouveau processus composé de trois phases comme le montre le schéma suivant :

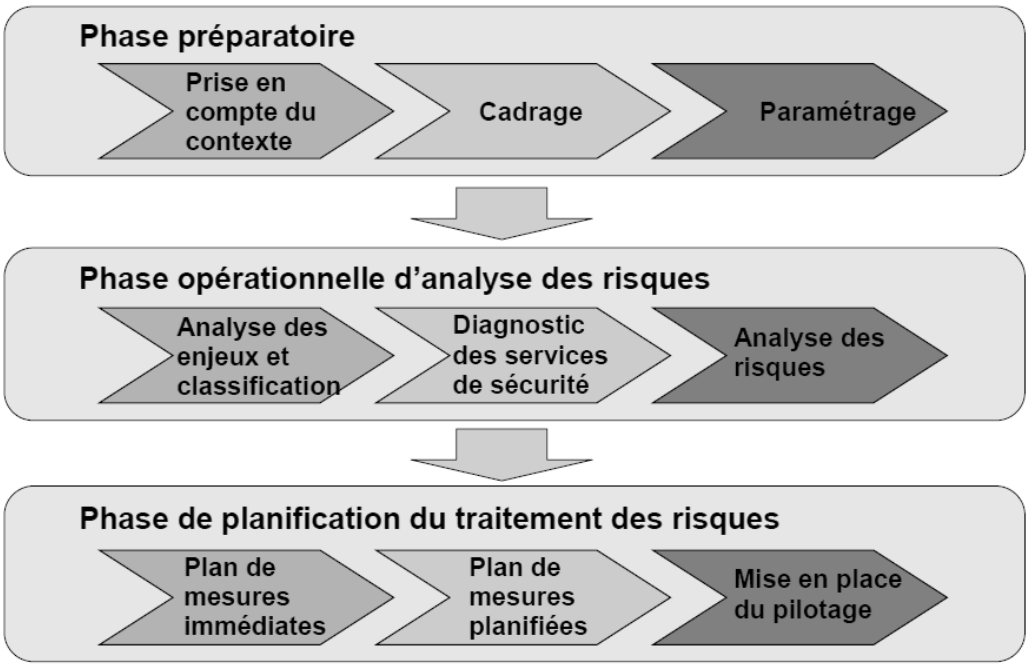


Figure 1.3 : *Processus de la méthode MEHARI 2010***. [08]**

La phase préparatoire : comprend elle-même trois étapes principales qui sont :

1- La prise en compte du contexte : contexte stratégique, contexte technique et contexte organisationnel.
2- Le cadrage de la mission d'analyse et de traitement des risques.
3- La fixation des principaux paramètres de l'analyse des risques en utilisant la grille d'acceptabilité des risques, la grille des expositions naturelles et la grille d'appréciation des risques.

Phase opérationnelle d'analyse des risques : comprend trois étapes principales qui sont :

1- L'analyse des enjeux et la classification des actifs en utilisant l'échelle de valeur des dysfonctionnements et le tableau d'impact intrinsèque.
2- Le diagnostic de la qualité des services de sécurité en établissant le schéma d'audit.
3- L'appréciation des risques en faisant une sélection et une estimation des scénarios de risques.

Phase de planification et de traitement des risques : comprend trois étapes principales :

1- La planification des actions immédiates : en sélectionnant les risques à traiter en priorité absolue et en choisissant les mesures à mettre en œuvre immédiatement.
2- La planification et le choix des mesures à prendre.
3- La mise en place du pilotage du traitement des risques en se basant sur les indicateurs et le tableau de bord.

MEHARI 2010 présente une nouvelle base de connaissances, elle a créée pour répondre aux besoins des utilisateurs et a évolué par rapport aux anciennes versions**. [17]** Le présent point lui sera consacré.

1.3.7.b. Mise en service de la base de connaissances de MEHARI 2010

La base de connaissances de MEHARI est disponible en téléchargement libre sur le site du CLUSIF ***http://***www.clusif.asso.fr/ au format Microsoft Excel ou OpenOffice, nous pouvons aussi utiliser un logiciel partenaire RISICARE 2010 qui a été développé spécialement pour cette méthode.**[16]** La base contient 35 feuilles, séparées en six catégories comme suit :

- Présentation,
- Module d'analyse des enjeux et classification des actifs,
- Module du diagnostic des services de sécurité,
- Module d'analyse de risque,
- Traitement des risques,
- Eléments permanents et paramétrage de la méthode.

Nous présenterons le module du diagnostic des services de sécurité ou d'audit qui fait l'objet d'une partie (essentielle) de notre base de connaissance. Il contient 17 feuilles, comme le montre la figure 1.4 :

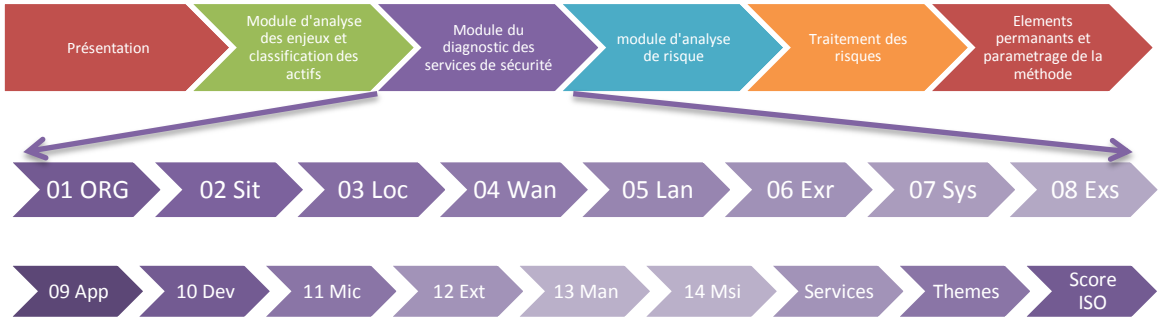


Figure 1.4 : présentation du module du diagnostic des services de sécurité ou d'audit

- **01ORG :** c'est le questionnaire d'audit sur l'organisation de la sécurité, il comprend 167 questions divisées en 5 catégories, 01A Rôles et structures de la sécurité, 01B Référentiel de sécurité, 01C Gestion des ressources humaines, 01D Assurances, 01E Continuité de l'activité. Voici un exemple du questionnaire 01ORG.

- **02 Sit :** Questionnaire d'audit sur la sécurité des sites, composé de 133 questions qui traitent le contrôle d'accès à l'ensemble du site ou à l'immeuble, la protection contre les risques environnementaux ainsi que la protection de l'information écrite.

- **03 Loc :** Questionnaire d'audit sur la sécurité des locaux, il traite plusieurs questions concernant les services techniques, le contrôle d'accès aux locaux sensibles, la sécurité contre les dégâts des eaux, et la sécurité incendie.

- **04 Wan :** Questionnaire d'audit sur les réseaux étendus intersites, il traite la sécurité de l'architecture du réseau étendu et la continuité du service, le contrôle des connexions sur le réseau étendu ainsi que la sécurité des donnes lors des échanges et des communications.

- **05 Lan :** Questionnaire d'audit sur les réseaux locaux (LAN), le réseau local est vu ici, comme le réseau reliant les différents serveurs et postes utilisateurs du site. Voici un extrait de ce questionnaire

- **06 Exr :** Questionnaire d'audit sur l'exploitation des réseaux, il aborde plusieurs questions concernant la sécurité des procédures d'exploitation, le paramétrage et contrôle des configurations matérielles et logicielles, le contrôle des droits d'administration, et les procédures d'audit et de contrôle des réseaux.

- **07 Sys :** Questionnaire d'audit sur la sécurité des systèmes et leur architecture, qui traite le contrôle d'accès aux systèmes, le confinement des environnements, la gestion et l'enregistrement des traces, et enfin, la sécurité de l'architecture.

- **08 Exs :** Questionnaire d'audit qui traite la production informatique dans ses différentes catégories qui sont :
  - Sécurité des procédures d'exploitation,
  - Contrôle des configurations matérielles et logicielles,
  - Gestion des supports informatiques de données et programmes,
  - Continuité de fonctionnement,
  - Gestion et traitement des incidents,
  - Contrôle des droits d'accès d'administration,
  - Procédures d'audit et de contrôle des systèmes de traitement de l'information,
  - Gestion des archives informatique.

- **09App :** Questionnaire traitant la sécurité applicative, selon plusieurs critères : intégrité, confidentialité et disponibilité des données.

- **10 Dev :** Questionnaire d'audit qui concerne la sécurité des projets et développements applications.

- **11 Mic :** Questionnaire d'audit traitant la protection des postes de travail utilisateurs et cela en mettant l'accent sur la protection de ces postes et leurs données, la continuité de service de l'environnement de travail et le contrôle des droits d'administration.

- **12 Ext :** Questionnaire d'audit de l'exploitation des télécommunications, il contient des questions qui traitent la sécurité des procédures d'exploitation, le contrôle des configurations matérielles et logicielles, la continuité du fonctionnement ainsi que le contrôle des droits d'administration.

- **13 Man :** Questionnaire d'audit sur le processus de gestion et cela en abordant différents points qui portent sur la protection des renseignements personnels, la communication financière, le respect de la législation concernant la vérification de la comptabilité informatisée, la protection de la propriété intellectuelle et la protection des systèmes informatisés.

- **14 Msi :** Questionnaire d'audit sur la gestion de la sécurité d'information, il vérifie que l'organisation réalise effectivement une démarche de management et de réduction continue des risques ou de SMSI. Il est recommandé de l'intégrer dans le processus complet de traitement de risque MEHARI et nécessaire de l'associer avec le domaine 01 (Organisation). Il aborde plusieurs questions divisées en plusieurs catégories qui sont :
  - Planification du système de management,
  - Déploiement du système de management,
  - Mise sous contrôle du système de management,
  - Amélioration du système de management,
  - Documentation,

- **Services:** c'est un récapitulé de la qualité de services de sécurité avec variante, il énumère tous les services et sous-services de chaque domaine.

- **Thèmes :** Thèmes de sécurité MEHARI, regroupement des services et sous-servies en 10 centres d'intérêts et 18 axes de représentation.

- **Score ISO :** Table de Scoring ISO 27002 suite au diagnostic des services MEHARI.

Un extrait du questionnaire d'audit sur la Production Informatique **(08 Exs)** est présenté Dans la figure 1.5.

Questionnaire d'audit : Production Informatique	
Référence	**Question**
08D07-01	A-t-on défini les actions à mener par le personnel informatique, pour prévenir, détecter et corriger les attaques par des codes malveillants (virus, spyware, autres) ?
08D07-02	Les serveurs de production (incluant la bureautique et la messagerie) sont ils pourvus de dispositifs de protection contre les virus et contre les codes malveillants ?
08D07-03	Utilise-t-on, sur les serveurs de production, des antivirus provenant de plusieurs fournisseurs ?
08D07-04	Les produits antivirus sont-ils régulièrement (quotidiennement) et automatiquement mis à jour ?
08D07-05	Est-on abonné à une centrale d'alerte permettant d'être prévenu et d'anticiper certaines attaques massives pour lesquelles les antivirus ne sont pas encore à jour ?
08D07-06	Existe-t-il une cellule de crise pouvant être mise en place très rapidement en cas d'alerte ou de détection d'infection ?
08D07-07	L'activation et la mise à jour des antivirus sur les serveurs font-elles l'objet d'un audit régulier ?
08D08	***Gestion des systèmes critiques (vis-à-vis de la permanence de la maintenance)***
08D08-01	A-t-on analysé les conséquences de la disparition d'un fournisseur (en cas de panne, de bogue ou de nécessité d'évolution) pour en déduire une liste de systèmes critiques ? *Ceci vaut aussi bien pour les fournisseurs de matériel que de logiciels ou de services.*
08D08-02	Existe-t-il, pour l'ensemble des systèmes critiques, une solution palliative étudiée pour faire face à la disparition ou la défaillance du fournisseur (dépôt de la documentation de maintenance ou du code source chez un tiers de confiance, remplacement du système par des systèmes standards, etc.) ?
08D08-03	A-t-on l'assurance que cette solution palliative pourra être rendue opérationnelle dans des délais compatibles avec la poursuite de l'activité et acceptés par les utilisateurs ?
08D08-04	A-t-on prévu des variantes de la solution de base au cas où celle-ci rencontrerait des difficultés imprévues ?
08D08-05	Procède-t-on régulièrement à une revue des systèmes pouvant être critiques et des solutions palliatives prévues ?
08D09	***Sauvegardes de secours (recours) externalisées***
08D09-01	L'ensemble des sauvegardes de logiciels et fichiers de configuration permettant de reconstituer l'environnement de production est-il également sauvegardé en dehors du site de production (sauvegardes de secours) ?
08D09-02	L'ensemble des sauvegardes de données est-il également stocké en dehors du site de production (sauvegardes de secours) ?
08D09-03	Procède-t-on régulièrement à des tests de relecture des sauvegardes de secours ?
08D10	***Maintien des comptes d'accès***
08D10-01	A-t-on défini, avec les utilisateurs, les délais maximum admissibles pour le rétablissement de leurs droits, en cas de blocage, accidentel ou non, de leurs de comptes (applicatifs, systèmes ou réseaux) ?
08D10-02	A-t-on défini la procédure a suivre en cas de blocage de compte ?.
08D10-03	Cette procédure est-elle connue de tous les utilisateurs ?
08D10-04	Cette procédure permet-elle de respecter les délais maximum admissibles en cas de bocages isolés de quelques comptes (attaque en déni de service) ?
08D10-05	A-t-on pris en compte la possibilité de blocage simultané de nombreux comptes ?
08D10-06	A-t-on défini avec les utilisateurs le processus à mettre en œuvre en cas de bocage simultané de nombreux comptes ?
08D10-07	Les mesures relatives aux maintien des comptes utilisateurs font-elles l'objet d'un audit régulier ?
08E	**Gestion et traitement des incidents**
08E01	***Détection et traitement (en temps réel) des anomalies et incidents d'exploitation***
08E01-01	A-t-on analysé les événements ou successions d'événements pouvant être révélateurs de comportements anormaux ou d'actions illicites et a-t-on mis en place des points ou indicateurs de surveillance en conséquence ?

Figure1.5 : Extrait du questionnaire d'audit : Production Informatique (08 Exs)

1.4. Conclusion

La sécurité informatique pour une entreprise est un élément déterminant surtout lorsqu'il s'agit de sécuriser l'information ou de l'élément qui la traire, la gère ou la transmet.

Ce chapitre nous a permis d'aborder les aspects généraux de la sécurité des systèmes d'information. Nous avons aussi étudié l'une des méthodes les plus utilisées pour gérer les risques au sein d'une entreprise, la méthode MEHARI.

Dans le prochain chapitre, nous définirons ce qu'est une ontologie et le rôle qu'elle joue pour représenter les connaissances d'un domaine. Nous aborderons ensuite, ses notions de base, ainsi que ses différents aspects.

Chapitre II

Les Ontologies et la Représentation des Connaissances

2.1. Introduction

Nées des besoins de représentation des connaissances, les ontologies sont à l'heure actuelle au cœur des travaux menés en ingénierie des connaissances. Utilisées depuis le début des années 1990, les ontologies sont définies comme une approche de modélisation et de représentation des connaissances. Elles se sont introduites dans le cadre des démarches d'acquisition des connaissances pour les systèmes à base de connaissances et ont évolué vers la représentation des connaissances et leur organisation. Les ontologies ont pour but de saisir la connaissance dans un domaine d'une façon générale, et de fournir une représentation communément acceptée qui pourra être réutilisée et partagée par divers applications et groupes.

Dans ce chapitre, nous allons définir la notion d'ontologie depuis ses origines dans la philosophie à la représentation des connaissances dans l'ingénierie des connaissances. Nous présenterons les différents les types d'ontologies, ses composants ainsi que le cycle de vie correspondant. Nous citerons ensuite quelques méthodologies de construction des ontologies, puis nous présenterons les différents outils de développement.

2.2. La notion d'ontologie

2.2.1. Origines

Le terme « ontologie » a été utilisé pour la première fois par les philosophes grecs au XIX^{ème} siècle. Tiré de « ontos » pour « ce qui existe ou l'existant », et de « logos » pour « discours ou étude », l'ontologie est une branche de la métaphysique qui s'intéresse à la notion d'existence et désigne l'étude des propriétés générales de ce qui existe.

Le terme « ontologie » a été repris en Informatique dans les années 1980 et abordé pour la première fois par John McCarthy dans le domaine de l'intelligence artificielle. Il affirmait qu'il était nécessaire d'abord d'énumérer tout ce qui existe avant de concevoir des systèmes intelligents fondés sur la logique. C'est alors à l'occasion de l'émergence de l'ingénierie des connaissances, que les ontologies sont apparues en Intelligence artificielle comme une réponse aux problématiques de représentation et de manipulation au sein des systèmes informatiques.

2.2.2. Définitions

Une ontologie peut être vue comme réseau sémantique qui regroupe un ensemble de concepts décrivant complétement un domaine de connaissance. Ces concepts sont liés les uns aux autres par des relations taxonomiques et sémantiques. Certains auteurs ont défini l'ontologie comme suit :

« Une ontologie définit les termes et les relations de base du vocabulaire d'un domaine ainsi que les règles qui indiquent comment combiner les termes et les relations de façon à pouvoir étendre le vocabulaire » **[34]**

« une ontologie est une spécification explicite d'une conceptualisation »**[28]**

Cette définition a été modifiée par Borst comme suit :

« une ontologie est une spécification explicite et formelle d'une conceptualisation partagée »**[23]**

Cette définition met l'accent sur quatre termes : explicite, formelle, conceptualisation et partagée.

Explicite : signifie que le type des concepts et les contraintes sur leurs utilisations sont explicitement définis.

Formelle : se réfère au fait que la spécification doit être lisible par une machine.

Conceptualisation : se réfère à un modèle abstrait d'un certain phénomène du monde reposant sur l'identification des concepts pertinents de ce phénomène.

Partagée : l'ontologie capture la connaissance consensuelle qui n'est pas propre à un individu mais validée par un groupe.

2.3. Les composants d'une ontologie

Une ontologie peut être vue comme un treillis de concepts et de relations entre ces concepts qui représentent un domaine de connaissances, sous une forme compréhensible aussi bien par les hommes que par les machines. Les connaissances intégrées dans les ontologies sont formalisées en mettant en évidence cinq principaux éléments qui sont: les concepts, les relations, les fonctions, les axiomes et les instances. Dans ce qui suit nous allons définir chacun de ces éléments.

2.3.1. Les Concepts

Appelés aussi termes ou classes de l'ontologie, les concepts sont des notions (ou objets) qui permettent de représenter un objet matériel, une notion ou une idée **[39].** Selon Gomez-Perez **[27],** les concepts peuvent être classifiés selon plusieurs dimensions :

- **Niveau d'abstraction** : concret ou abstrait,
- **Atomicité :** élémentaire ou composé,
- **Niveau de réalité :** réel ou fictif.

Les concepts sont organisés en taxonomie, une taxonomie est une hiérarchie de concepts (ou d'objets) reliés entre eux en fonction de critères sémantiques particuliers. Un concept est composé de trois parties : un terme (ou plusieurs), une notion et un ensemble d'objets.

- **Le terme :** permet de désigner le concept, ce terme est appelé aussi 'label de concept'.
- **La notion (Intention):** également appelée intention du concept, elle contient la sémantique du concept, exprimée en termes de propriétés, d'attributs, de règles et de contraintes.
- **L'ensemble d'objets (Extension) :** également appelé extension du concept, regroupe les objets manipulés à travers le concept, ces objets sont appelés instances du concept.

Bachimont **[21]** utilise le terme de « concept formel » pour désigner l'extension d'un concept et de « concept sémantique » pour désigner l'intention d'un concept. Un concept est ainsi doté d'une sémantique référentielle (celle imposée par son extension) et d'une sémantique différentielle (celle imposée par son intention). Il est à noter qu'un concept peut très bien avoir une extension vide, il s'agit alors d'un concept générique et correspond à une notion abstraite. Deux concepts peuvent partager la même extension sans pour autant avoir la même intention. De plus, des concepts partageant la même extension mais pas leur intention peuvent être désignés par le même terme, ceci correspond à des points de vue différents sur un même objet.

Bien que le langage naturel contienne des termes désignant plusieurs concepts sémantiquement différents, de telles ambiguïtés ne sont pas gérables en machine, où un domaine de connaissances permet généralement d'éviter les homonymes de concepts. Il apparait par contre souhaitable de gérer les synonymes et de permettre la désignation d'un concept par plusieurs termes, pour assurer une plus grande souplesse d'utilisation de l'ontologie.

Les caractéristiques d'un concept :

Pour caractériser un concept dans un domaine particulier, un ensemble de propriétés lui est associé, les propriétés portant sur un concept sont les suivantes **[21]:**

- **La généricité :** un concept est générique s'il n'admet pas d'extension, exemple : la vérité est un concept générique.
- **L'identité** : un concept porte une propriété d'identité si cette propriété permet d'identifier de manière unique les différentes instances de ce concept. Cette propriété peut porter sur des attributs du concept ou sur d'autres concepts, exemple : le concept « étudiant » porte une propriété d'identité liée au numéro de l'étudiant, deux étudiants ne peuvent pas avoir le même numéro.

- **La rigidité :** un concept est rigide si toute instance de ce concept reste toujours la même en fonction du temps, exemple : « humain » est un concept rigide, « étudiant » est un concept non rigide.
- **L'anti-rigidité :** un concept est anti-rigide si toute instance de ce concept est essentiellement définie par son appartenance à l'extension d'un autre concept, exemple : « étudiant » est un concept anti-rigide car l'étudiant est avant tout un humain.
- **L'unité :** un concept est dit concept unité, si pour chacune de ses instances, les différentes parties de l'instance sont liées par une relation qui ne lie pas d'autres instances de concepts.

Les propriétés portant sur deux concepts sont: [38]

- **L'équivalence :** deux concepts sont équivalents s'ils ont la même extension.
- **La disjonction :** deux concepts sont disjoints si leurs extensions sont disjointes, exemple : homme et femme.
- **La dépendance :** un concept C1 est dépendant d'un concept C2 si pour toute instance de C1 il existe une instance de C2 qui ne soit ni partie ni constituant de l'instance de C2, exemple : parent est un concept dépendant de enfant (et vice-versa).

2.3.2. Les Relations

Les relations représentent un type d'interaction ou bien des associations existant entre les concepts d'un domaine. Une relation R qui relie plusieurs concepts C_i avec i allant de 1 à n, se définie à partir d'un produit de n concepts C_1, C_2, C_3, … C_n, comme suit :

$$\mathbf{R : C_1 \times C_2 \times C_3 \times \ldots \times C_n}$$

Une relation permet de lier des concepts ou des instances de concepts. Elle est caractérisée par un terme (ou plusieurs termes) et une signature qui précise le nombre d'instances de concepts que la relation lie, son type et l'ordre des concepts, c'est-à-dire, la façon dont la relation doit être lue. Par exemple, la relation « écrit » lie une instance du concept « personne » et une instance du concept « texte », dans cet ordre. Nous pouvons citer quelques exemples de relation binaires :

- sous-classe-de / is-a : spécialisation, généralisation,
- partie-de : agrégation ou composition,
- Associée-à, instance-de.

La relation taxonomique (ou Subsomption) : La relation taxonomique ou de subsomption est une relation binaire qui permet d'organiser hiérarchiquement un ensemble de concepts, c'est la relation is-a, de spécialisation ou de généralisation.

Exemples :

Le concept POMME ROUGE **est subsumé** par le concept POMME.

Le concept ENFANT **subsume** les concepts FILLE et FILS.

2.3.3. Les fonctions

Les fonctions constituent des cas particuliers de relation où le $N^{ième}$ élément de la relation est défini en fonction des n-1 premiers éléments. Formellement, les fonctions sont définies ainsi :

$$\mathbf{F : C_1 \times C_2 \times ..... \times C_{n-1} \rightarrow C_n}$$

Exemple : le prix d'une voiture d'occasion est calculé en fonction de son modèle, de sa date de construction et de son kilométrage.

2.3.4. Les axiomes

Les axiomes ou règles d'inférence permettent de définir la sémantique des concepts et relations, leurs propriétés et toutes contraintes quant à leur interprétation. Ils sont définis à l'aide de formules bien formées de la logique du premier ordre en utilisant les prédicats de l'ontologie. Les axiomes sont des expressions qui sont toujours vraies, leur inclusion dans une ontologie peut avoir plusieurs objectifs comme :

- 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.

2.3.5. Les instances

Constituent la définition extensionnelle de l'ontologie, les instances sont les valeurs concrètes et occurrences des concepts et des relations, elles sont utilisées pour représenter des éléments dans un domaine. Par exemple, « Ali » et « Mustapha » sont des instances du concept « Etudiant ».

2.4. Les ontologies et la représentation des connaissances

La représentation des connaissances est une branche de l'Ingénierie des Connaissances qui désigne un ensemble d'outils et de procédés, destinés à représenter et à organiser le savoir humain, pour l'utiliser et le partager. Afin de permettre la représentation de ces connaissances, plusieurs formalismes ont été proposés. Dans ce qui suit, nous allons définir ces formalismes de représentation.

2.4.1. Formalismes de représentation des connaissances

Représenter des connaissances propres à un domaine particulier consiste à décrire et à coder les entités de ce domaine de manière à ce qu'une machine puisse les manipuler afin d'effectuerdes raisonnements. Comme alternative à la logique classique, l'Intelligence Artificielle a proposé divers formalismes de représentation. Ceux qui ont été le plus utilisé pour représenter les ontologies sont :

- Les frames
- Les réseaux sémantiques
- Les logiques de description

2.4.1.a. Les Frames (Schémas)

C'est un formalisme qui a été introduit par M. Minsky en 1975 **[32]** . Le principe de ce modèle est de décomposer les connaissances en classes (Frames) qui représentent les concepts du domaine. Un Frame est une structure de données permettant de représenter une situation ou un objet stéréotypé. Un Frame est pourvu d'un certain nombre d'attributs ou slots qui sont mis en correspondance avec des données spécifiques à une situation, chaque attribut peut être caractérisé pas un ensemble d'informations (facettes).

2.4.1.b. Les réseaux sémantiques

Un réseau sémantique est un modèle de représentation du contenu sémantique des concepts sous forme de graphe. Un graphe est formé de nœuds, représentant les concepts, reliés par des arcs décrivant les relations entre eux.

En Intelligence Artificielle, M. Quillian fut le premier à développer de tels réseaux en tant que modèles de la mémoire associative humaine (Quillian, 1968). La théorie des graphes conceptuels représentant les relations sémantiques, constitue le formalisme le plus répandu pour conceptualiser les ontologies.

2.4.1.c. Les logiques de description

Les logiques de description (LDs) sont un formalisme dont le développement fut fortement influencé par les travaux sur la logique des prédicats, les Frames et les réseaux sémantiques. Les Logiques de Description permettent de représenter les connaissances sous forme de concepts, de rôles et d'individus **[30]**. Les rôles sont des relations binaires entre les concepts. Les propriétés des concepts, rôles et individus sont exprimées en logique des prédicats, en particulier les propriétés de subsumption. La modélisation des connaissances d'un domaine avec les Logiques de description se réalise en deux niveaux :

- ***Niveau Terminologique ou TBox :*** décrit les connaissances générales d'un domaine, il comprend la définition des concepts et des rôles.
- ***Niveau Assertionnel ou ABox :*** représente une instanciation spécifique. Il décrit les individus en les nommant et en spécifiant en termes de concepts et de rôles des assertions qui portent sur ces individus nommés.

2.5. Les domaines d'utilisation des ontologies

2.5.1. Les ontologies et les Systèmes à Base de Connaissances SBC

Les ontologies sont apparues en informatique, plus précisément en Ingénierie des connaissances, dans le cadre des démarches d'acquisition des connaissances pour les systèmes à base de connaissances (SBC). Les SBC proposaient alors de spécifier, d'un côté des connaissances du domaine modélisé, de l'autre, des connaissances de raisonnement qui manipulent et utilisent ces connaissances du domaine. L'idée de cette séparation modulaire était de construire plus rapidement des SBC en réutilisant le plus possible des composants génériques, que ce soit au niveau du raisonnement ou des connaissances du domaine **[21].** L'objectif des ontologies est de diversifier les applications des SBC, et de permettre une représentation des connaissances indépendantes de ses diverses applications, de manière à assurer leur portabilité d'une application à une autre.

2.5.2. Les ontologies et le Web Sémantique

L'expression Web sémantique a été introduite par Tim Berners-Lee au sein du W3C. **[23]** Elle fait référence à la vision du Web de demain, qui est vu comme un vaste espace d'échange de ressources entre êtres humain et machine permettant une exploitation, qualitativement supérieure, de grands volumes d'informations et de services variés.

Le Web actuel est essentiellement syntaxique, dans le sens que la structure des documents (ou de ressources au sens large) est bien définie, mais que son contenu reste quasi-inaccessible aux traitements machines. Seuls les humains peuvent interpréter leurs contenus. La nouvelle génération du Web – le Web Sémantique – a pour ambition de lever cette difficulté, en associant aux ressources du Web des entités ontologiques comme références sémantiques, ce qui permettra aux différents agents logiciels d'accéder et d'exploiter directement le contenu des ressources et de raisonner dessus. Ce référencement sémantique peut aussi résoudre les problèmes d'interprétation des ressources informationnelles provenant des applications hétérogènes et réparties et de permettre ainsi à ces applications d'être intégrées sémantiquement. **[40]** Le Tableau 1 présente une comparaison entre le Web actuel et le Web Sémantiques :

Le Web Actuel	Le Web Sémantique
- Essentiellement syntaxique - Ensemble de documents - Basé essentiellement sur HTML - Recherche par mots clé - Utilisation par l'humain - URL (Uniform Resource Locator)	- Orienté Sémantique - Ensemble de connaissances - Basé sur XML et RDF(S) - Recherche par concepts - Utilisation par la machine et l'humain. - URI (Uniform Resource Identifier)

Tableau 2.1 : Comparaison entre le Web Actuel et le Web Sémantique.[35]

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. **[35]**

Langages du Web sémantique

L'approche retenue par la W3C, du point de vue des langages a été de reprendre ce qui a largement contribué au succès du Web. Des langages standardisés dont les fichiers de textes balisés sont facilement échangeables et utilisables par de multiples outils. Nous présenterons dans la figure 1les versions récentes des différentes couches du Web Sémantique :

- **Couche URI :** reprend le mode d'adressage des ressources,
- **Couche XML :** la description de documents sous forme d'arbres XML
- **Couche RDF/RDFS :** concerne le modèle de données basique RDF, et RDF Schéma pour les ontologies.
- **Couche Ontologie :** c'est le standard courant pour le Web, OWL.
- **Couche Logique :** c'est l'évolution des langages pour les ontologies, et les applications spécifiques pour les connaissances déclaratives.
- **Couche Contrôle :** concerne la génération de contrôles, et la validation.
- **Couche Sécurisation :** signature numérique, recommandations, etc.

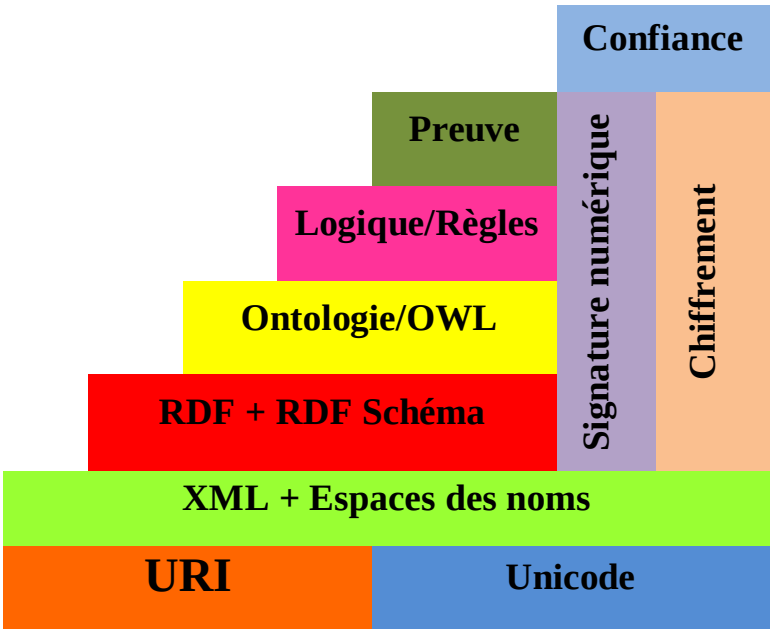


Figure 2.1 : Langages du Web sémantique [52]

2.6. Classement des ontologies

Les ontologies présentes dans la littérature peuvent être classifiées suivant différentes dimensions qui sont : le degré de formalisation, le type de la structure de la conceptualisation et la granularité.

2.6.1. Selon le degré de formalisme

L'ontologie se présente de façon différente selon le degré de formalisation du langage utilisé pour définir la signification des termes**. [39]** Nous pouvons citer quatre catégories qui sont :

- **Ontologie hautement informelle :** Une ontologie est dite hautement informelle si elle est exprimée en langage naturel sans aucune restriction.

- **Ontologie semi-informelle :** Une ontologie est dite semi-informelle si elle est exprimée sous une forme limitée, restreinte et structurée du langage naturel.

- **Ontologie semi-formelle :** Une ontologie est dite semi-formelle si elle est représentée à l'aide d'un langage artificiel défini de façon formelle.

- **Ontologie rigoureusement formelle :** Une ontologie est dite rigoureusement formelle si elle est exprimée dans un langage contenant une sémantique formelle, des théorèmes et des preuves de propriétés.

2.6.2. Selon la granularité

La granularité signifie le niveau de détail des objets de la conceptualisation préconisé, nous pouvons citer deux catégories, granularité fine et granularité large.

- **Granularité fine :** Ce sont les ontologies très détaillées, possédant un vocabulaire plus riche, capable d'assurer une description détaillée des concepts pertinents d'un domaine ou d'une tache.
- **Granularité large :** Ce sont les ontologies qui ont un vocabulaire moins détaillé.

2.6.3. Selon les objets modélisés

Ceci est lié au type de la structure de la conceptualisation, en effet, l'ontologie se définit selon le domaine étudié et le degré de généralité ou de précision des connaissances représentées, nous avons quatre types d'ontologies:

2.6.3.a. Les ontologies de haut niveau (supérieures)

Les ontologies de haut niveau portent sur des concepts généraux comme le temps, l'espace, la matière, les évènements…etc, qui se veulent indépendants d'un domaine ou d'une application particulière, ce qui rend l'ontologie réutilisable d'un domaine à un autre.

2.6.3.b. Les ontologies de domaine

Les ontologies de domaines sont plus spécifiques. Elles synthétisent les connaissances spécifiques à un domaine particulier et décrivent le vocabulaire ayant trait à un domaine générique comme l'enseignement, la médecine, …etc.

2.6.3.c. Les ontologies de tâches

Ce type d'ontologies est utilisé pour conceptualiser des taches ou activités spécifiques comme le diagnostic, la planification, la conception, la vente,…etc. L'ontologie de tâche régit un vocabulaire qui décrit une structure de résolution de problèmes inhérente aux tâches et indépendante du domaine.

2.6.3.d. Les ontologies d'application

Ces ontologies sont les plus spécifiques, elles contiennent des concepts dépendants d'un domaine et d'une tache particuliers. Ces concepts correspondent souvent aux rôles joués par les entités du domaine tout en exécutant une certaine activité. Ces ontologies peuvent contenir des extensions spécifiques telles les méthodes et tâches. Elles contiennent aussi toutes les définitions nécessaires pour décrire la connaissance requise pour une application particulière.

2.7. Construction d'une ontologie

2.7.1. Méthodes de base de construction d'une ontologie

Il existe trois méthodes possibles de création d'une ontologie. En effet, une ontologie peut être construite d'une façon manuelle, automatique ou mixte.

- **Manuel :** dans le mode manuel, les experts créent l'ontologie en s'appuyant sur des techniques classiques de collecte et d'analyse des connaissances comme c'est le cas pour les niveaux supérieurs des ontologies Cyc et Wordnet.

- **Automatique :** la création d'une ontologie d'une manière automatique se base sur des méthodes formelles et des techniques d'extraction des connaissances en employant des outils linguistiques et statistiques.

- **Mixte :** dans le mode mixte, les ontologies sont construites par les techniques automatiques tout en intégrant des méthodes permettant d'étendre des ontologies ayant été construites manuellement.

Quel que soit le mode choisi, l'élaboration de toute ontologie doit s'appuyer sur un certain nombre de règles qu'il est nécessaire de respecter et une méthodologie de construction d'ontologies.

2.7.2. Cycle de développement d'une ontologie

Le processus de construction d'une ontologie est une collaboration qui réunit des experts du domaine de connaissances, des ingénieurs de la connaissance, voire les futurs utilisateurs de l'ontologie. Cette collaboration ne peut être fructueuse que si les objectifs du processus ont été clairement définis, ainsi que les besoins qui en découlent. **[27]**

La figure 2.2 montre le cycle de développement d'une ontologie, en effet , la première étape consiste à spécifier les besoins, ce qui nous donne comme résultat un corpus contenant les données brutes de notre contexte. La deuxième étape est la conceptualisation, elle permet le passage du corpus retenu au modèle conceptuel. Du modèle conceptuel nous passons à l'étape de l'ontologisation qui donne comme résultat une ontologie semi-formalisée. L'étape suivante est l'opérationnalisation, elle permet d'obtenir une ontologie opérationnelle, prête à l'emploi. Une dernière étape est nécessaire, qui l'évaluation et l'évolution, afin de s'assurer du bon fonctionnement de l'ontologie obtenue.

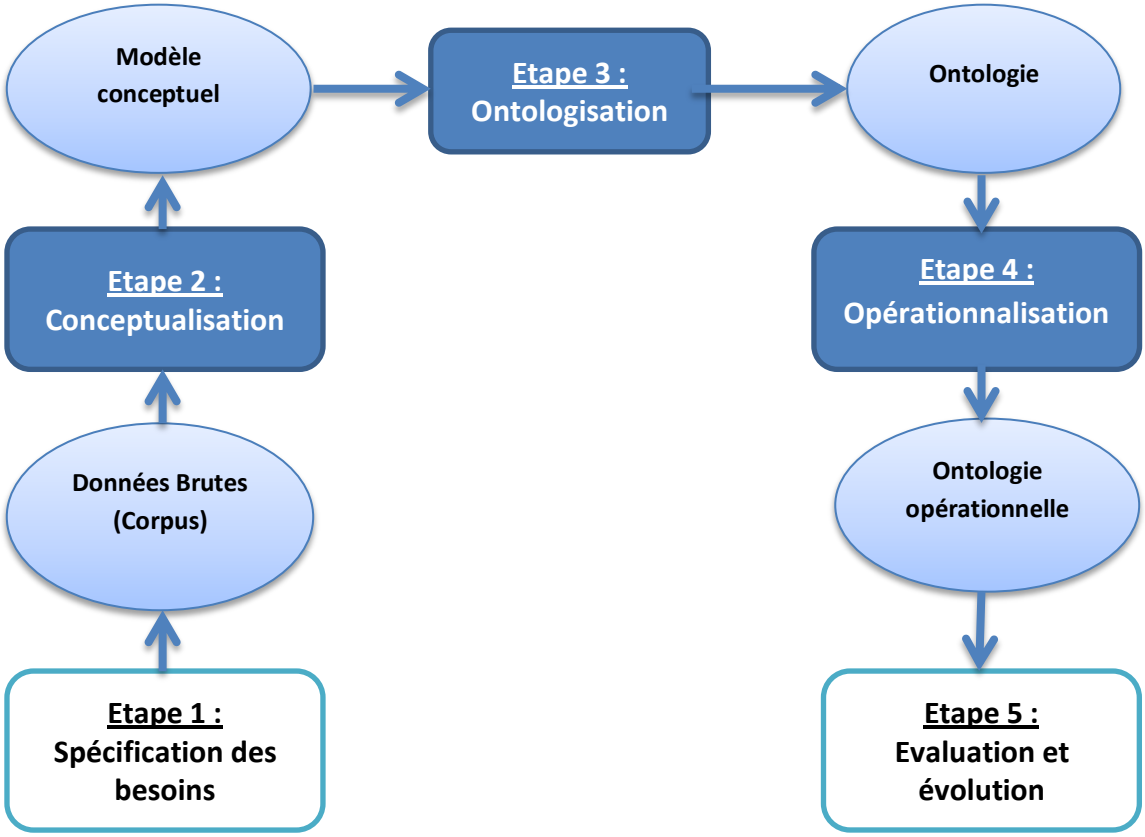


Figure 2.2 : Cycle de développement d'une ontologie [25]

Dans ce qui suit, nous allons expliquer en détail, chacune de ces étapes.

1. Spécification des besoins

Avant le lancement de chaque projet, une étape indispensable doit être faite. Cette étape est la spécification des besoins, elle a pour but de fournir une description claire du problème étudié ainsi que la façon de le résoudre, et cela en exécutants les tâches suivantes :

- Délimiter l'objectif opérationnel de l'ontologie à travers des scénarios d'usage.
- Délimiter le domaine de connaissance aussi précisément que possible.
- Identifier les utilisateurs pour pouvoir choisir en accord avec l'objectif opérationnel, le degré de formalisme de l'ontologie, et sa granularité.

Une fois le but défini, le processus de construction de l'ontologie peut démarrer, en commençant par la phase de conceptualisation.

2. Conceptualisation

La conceptualisation est un processus d'abstraction qui consiste à identifier les concepts essentiels du domaine de connaissances et d'établir les relations entre ces concepts, au sein d'un corpus représentatif du domaine. Il s'agit donc de décrire le domaine de connaissances grâce à des concepts plus ou moins précis et aux relations qui peuvent exister entre ces concepts, ce qui permet d'aboutir à un modèle conceptuel.

L'identification des concepts et relations peut se faire selon l'analyse des textes (documents, notes, comptes rendus d'interviews, etc.) cette analyse est généralement : « une analyse informelle des textes qui peut être doublée par une analyse automatique permettant de détecter les termes et structures sémantiques (définitions, règles) présentes dans le corpus de documents ». **[26]**

Néanmoins, cette analyse n'est pas suffisante pour spécifier la sémantique du domaine, car certaines connaissances ne prennent sens que lorsqu'elles sont lues par un expert ou un spécialiste du domaine. La sémantique doit alors être précisée ou validée par les experts du domaine considéré.

Pour identifier les concepts, Uschold **[39]** suggère trois stratégies qui sont :

- **Une approche descendante,** où il s'agit de partir des concepts les plus généraux et de les spécialiser par la suite,
- **Une approche ascendante,** qui consiste à considérer tous les termes spécifiques et ensuite à trouver les termes génériques associés,
- **Une approche intermédiaire,** dans laquelle les concepts se structurent autour des concepts importants du domaine (ni trop généraux, ni trop spécifiques). Ces concepts centraux sont ensuite reliés avec les concepts proches soit par spécialisation soit par généralisation.

Après l'identification des concepts et termes importants, des définitions doivent leur être attribuées. Les définitions d'ontologie ont un rôle normatif et indiquent comment un ensemble réduit de termes peut être utilisé par rapport à un autre **[39]**. Ainsi, chaque définition exige une bonne compréhension. Notamment dans sa relation avec les autres définitions dans l'ontologie.

Néanmoins, la principale difficulté rencontrée durant le processus de la conceptualisation réside dans l'émergence des différences significatives ou des contradictions au niveau du sens prêté à certains concepts ou relations, ce qui implique la nécessité d'une normalisation sémantique. Cette normalisation doit être le fruit d'un dialogue entre les experts **[21]**. De manière générale, l'échange entre experts est le meilleur moyen de faire émerger une sémantique claire et non ambiguë. **[26]**

3. La Formalisation ou l'Ontologisation

L'ontologisation consiste en une formalisation partielle sans perte d'information du modèle conceptuel obtenu à l'étape de conceptualisation. **[28]** Elle est réalisée par le biais d'un langage formel ou formalisme qui est un ensemble de composants sémantiques(contenu), de règles structurelles (mode d'emploi) et d'une notation formelle particulière (forme) destinée à organiser les relations entre les éléments constituant l'ontologie. L'objectif de l'utilisation d'un langage de formalisation d'ontologies, est de permettre d'une part de réduire les ambigüités du langage naturel en offrant une plus grande expressivité, et d'autre part de rendre l'ontologie compréhensible par les machines.

Différents formalismes tels que les logiques de description, les réseaux sémantiques et les frames (schémas) peuvent être employés pour représenter formellement une ontologie. Les logiques de description représentent la connaissance sous forme de propositions ou affirmations sur le domaine. Les réseaux sémantiques, tout en gardant cette approche propositionnelle, tiennent compte de la structure et des relations entre ces propositions. Les frames représentent le domaine en termes de ses objets et leurs propriétés et relations. Ces paradigmes se différencient les uns des autres par leur formalisme de représentation des connaissances et leurs mécanismes d'inférence permettant de raisonner sur les représentations. Compte tenu de la diversité des formalismes de représentation des connaissances, les concepteurs d'ontologies sont amenés à prendre en considération certains critères avant de choisir le langage le plus adéquat. Ces critères peuvent être :

- La puissance d'expressivité,
- La sémantique et la syntaxe,
- L'existence d'outils associés d'aide au développement (outils d'aide à la construction de l'ontologie).

A l'issue de cette étape, on obtient une ontologie formelle. Cependant, certaines connaissances du domaine peuvent être abandonnées, du fait de l'impossibilité de lever certaines ambiguïtés, ou du fait des limitations de l'expressivité du langage utilisé.

4. Opérationnalisation

Cette étape consiste à formaliser complètement l'ontologie obtenue dans un langage de représentation de connaissances formel et opérationnel, par exemple, le modèle des Graphes Conceptuels ou la Logique de Descriptions. On obtient alors une représentation formelle des connaissances du domaine. Ainsi, le caractère formel de l'ontologie permet à une machine, via cette ontologie, de manipuler des connaissances du domaine. La machine doit donc pouvoir utiliser des mécanismes opérant sur les représentations de l'ontologie. Mais avant d'être livrée aux utilisateurs, l'ontologie doit être testée par rapport au contexte d'usage pour lequel elle a été conçue, ceci est assuré par la dernière étape qui est l'évaluation de l'ontologie.

5. L'évaluation et l'évolution de l'ontologie

Dans cette étape, il est nécessaire d'effectuer deux tâches, qui sont :

- La vérification : l'ontologie doit être conforme à un modèle formel de représentation de connaissances.
- La validation : s'assurer de la conformité et fidélité sémantique de l'ontologie au domaine de connaissances.

2.8.3. Méthodologies de création d'ontologies

L'ingénierie ontologique ne propose à l'heure actuelle, aucune méthode normalisée ou méthodologie générale de construction d'ontologies. Cependant, certains auteurs ont proposé des méthodologies inspirées de leur expérience de construction d'ontologies. Ces méthodologies proposent à travers un ensemble d'étapes, un cycle de développement d'ontologies qui peut être adopté lors de la construction d'une nouvelle ontologie. Dans ce qui suit nous présenterons quelques méthodes de construction d'ontologies.

2.8.4.a. La méthode ENTERPRISE

Proposée par Uschold **[38]**, la méthode Entreprise est basée sur l'expérience du développement de l'ontologie (The Entreprise Ontology), conçue pour la modélisation des processus d'entreprise. Elle passe par quatre étapes qui sont :

Etape 1 : Identification des objectifs et du contexte

Cette étape consiste à identifier les raisons pour lesquelles l'ontologie sera construite, les utilisations prévues, les finalités, et les utilisateurs potentiels de l'ontologie.

Etape 2: Construction de l'ontologie

Cette étape est composée de trois tâches :

- **Capture de l'ontologie** : cette étape consiste à identifier les concepts et les relations clés du domaine d'intérêts, les définir précisément, sans ambiguïté et en langage naturel, et enfin identifier les termes dénotant ces concepts et relations.
- **Codage de l'ontologie** : Il s'agit de représenter explicitement la conceptualisation issue de l'étape précédente à l'aide d'un langage de représentation formel.
- **Intégration des ontologies d'autres domaines :** C'est l'une des tâches les plus importantes et la plus difficile. Elle consiste à trancher sur la possibilité d'intégrer des ontologies d'autres domaines afin de compléter l'ontologie en cours de création.

Etape 3: Evaluation

Il s'agit de la mise en épreuve de l'ontologie en la faisant confronter aux objectifs pour lesquels elle a été conçue, s'assurer de la validité de la taxonomie en vérifiant qu'il n y a pas de cycle,

que toutes les instances d'une classe sont aussi les instances de la classe mère et qu'il n y a pas de classe isolée.

Etape 4 : Documentation

Selon Uschold **[38],** il est désirable d'établir des directives pour documenter l'ontologie, qui différent probablement selon le type et le but de cette dernière.

2.8.4.b. La méthode développée par l'université de Stanford

Cette méthode comporte sept étapes qui sont les suivantes **[31]** :

Etape 1 : déterminer le domaine et la portée de l'ontologie

Cette étape se fait en répondant aux questions ci-dessous tout au long de la conception de l'ontologie et qui aident à définir la portée du domaine de l'ontologie :

- Quel est le domaine que va couvrir l'ontologie ?
- Dans quel but utiliserons-nous l'ontologie ?
- A quels types de questions l'ontologie devra-t-elle fournir des réponses ?
- Qui va utiliser et maintenir l'ontologie ?

Etape 2: envisager une éventuelle réutilisation des ontologies existantes

Dans tout domaine de recherche, il est utile de profiter de ce que les autres ont fait afin d'en tirer des informations et ainsi permettre d'élargir le travail et l'affiner pour répondre à nos propres besoins. Il peut être intéressant d'importer des ontologies déjà existantes (dans le même domaine),les raffiner et les perfectionner pour aboutir à une ontologie plus complète et étendue.

Etape 3: énumérer les termes importants dans l'ontologie

Il est important d'établir en premier lieu une liste complète des mots et termes concernant le domaine d'intérêt, et cela sans se soucier de la catégorisation de ces derniers dans des classes, hiérarchie, chevauchement, etc. Les questions à se poser pour établir cette liste sont les suivantes :

- Sur quels termes souhaiterons-nous discuter ?
- Quelles sont les propriétés de ces termes ?
- Que veut-on par ces termes ?

Etape 4 : définir les classes et la hiérarchie des classes

A partir de la liste de l'étape précédente, on commence par définir les classes en sélectionnant les termes qui décrivent des objets ayant une existence indépendante. Ce sont ces termes qui constitueront les classes ou concepts de l'ontologie. Il faut ensuite organiser ces classes dans une taxonomie hiérarchique en suivant la règle : 'Si une classe A est super-classe d'une classe B, alors toute instance de B est également, une instance de A'.

Etape 5 : définir les propriétés des classes (les attributs ou rôles)

Dans cette étape, on devra décrire la structure interne des concepts tirés pendant l'étape précédente. Les propriétés définissent la structure interne et les caractéristiques des classes.
La plupart des termes restants (qui ne sont pas des classes) ont de fortes chances de représenter des propriétés de ces classes. Chaque propriété sera ensuite rattachée comme attribut à la classe qu'elle décrit. Il faut ensuite prendre en considération les classes et sous classes, ainsi un attribut doit être rattaché à la classe la plus générale pouvant avoir cette propriété et toutes les sous-classes de cette classe héritent cet attribut.

Etape 6 : définir les facettes des attributs

Les attributs peuvent avoir plusieurs facettes (appelées parfois restrictions de rôles). Les facettes les plus communes décrivent :

- Le type de valeur des attributs,
- Le nombre de valeurs ou cardinalité,
- Le rang et le domaine d'un attribut.

Etape 7: créer les instances

Cette étape consiste à créer les instances qui représentent des entités réelles des classes. On commence par créer une instance individuelle de la classe choisie, puis on la renseigne avec les valeurs des attributs.

2.8.4.c. METHONTOLOGY

La méthodologie de construction d'ontologies « METHONTOLOGY » se situe entre le génie logiciel et L'Ingénierie des connaissances**. [28]** Elle identifie une séquence d'activités techniques à appliquer pour le développement de l'ontologie. L'approche METHONTOLOGY distingue les étapes suivantes :

- **Spécification**

Le développement d'une ontologie commence par la définition du domaine et portée de celle-ci. Cela est basé sur la réponse à certaines questions : Quel est le domaine que l'ontologie va découvrir ? A quoi cette ontologie va servir ? A quels types de questions les informations de l'ontologie doivent fournir des réponses ? Qui va utiliser et maintenir l'ontologie ? , etc. Les réponses à ces questions peuvent changer durant le processus de développement de l'ontologie, mais à chaque étape, elles permettent de limiter la portée du modèle. L'une des solutions qui permet de déterminer la portée d'une ontologie consiste à définir ou planifier une liste de question auxquelles une base de connaissance, basée sur l'ontologie, doit être capable de répondre. **[28]**

- **Conceptualisation**

Elle consiste à identifier et à structurer les connaissances du domaine, à partir des sources d'informations. L'acquisition de ces connaissances peut s'appuyer à la fois sur l'analyse de documents et sur l'interview des experts du domaine. Une fois que les concepts sont identifiés par leurs termes, leur sémantique est décrite dans un langage semi-formel (tables et graphes) à travers leurs propriétés, leurs instances connues et les relations qui les lient entre eux.

- **Implémentation**

Cette étape consiste à formaliser le modèle conceptuel obtenu dans l'étape précédente par un formalisme de représentation d'ontologie telles que les logiques de description. Puis, à coder l'ontologie dans un langage d'ontologie formel.

- **Maintenance**

Cela peut s'agir d'une maintenance corrective ou évolutive de l'ontologie (nouveaux besoins de l'utilisateur), ce qui permet la validation et l'évolution de celle-ci. Cette activité est généralement faite par le constructeur et des experts du domaine. La validation se base sur l'exploitation des services d'inférences associés aux Logiques de Description, et qui sont offerts par des raisonneurs.

2.8.4.d. TOVE

TOVE (Toronto Virtual Enterprise) développée par l'université de Toronto. Cette méthodologie repose sur les expériences de développement d'une entreprise. **[39]** Elle s'appuie également, pour le développement d'une ontologie, sur les principales étapes suivantes :

- **Capturer des scénarios de motivations** : cette étape consiste à identifier des scénarios qui clarifient le domaine que l'on investit et les différentes applications dans lesquelles l'ontologie sera employée.
- **Formuler des questions de compétences informelles :** cette étape consiste à formuler un ensemble de questions (basées sur les scénarios) exprimées en langage naturel, afin de déterminer la portée de l'ontologie. Ces questions et leurs réponses sont utilisées pour extraire les concepts principaux, leurs propriétés et les relations qui existent entre ces concepts.
- **Spécifier la terminologie de l'ontologie :** cette étape consiste à représenter les termes (concepts, propriétés et relations), identifiés dans l'étape précédente, en utilisant le formalisme de la logique du premier ordre. Les concepts seront représentés sous forme de constantes ou bien des variables. Par ailleurs, les propriétés et les relations seront représentées par des prédicats.
- **Evaluer la complétude de l'ontologie.**

Dans ce qui suit, nous allons aborder les différents outils utilisés pour développer une ontologie.

2.9. Outils de développement d'ontologies

Les outils de développement d'ontologies qui existent aujourd'hui sont divers et variés. Cet état des choses suscite beaucoup d'interrogations lorsque vient le moment d'en choisir un pour construire une nouvelle ontologie **[27]** : l'outil offre-t-il une assistance au développement ? Dispose-t-il d'un moteur d'inférence ? Quels langages d'ontologies supporte-t-il ? Offre-t-il un support graphique ?...etc. les réponses à toutes ces questions pourraient s'avérer décisives dans le choix de l'un ou l'autre outil. Dans cette section nous passons en revue les principaux outils disponibles.

2.9.1. Les langages de représentation d'ontologies

Pour pouvoir exploiter une ontologie dans un système informatique, il est nécessaire de la représenter en utilisant un langage de représentation. Il existe plusieurs langages de représentation des connaissances. Dans ce qui suit, nous allons présenter les langages RDF/RDFS et OWL qui sont des recommandations du W3C **[69]**. Mais d'abord nous parlerons du langage XML, qui est la base des langages du Web Sémantique.

2.9.1.a. XML (eXtensible Markup Language)

XML **[70]** est un langage permettant de séparer le contenu des documents des instructions de présentation. Les documents XML sont dits semi-structurés car ils possèdent une structure qui n'est pas imposée par une norme, un standard ou une recommandation, mais une structure que le créateur peut déterminer lui-même au moment de la conception du document. XML est aussi un métalangage (un langage pour écrire d'autres langages), il constitue la base de plusieurs langages comme RDF, RDFS, OWL, etc.

2.9.1.b. RDF (Resource Description Framework)

Proposé dès 1997 par le W3C **[69],** RDF est un modèle conceptuel qui permet la description des ressources Web et leurs métadonnées, de façon à assurer leur traitement automatique. RDF **[71]** est un langage de base du Web sémantique, caractérisé par sa flexibilité et son extensibilité, il a pour raison d'être de permettre que les informations sur les ressources soient manipulées par des applications, plutôt que d'être simplement affichées aux utilisateurs Web, c'est pour cette raison qu'une syntaxe XML a été proposée pour véhiculer des informations modélisées en RDF, en effet, cherchant à rendre la lecture plus compréhensive, plusieurs syntaxes ont vu le jour afin de permettre des descriptions facilement manipulables par la machine, c'est le cas des syntaxes RDF/XML, N-tripes et Notation 3 (N3), et comme il faut bien pouvoir spécifier les contraintes de toute syntaxe utilisée pour représenter un graphe RDF, une syntaxe abstraite a donc été définie.

Syntaxe abstraite :

RDF permet la représentation des métadonnées sous forme d'ensemble de ressources reliées par des liens étiquetés sémantiquement en se basant sur un modèle de triplets {ressource, propriété, valeur} ou bien {sujet, prédicat, objet} où :

Sujet ou Ressource : Représente la ressource à décrire, elle est pointée par une URI,

Prédicat ou Propriété : Représente un type de propriétés applicable à cette ressource, c'est une relation binaire sur le domaine, entre un sujet et un objet,

Objet ou Valeur : Représente une donnée ou une autre ressource attribuée à une propriété de la ressource considérée.

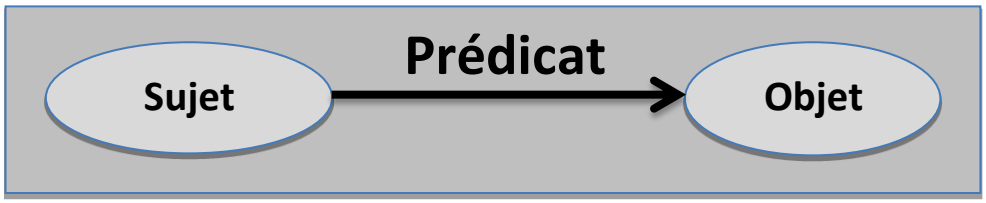


Figure 2.3 : *Structure d'un Triplet RDF.* **[71]**

La ressource est identifiée grâce à un nom unique ou identifiant appelé URI (Universal Ressource Identifier), qui est une adresse locale ou distante de la ressource, en effet, l'utilisation d'URI présente plusieurs avantages, car elles permettent de désambiguïser les désignations utilisées et de permettre à plusieurs applications de partager le même vocabulaire tout en évitant les conflits de noms.

Exemple : [42]

<rdf :Descrption about = '' http://www.lacot.org''/>

<schema :auteur> Avier Lacot </schema :auteur>

</rdf :Description>

2.9.1.c. RDFS (RDF Schéma)

RDFS **[71]** est langage extensible de représentation des connaissances, il fournit des éléments de base pour la définition d'ontologies ou vocabulaires destinés à structurer des ressources RDF, RDFS ajoute à RDF la possibilité de définir des classes, des hiérarchies et des propriétés, et de contraindre leur domaine RDFS permet aussi de définir le vocabulaire pour décrire des classes et des propriétés hiérarchisées en taxinomies.

2.9.1.d. DAML-OIL

DAML est un langage qui a pour but de fournir les fondations pour la génération suivante du Web sémantique. Comme RDFS, ce langage n'est pas assez expressif relativement aux exigences du Web Sémantique, un nouveau langage nommé DAML-ONT a été développé en tant qu'extension de RDF avec les capacités d'un langage de représentation du savoir. En même temps, un nouveau langage nommé OIL a été développé par un groupe de chercheurs pour le même but. Ce langage a une syntaxe basée sur RDF et il est explicitement construit pour que sa sémantique puisse être spécifiée à travers une description logique très expressive, la logique de description de type SHIQ. DAML+OIL est la combinaison de ces deux langages, il hérite des avantages de ces deux langages. En conséquence, DAML+OIL est un langage très expressif et lisible par la machine ainsi que par un être humain avec une syntaxe basée sur RDF. **[72]**

2.9.1.e. OWL

Basé sur la recherche effectuée dans le domaine de la logique de description, OWL **[73]** est un langage d'ontologie Web qui appartient à la famille W3C liées au Web sémantique, c'est une extension de RDF et RDFS qui apporte une meilleure intégration, une évolution, un partage et une inférence plus facile des ontologies. OWL intègre des outils de comparaisons des propriétés et des classes : identité, équivalence, contraire, cardinalité, symétrie, transitivité, disjonction,…etc.

OWL est doté de trois sous-langages offrant des capacités d'expression croissantes, et naturellement, destinés à des communautés différentes d'utilisateurs :

- **OWL Lite :** c'est le plus simple, il est destiné aux utilisateurs qui ont besoin d'une hiérarchie de concepts simple.
- **OWL DL :** il est plus complexe qu'OWL Lite, ce sous-langage permet une expressivité plus importante et garantit la complétude des raisonnements et leur décidabilité.
- **OWL Full :** c'est la version la plus complexe d'OWL, elle permet le plus haut niveau d'expressivité.
-

2.9.2. Les éditeurs d'ontologies

Aujourd'hui, le nombre d'éditeurs d'ontologies a considérablement augmenté. Il existe différents éditeurs qui utilisent des formalismes variés et offrent différentes fonctionnalités. Parmi ces outils on trouve : OILed, OntoEdit, Web ODE, Protégé2000… etc. Voici la description de quelques-uns :

2.9.2.a. OILEd

L'éditeur OIDEd **[74]** a été développé par l'université de Manchester pour éditer des ontologies dans les langages de représentation OIL, il est explicitement orienté vers la représentation

en logique de description expressive, il fournit tous les éléments d'interface permettant de spécifier des hiérarchies de concepts et de rôles ainsi que la construction des expressions complexes définissant ces entités. Les versions disponibles d'OILED ne constituent pas un environnement complet pour le développement d'ontologies d'envergure. En effet, cet outil n'implémente pas la migration et l'intégration d'ontologies, ne gère pas les différentes versions et autres activités impliquées dans la construction d'ontologies. Néanmoins, la simplicité, la robustesse de cet outil et la présence d'un raisonneur logique de description FaCT, capable de tester la faisabilité des ontologies construites ou d'expliciter de nouvelles relations de subsomption entre concepts complexes en font un outil de référence relativement populaire.

2.9.2.b. ONTOEDIT

C'est un outil **[45]** mis au point par l'institut AIFB de l'université de Karlsruhe et qui est maintenant commercialisé par la société Ontoprise GmbH. Il s'inspire de l'approche par Frames, mais gère aussi de nombreux formats libres de la communauté Web Sémantique. C'est un environnement de construction d'ontologies indépendant de tout formalisme, il présente plusieurs avantages qui sont **[45]:**

- Des outils graphiques dédiés à la visualisation d'ontologies sont inclus dans l'environnement.
- Il intègre un serveur destiné à l'édition d'une ontologie par plusieurs utilisateurs.
- Il permet d'éditer une hiérarchie de concepts ou de classes.
- Ontoedit gère la synonymie en admettant plusieurs noms pour un même concept.
- Ontoedit permet d'exporter les ontologies construites dans différents langages.

2.9.2.c. Ontolingua

Développée à l'université Stanford, le serveur Ontolingua **[75]** est le plus connu des environnements de construction d'ontologies en langage Ontolingua. Il consiste en un ensemble d'environnements et de services qui supportent la construction en coopération d'ontologies, entre des groupes séparés géographiquement. Il supporte plusieurs langages et dispose de traducteurs permettant de passer de l'un à l'autre. Ontoligua propose un outil permettant d'inclure une ontologie dans celle en cours de construction.

2.9.2.d. DOE

DOE (Differential Ontologie Editor) a été développé à l'Institut National de l'Audiovisuel par Raphael Troncy et Antoine Issac **[31].** L'éditeur DOE offre des interfaces de création, modification et suppression de concepts et de relations, une représentation graphique de l'arbre ontologique, et des fonctionnalités de recherche et de navigation dans la structure créée. Offre aussi la possibilité de construire les hiérarchies de concepts et relations en utilisant les principes différentiels énoncés par B. Bachimont, puis en ajoutant les concepts référentiels. La sémantique des relations est ensuite précisée

par des contraintes. Ce n'est qu'une fois que l'ontologie ainsi structurée qu'elle est formalisée en utilisant la syntaxe XML.

2.9.2.e. Protégé

Protégé **[33]** a été développé par le Stanford Medical Informatics de l'université de Médecine de Stanford depuis 1995. Il est construit autour d'un modèle de connaissances inspiré par le paradigme des Frames : classes, slots et facettes qui sont les primitives de modélisation proposées. Ce modèle est adapté à la construction d'ontologies depuis la version Protégé 2000. L'interface très complète ainsi que l'architecture logicielle bien pensée permettant l'insertion de plugins. En quelques années cet éditeur s'est imposé comme la référence, avec une communauté d'utilisateurs extrêmement importante et active. Ses nombreuses extensions qui permettent en particulier de gérer des langages standards comme RDF(S) et surtout OWL.

- **Protégé2000 :** C'est une interface modulaire permettant l'édition, la visualisation, le contrôle (vérification des contraintes) d'ontologies, l'extraction d'ontologies à partir de sources textuelles, et la fusion semi-automatique d'ontologie, de nombreux plug-ins sont disponibles ou peuvent être ajoutés par l'utilisateur.

2.9.2.f. WebODE

C'est une plateforme en ligne développée par le groupe Ontological Engineering du département d'Intelligence artificielle de la faculté d'Informatique de l'université polytechnique de Madrid. Elle se place au niveau méthodologique dans la lignée d'ODE, un éditeur qui assurait le support de METHONTOLOGY, la méthodologie proposée par ce laboratoire. L'ambition nouvelle de WebODE par rapport à ODE est de considérer que les ontologies doivent être construites et mises à disposition via le web pour faciliter le développement d'application du web sémantique. WebODE **[76]** est composé de plusieurs modules : un éditeur d'ontologies qui intègre la plupart des services nécessaires à la construction d'ontologies, un système de gestion des connaissances à base ontologique, un générateur automatique de portail du web sémantique, ou outil pour annoter les ressources du web et un éditeur de services pour le web sémantique. La plateforme WebODE met l'accent sur la possibilité d'un travail collaboratif et sur la possibilité, comme dans Protégé, d'étendre la plateforme à l'aide de modules complémentaires, comme un moteur d'inférences, et enfin elle accepte l'export et l'import d'ontologies en RDFS, DAML+OIL et OWL. **[76]**

2.9.2.g. SWOOP

C'est un éditeur d'ontologie développé par l'Université du Maryland dans le cadre du projet MINDSWAP. Contrairement à Protégé, il a été développé de façon native sur les standards RDF et OWL, qu'il prend en charge dans leurs différentes syntaxes. C'est une application plus legere que Protégé, moins évoluée en terme d'interface, mais qui integre aussi des outils de raisonnement. **[77]**

2.9.3. Les moteurs d'inférence

Un moteur d'inférence **[31]** est un mécanisme qui déduit des faits à partir de faits initiaux et de règles suivant l'une des deux approches de base qui sont : le chainage avant (à partir du but, remonter jusqu'aux faits initiaux) et le chainage arrière (partir des faits initiaux pour aller vers le but).Les moteurs d'inférences sont souvent utilisés dans les ontologies afin de déduire de nouvelles connaissances, dans ce qui suit, nous allons présenter deux moteurs d'inférences, Racer et Pellet.

2.9.3.a. Racer

Racer **[68]** est le moteur d'inférence le plus connu et l'un des plus utilisés dans le domaine pour ses performances et sa stabilité. Il est commercialisé par Racer Systems GmbH & Co.KG, fondé en 2004 par Volker Haarslec, Kay Hidde, Ralf Moller et Michael Wessel qui travaillaient à l'université de Hambourg. Racer travaille sur les ontologies modélisées par son langage, mais il accepte des ontologies décrites en RDF ou OWL, ces dernières étant traduites vers le langage utilisé par Racer. Ce moteur d'inférence possède également son propre langage de requêtes nRQL (new Racerpro Query Language) pour interroger l'ontologie sur la TBox et la ABox. Voici un tableau résumant les avantages et inconvénients de Racer :

LE moteur d'inférence RACER	
Avantages	**Inconvénients**
• Permet l'ajout d'assertions et d'individus dans les ABox après le chargement de l'ontologie • Permet l'utilisation de règles SWRL.	• Suppose que toutes les propriétés sur les datatypes sont fonctionnelles. • Ne permet pas l'utilisation de type de données utilisateur • Il n'existe pas de version libre d'utilisation.

Tableau 2.2 : *Les avantages et Inconvenients du Moteur D'inférence RACER.* **[68]**

2.9.3.b. Pellet

Le moteur Pellet **[67]** est beaucoup plus récent. C'est l'un des projets du MINDSWAP Group, un groupe de recherche sur le Web Sémantique de l'université du Maryland. Il est disponible en Open-Source et offre des évolutions fréquentes. Pellet travaille sur des ontologies décrites en RDF ou OWL et permet les requêtes avec RDQL et SPAQL sur la ABox et la TBox. Voici un tableau résumant les avantages et inconvénients du moteur d'inférence Pellet :

LE moteur d'inférence PELLET	
Avantages	**Incovenients**
• Il est open-source et développé en Java, • C'est un risonneur OWL DL complet, • Propose en cas d'incohérence dans l'ontologie des réparations possibles.	• Possede une documentation pauvre en comparaison de celle de Racer. • N'offre pas de système de souscription à un concept.

***Tableau 2.3** : Les avantages et Inconvenients du Moteur D'inférence Pellet.* **[67]**

Le moteur d'inférence Pellet utilise deux langages d'interrogation de requettes qui sont RDQL et SPARQL, Racer quant à lui, utilise sont propre langage de requetes qui est nRQL. Dans la prochaine section, nous allons aborder ces trois langages d'interrogation d'ontologies.

2.9.4. Les langages d'interrogation d'ontologies

Un langage d'interrogation d'ontologies est un outil qui permet d'effectuer des requêtes sur l'ontologie, il joue le rôle d'une interface entre l'utilisateur, ou l'application et l'ontologie, dans ce qui suit nous allons présenter trois langages d'interrogation d'ontologies qui sont RDQL, SPARQL et nRQL.

2.9.4.a. RDQL

RDQL (RDF Data Query Language) **[78]** est un langage d'interrogation de données définies en RDF. Ce langage n'est pas standardisé, sa syntaxe est très proche de SQL (Structured Query Language)

```
SELECT variable [,variable]*
FROM documents_rdf [, documents-rdf]*
WHERE modele_de_triplets
AND restrictions_booléeennes
USING definition_des_raccourcis
```

***Figure 2.4 :**Structure d'une requête RDQL* **[78]**

- **La clause SELECT** définit la liste des variables que l'on désire obtenir, une variable est composée de caractères alphanumériques et commence par ' ? '.
- **La clause FROM** définit l'emplacement des documents RDF utilisés pour la requêtes.
- **La clause WHERE** définit le triplet RDF (sujet – prédicat – objet), les éléments de ce triplet sont décrits soit par les valeurs de l'ontologie interrogée soit par des variables.
- **La clause AND** définit les restrictions booléennes de la requête. Une restriction booléenne est constituée de valeurs ou variables composée à l'aide d'opérateurs.

Voici un exemple de requête RDQL, qui sélectionne l'âge d'une personne dans le document annuaire.rdf sachant que la personne possède un âge inférieur ou égal à 50 ans, que son nom est Dujardin et son adresse email est Dujardit@lifl.fr

```
SELECT ?name, ?email, ?age
FROM<http://www.lifl.fr/ANNUAIRE/annuaire.rdf
WHERE( ?email, vcard :EMAIL, ''Dujardit@lifl.fr'')
AND ( ?age <= 50) && ( ?name EQ ''Dujardin'')
WHERE ( ?email, vcard : EMAIL, ''Dujardit@lift.fr'')
USING vcard FOR <http://www.w3.org/2001/vcard-rdf/3.0/>
```

Figure 2.5 : *Exemple d'une requête RQDL*

2.9.4.b. SPARQL

SPARQL **[79]** est une amélioration de RDQL, ce langage est en cours de standardisation au niveau du W3C. SPARQL ajoute à la syntaxe de RDQL les opérateurs UNION et OPTIONAL dans la clause WHERE.

2.9.4.c. nRQL

nRQL(new Racerpro Query Language**) [46]** est le langage d'interrogation de Racer, comme RDQL et SPARQL, nRQL est basé sur la recherche de graphes RDF, sa syntaxe est proche des deux autres, sauf pour sa notation préfixée des opérateurs. Ci-dessous, nous présentons une requête en nRQL cherchant tous les oncles (variable z) dans l'ontologie myOntology.

```
RETRIEVE( $ ?z )
(AND ($ ?x $ ?y | myOntology # aEnfant | )
( $ ?z $ ?x | myOntology # estFrereDe | ))
```

Figure 2.6 : *Exemple d'une requête nRQL.*

Nous avons présenté dans ce chapitre les ontologies et leur utilisation pour la représention des connaissances. Nous avons parlé de ses différents types, composants, ainsi que les méthodologies de sa construction. Le but de notre travail étant de construire une ontologie pour la sécurité informatique, nous allons consacrer le point suivant aux différents travaux qui ont été faits dans le cadre de l'utilisation des ontologies pour la sécurité informatique ainsi que leur collaboration dans ce domaine.

2.10. Les ontologies et la sécurité informatique

Dans cette partie, nous allons présenter quelques travaux qui ont été réalisés dans le contexte de l'utilisation des ontologies pour la Sécurité Informatique. Les travaux auxquels nous avons eu accès sont :

- **De l'analyse des risques à l'expression des exigences de sécurité des systèmes d'information. [48]**

Le travail fait dans cet article s'inscrit dans le domaine de la sécurité des systèmes d'informations. Il traite les différentes méthodologies d'analyse de risques et de détermination des exigences de sécurité des Systèmes d'information. Mais souligne aussi que rares sont les démarches qui offrent un guidage permettant de dériver les exigences de sécurité à partir des risques. Cet article propose un mécanisme de guidage qui permet de passer, de l'analyse des risques encourus par les entreprises à l'expression des exigences de sécurité. Pour atteindre cet objectif, un alignement des concepts de deux ontologies a été proposé. Cet alignement se fait en élaborant deux ontologies dans un premier temps. Une ontologie des risques fondée sur les concepts inhérents aux différentes méthodes d'identification et d'analyse des risques (Annexe D, D.1), et une autre ontologie des exigences de sécurité(Annexe D, D.2), obtenue en capitalisant sur les concepts constitutifs des méthodes de l'ingénierie des exigences analysées. Puis dans un second temps, faire un alignement de ces deux ontologies qui est fondé sur des relations sémantiques existantes entre leurs concepts. Cet alignement permettra de dériver les exigences de sécurité à partir des risques encourus, apportant ainsi une cohérence dans la gestion de la sécurité et permettra d'arriver à des résultats plus pertinents. La démarche suivie peut être résumée dans la figure 2.7.

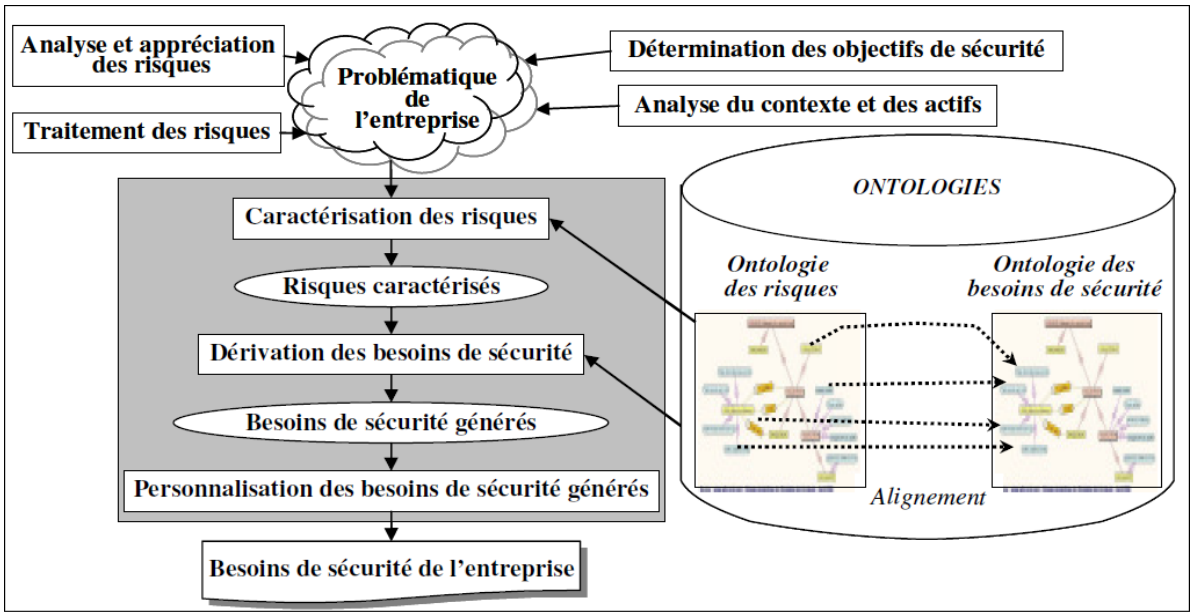


Figure 2.7 : Architecture du schéma de dérivation des besoins de sécurité

Les principaux résultats obtenus peuvent être résumés ainsi :

- La création d'une structure conceptuelle formelle (des concepts et des relations) à partir de la sélection, l'uniformisation, l'intégration et la mise en conformité de sources d'information hétérogènes, diverses, qui ont été créées ou adaptées pour le traitement de la sécurité des systèmes d'information. Une des difficultés résulte du fait que ces méthodologies définissent des concepts avec une syntaxe et/ou une sémantique différente.
- La description d'un processus de guidage pour la dérivation des besoins de sécurité à partir de l'analyse des risques. Cette dérivation a été déduite des bases de connaissances incluses dans chaque méthodologie étudiée.

L'originalité du travail fait réside dans l'interaction de l'analyse des risques avec l'identification des exigences de sécurité. Cette interaction est exprimée à l'aide de liens d'alignement entre les risques et les exigences de sécurité. Elle exploite les connaissances accumulées dans les différentes méthodologies disponibles tant dans le monde académique que dans la pratique des entreprises.

Plusieurs perspectives de recherche future ont été énumérées. La première consiste à enrichir les deux ontologies en introduisant des liens d'association entre concepts qui permettront, par exemple, de détecter des conflits potentiels entre risques et/ou exigences de sécurité. Cette détection de conflits offrirait un meilleur guidage dans la spécification des risques et permettrait de proposer plusieurs scénarios d'exigences à satisfaire. La deuxième perspective concerne l'automatisation de l'étape de caractérisation des risques. Enfin, le processus de validation des ontologies et de l'approche doit être poursuivi pour tirer les leçons nécessaires à l'amélioration de la démarche. **[48]**

- **Modélisation et classification automatique des informations de sécurité [49]**

De nombreux travaux permettant de représenter le domaine de la sécurité ont été menés parmi lesquels on retrouve celui de Benali Fatiha lors de sa thèse sur la « Modélisation et classification automatique des informations de sécurité ». L'auteur commence par mettre l'accent sur les Systèmes de Détection d'Intrusions en spécifiant le but, la source des données et l'approche utilisée pour la modélisation. Elle considère que l'intrusion apparaît comme étant «une action non autorisée ou L'intrus peut être externe comme il peut être un utilisateur interne qui tente de dépasser ses privilèges». Il est difficile pour un administrateur sécurité de détecter ou d'être informer sur une quelconque tentative d'intrusion avant que l'attaquant ne soit en interaction directe avec le système. Deux expérimentations ont été faite sur l'ontologie. La première classifie des événements bruts programmés dans des produits hétérogènes et la deuxième expérience teste l'ontologie dans un milieu opérationnel qui est le réseau d'une entreprise pour évaluer la sécurité d'un SI. Ces démarches ont été mises en créant une ontologique nommée « Ontology for Intrusion Detection (OID) », représentant ainsi la classification des informations comme étant une caractéristique d'un type particulier, se basant à la fois sur la théorie de l'action pour définir les concepts de l'ontologie, et sur la démarche de

l'attaquant pour définir les intentions dans le SI, les mouvements et les cibles qui vont avec. L'auteur reprend les taxonomies établies pour la sécurité informatique (Annexe D, D.3).**[49]**

- **Construction d'une ontologie pour le domaine de la sécurité : Application aux agents mobiles. [31]**

Le travail fait dans ce mémoire s'inscrit dans le domaine de la sécurité informatique et plus particulièrement les agents mobiles. Les systèmes ouverts utilisant le paradigme d'agent mobile ont besoin, de plus en plus, de souplesse et d'efficacité pour garantir une meilleure communication et réussir l'interopérabilité entre les différentes entités, en particulier en ce qui concerne la sécurité. Désormais, la croissance de complexité et d'extensibilité au sein des environnements dynamiques empêche l'utilisation d'une politique de sécurité commune, le développement d'une architecture unique préalablement établie et augmente la difficulté d'appliquer un système sécuritaire unifié. L'agent mobile présente des problèmes de sécurité qui limite son utilisation, entrave son extension et son application. Cela exige une alternative pour pallier au problème d'hétérogénéité en fournissant une sémantique interprétable, claire, et partagée entre les entités communicantes. Afin de réaliser l'interopérabilité et résoudre la question d'hétérogénéité de la politique de sécurité des agents mobiles, l'auteur indique qu'une intégration sémantique est nécessaire, et précise que l'utilisation d'une ontologie est la solution idéale pour résoudre le problème d'hétérogénéité. Car elle offre une sémantique partagée capable d'empêcher l'échec de communication et d'interaction entre les agents mobiles dû à l'hétérogénéité de leurs propriétés sécuritaires.

Le but principal de ce mémoire est la construction d'une ontologie pour le domaine de la sécurité et son application aux agents mobiles, selon un processus de construction d'ontologies et en se basant sur des informations concernant les propriétés et procédés sécuritaires utilisées par l'agent mobile et l'hôte. L'utilisation de cette ontologie permettra d'éliminer les différences sémantiques qui existent au niveau des objets, attributs, et structures de données de politiques de sécurité pour faciliter l'interopérabilité des agents mobiles. Le diagramme des concepts et des des relations faits seront présentés en (Annexe D, D.3 et D.4).

Comme perspective, l'auteur souligne que la spécification et la mise en œuvre d'une architecture ouverte basée sur les ontologies pour les agents mobiles et leurs plates-formes, même en se limitant aux propriétés de sécurité s'avère être une tâche complexe, Il propose quelques scénarios qui montrent comment utiliser l'ontologie construite et comment l'ingérer dans le système des agents mobiles.

2.11. Conclusion

Les ontologies apparaissent désormais comme une clé pour la manipulation automatique de l'information au niveau sémantique. Au fur et à mesure des recherches, des idées se dégagent autour du contenu des ontologies, des méthodes à utiliser pour les construire et des modèles et langages servant à leur représentation.

Au cours de ce chapitre, nous nous sommes intéressés à ce qu'était une ontologie, et pour cela nous sommes partis des origines philosophiques du terme pour définir son sens en ingénierie des connaissances. Ensuite, nous avons étudié la manière de concevoir et de réaliser une ontologie en ingénierie des connaissances en énumérant ses composants, ses types, et les différents formalismes de sa représentation. Nous avons aussi abordé les points techniques en présentant les différents outils de construction d'une ontologie.

Aujourd'hui les domaines d'utilisation des ontologies s'élargissent considérablement comme nous l'avons vu, dans les systèmes à base de connaissance et le Web sémantique. Le prochain chapitre sera consacré au développement d'une ontologie pour la sécurité informatique. Nous présenterons les différentes étapes du processus suivi et nous décrirons la méthodologie qui a servi de base à la construction puis nous nous reprendrons ce processus pour l'appliquer à notre cas, la sécurité informatique.

Chapitre III

Conception d'une ontologie pour la sécurité informatique

3.1. Introduction

L'utilisation d'une ontologie pour la sécurité informatique permettra d'éliminer et de relever les ambiguïtés qui se trouvent au niveau des définitions des concepts de la sécurité informatique. Elle assurera l'interopérabilité entre les différents systèmes et applications de gestion et de management des risques, et permettra la réutilisation des informations ainsi qu'une indépendance de l'implémentation à des fins personnelles ou une pour une continuité de travail. Elle permettra également de déduire de nouvelles connaissances à partir des informations recueillies. De plus, l'ontologie de la sécurité informatique apportera des solutions à la gestion des risques et de la sécurité au sein d'une entreprise et ouvrira une porte à la recherche avec de nouvelles perspectives.

3.2. Présentation de notre travail

Nous proposons de construire une ontologie pour la sécurité informatique qui sera utilisée par les applications qui gèrent cette dernière, en apportant :

- Un vocabulaire et une terminologie communs pour le domaine de la sécurité informatique, qui constitueront le terrain d'entente de tous les spécialistes de la sécurité,
- Une architecture ouverte et globale pour la sécurité informatique et ses différents concepts,
- Offrir un moyen de relier les différentes étapes de l'élaboration d'une politique de sécurité, afin d'assurer une meilleure cohérence, et d'éliminer les divergences pour une gestion ciblée et correcte des risques qu'une entreprise peut avoir.

Et pour cela, nous procédons comme suit :

- Préparer les ressources documentaires à utiliser pour extraire les informations à modéliser dans l'ontologie. Pour notre travail, nous nous intéressons à la documentation de la Méthode MEHARI, La base de connaissances de MEHARI 2010 et ses questionnaires, la documentation du site du CLUSIF et la documentation CISCO CCNA4 et certains articles traitant la sécurité informatique.
- une fois que les notions, principes et concepts de la sécurité informatique sont assimilés, nous délimitons le contexte de notre travail. Ce contexte est l'un des domaines de la sécurité des systèmes d'information. Il sera représenté par l'ontologie.
- Notre contexte porte sur les attaques qui utilisent l'outil informatique et qui visent un des actifs d'une entreprise. Ces attaques peuvent exploiter la vulnérabilité d'un actif, pour attaquer un ou plusieurs actifs. Elles ont des conséquences qui peuvent être plus ou moins graves et peuvent atteindre un ou plusieurs objectifs de sécurité informatique. Pour se protéger contre ces attaques, il existe plusieurs mesures de sécurité qui sont mise en place au sein d'une entreprise. Ces contre-mesures protègent les actifs, réduisent leurs vulnérabilités et préviennent les attaques.

- Nous utilisons les ressources documentaires pour extraire manuellement les termes relatifs à notre contexte. Parmi ces termes, nous pouvons citer comme exemple : Protection, Attaque, Pirate, Déni de service, Authentification, Intrusion, …etc.
- Une fois que les termes sont définis et spécifiés, nous continuons à appliquer le processus de Methontology. Nous passons à l'étape de conceptualisation, pour identifier les concepts, les relations entre eux et leurs instances. Puis nous identifions les différentes hiérarchies des concepts, en s'appuyant sur les contraintes et les règles imposées par Methontology.
- Apres la conceptualisation, il est nécessaire de passer à la formalisation de l'ontologie obtenue. Pour réaliser cette partie, nous utilisons les logiques de description pour formaliser notre ontologie.
- une fois que l'ontologie est formalisée, il faudra l'implémenter en utilisant un éditeur d'ontologie afin de générer le fichier OWL qui constituera notre ontologie. Pour ce faire, Nous utilisons l'éditeur Protégé version 3.5.
- Une fois l'implémentation faite, il faudra tester la consistance de l'ontologie en utilisant un raisonneur qui fera des tests et repèrera les erreurs et les ambigüités. Il est aussi possible d'intégrer des raisonneurs à protégé, comme Pellet et Racer **[48].**
- L'ontologie conçue, nous pouvons générer le fichier OWL, et utiliser un programme Java pour l'interroger et faire des inférences. Nous pouvons utiliser l'API Jena pour interroger notre ontologie et exécuter des requêtes SPARQL **[48].**

Cette partie de notre travail permettra de présenter la méthodologie que nous avons appliqué et les différents outils qui ont été utilisés. Nous citerons :

- La méthode de construction d'ontologie METHONTOLOGY.
- Le formalisme de la logique de description
- Le langage OWL.

3.3. Processus de construction de l'ontologie

Pour la construction de notre ontologie, nous utilisons un processus de construction partant de connaissances brutes pour arriver à une ontologie d'application opérationnelle, représentée par le langage OWL. Nous utilisons la méthode Methontology **[26]** qui a été développée au laboratoire d'intelligence artificielle de l'université de Madrid. Methontology propose un processus complet de gestion du cycle de vie des ontologies : gestion de projet, développement de l'ontologie et support.

Methontology propose une démarche constituée de cinq étapes :

- Spécification des besoins.
- Conceptualisation.
- Formalisation.

- Implémentation.
- Test et évolution de l'ontologie.

Le choix de cette méthodologie est accentué par le fait que :

- Elle s'inspire d'une méthode de développement de systèmes à base de connaissances. Elle spécifie de façon très détaillée l'étape de conceptualisation.
- Elle utilise des structures intermédiaires qui facilitent la communication entre le concepteur de l'ontologie et l'expert du domaine, afin de bien structurer et vérifier les connaissances collectées.
- Ses structures permettent une construction pas à pas de l'ontologie et situent le concepteur par rapport à l'objectif fixé.
- L'étape de conceptualisation permet d'obtenir des ontologies semi-formelles qui sont d'une part facile à comprendre et d'autre part, indépendantes de tout formalisme de représentation.

Processus de la Méthode Methontology

Etape 1 : Spécification

Cette étape consiste à établir un document formel de spécification des besoins, ce dernier permet de décrire l'ontologie à construire à travers les cinq aspects suivants :

- **Le domaine de connaissances :** déterminer aussi précisément que possible le domaine que va couvrir l'ontologie.
- **L'objectif :** le but de l'ontologie à créer pour le domaine considéré.
- **Les utilisateurs :** identifier les futurs utilisateurs de l'ontologie à créer.
- **Les sources d'informations :** déterminer les sources des connaissances d'où les connaissances seront obtenues.
- **La portée de l'ontologie :** déterminer la liste des termes les plus importants pour le domaine à représenter.

Etape 2 : Conceptualisation

C'est l'étape la plus importante dans le processus de construction d'une ontologie. Elle consiste à identifier et à structurer, à partir des sources d'informations, les connaissances du domaine. Elle permet d'aboutir à un ensemble de représentations intermédiaires semi-formelles indépendamment des langages de formalisations à utiliser pour représenter l'ontologie. A la fin de cette phase, nous obtenons une ontologie conceptuelle. Cette étape se fait comme suit :

- Créer un glossaire de termes, que l'on divise en concepts et verbes. Les concepts vont devoir être regroupés en arbres de classification de concepts et les verbes serviront à créer des diagrammes de relations binaires.

- ❖ A partir des deux structures, on va construire un dictionnaire des concepts, qui regroupe toutes les informations concernant les dits concepts (nom, synonymes, instances, attributs de la classe et de ses instances, relations rattachées au concept).
- ❖ D'autres structures vont également apparaitre : table des relations binaires, table des attributs d'instances, table des attributs de classes, tables des axiomes logiques, table des constantes, table des formules, arbres de classifications des attributs et tables des instances.

Le processus peut se résumer en une suite de tâches :

1. Construction du glossaire des termes.
2. Construction du diagramme de classification de concepts.
3. Construction du diagramme de relations binaires.
4. Construction du Dictionnaire de concepts
5. Construction de la Table des relations binaires.
6. Construction de la Table des attributs.
7. Construction de la Table des axiomes logiques.
8. Construction de la Table des instances.

Etape 3 : Formalisation

Cette phase consiste à formaliser l'ontologie conceptuelle obtenue dans la phase précédente afin de faciliter sa représentation ultérieure dans un langage complètement formel et opérationnel. Notre choix s'est porté sur le formalisme de représentation qui est la logique de description en s'appuyant sur la syntaxe de la logique de description de type SHOIN qui présente une logique de description très expressive et offre un ensemble de constructeurs riche pour décrire les concepts.

La logique de description est constituée de deux parties : une partie terminologique (TBox) permettant de décrire les concepts et les rôles et d'une partie assertionnelle (ABox) décrivant les instances.

Etape 4 : Implémentation

L'ontologie que nous avons obtenue dans la phase de la formalisation est appelée une ontologie formelle. Le but de cette étape sera donc de coder l'ontologie formelle en OWL DL qui dispose de fonctionnalités sémantiques plus riches que ses prédécesseurs RDFS ou DAML+OIL. A la fin de cette phase, nous obtenons une ontologie opérationnelle.

Afin de faciliter le processus de codification, nous utilisons PROTEGE OWL version 4.3 qui dispose d'une interface modulaire, développée au Stanford Medical Informatics de l'université de Stanford, permettant l'édition, la visualisation et le contrôle d'ontologies, et contient des classes

(concepts), des slots (propriétés) et des facettes (valeurs des propriétés et contraintes), ainsi que des instances des classes et des propriétés.

Etape 5 : Tests et évaluation

Cette étape consiste à exploiter les services d'inférence fournis par la logique de description afin de supporter le processus de construction et d'améliorer la qualité de l'ontologie. Pour ce faire, nous proposons l'utilisation de l'outil RACER, un système de la logique de description. Ce dernier, permet de lire un document au format OWL (ontologie OWL) et de le représenter sous forme d'une base de connaissances LD et de fournir des services d'inférence pour les niveaux TBox et ABox.

Cette étape sert aussi à suivre l'évolution de l'ontologie, c'est-à-dire les nouveaux concepts à ajouter dans la partie terminologique (TBox) de l'ontologie. Une classification a lieu à chaque fois qu'une définition de concept est nouvellement créée. Le mécanisme de raisonnement de base des logiques de description est la classification de concepts. Elle est réalisée par un algorithme de classification, appelé « le classificateur ». Le classificateur utilise la description d'un nouveau concept pour le placer à l'endroit correspondant dans la hiérarchie. Afin de trouver la place appropriée au nouveau concept, l'algorithme de classification détermine les relations de subsomption entre ce concept et les autres. Ces relations peuvent être spécifiées directement, trouvées par transitivité ou calculées à partir de la sémantique des conditions des rôles. La recherche de la place correcte pour le nouveau concept comporte trois étapes :

- La recherche des subsumant les plus spécifiques SPS (Concepts qui subsument le concept à classer et dont les fils ne le subsument pas).
- La recherche des subsumés les plus généraux SPG (Concepts subsumés par le concept à classer et dont les pères ne sont pas subsumés par lui).
- Insertion du nouveau concept dans la hiérarchie.

3.4. Les logiques de description

Les logiques descriptives représentent l'une des familles de formalismes utilisés pour présenter une base de connaissances. Ils permettent aussi de raisonner efficacement pour minimiser les temps de réponses. Ce formalisme autorise la représentation des classes d'un domaine et les relations pouvant exister entre les instances et les classes. **[47]**

3.5. OWL

Le langage OWL devient en 2004, une recommandation du W3C. Il découle de RDF et RDFS, possédant ainsi des connecteurs logiques, permettant d'exprimer des cardinalités sur les propriétés et d'en spécifier la nature. **[48]**

OWL permet donc de définir des ontologies qui sont l'association de définitions de termes d'un domaine particulier et des relations entre ces termes. Son objectif est de fournir au travers des ontologies des moyens de raisonner sur des Métadonnées et de déduire de nouveaux faits/assertions.

3.6. Conception d'une ontologie pour la sécurité informatique

Apres le choix de la méthode et des outils, nous passerons à la construction de notre ontologie. Et pour cela, nous allons présenter notre conception suivant le processus de Methontology.

Etape 1 : Spécification

La première étape du processus consiste à établir un document formel de spécification des besoins qui permet de décrire l'ontologie à construire à travers les cinq aspects suivants :

- **Le domaine de connaissances :** L'ontologie que nous allons construire s'inscrit dans le cadre de la sécurité informatique, plus précisément les risques auxquels est confrontée une entreprise. Pour le besoin de notre travail, nous considérons des concepts qui caractérisent le contexte des attaques informatiques qui peuvent avoir lieu au sein d'une entreprise et qui peut toucher un de ses actifs.
- **L'objectif de l'ontologie :** L'objectif de notre ontologie est de :
  - ❖ Analyser et modéliser un ensemble de connaissances de la sécurité informatique.
  - ❖ Définir un vocabulaire commun pour les responsables de la sécurité qui ont besoin de partager l'information sur la sécurité.
  - ❖ Permettre la ininterrompue de l'information.
  - ❖ Permettre la réutilisation par d'autres ontologies et applications.
- **Les utilisateurs futurs :** les responsables de sécurité, les responsables administratifs, les employés d'une entreprise, les éditeurs des méthodes de gestion des risques,…
- **Les sources d'informations :** Nous allons utiliser comme principales sources d'informations, la documentation de la méthode de gestion des risques MEHARI**,** sa base de connaissances version 2010 les rapports et articles publiés par le CLUSIF. Nous nous sommes basés sur un les quatorze questionnaires de la base de connaissances MEHARI 2010 **[05]**, traitant la gestion des risques dans une entreprise. Nous avons exploité la documentation de CISCO CCNA4 **[20]** et bien d'autres corpus.

- **La portée de l'ontologie :** La portée de l'ontologie est une liste des termes les plus importants. Parmi ces termes, nous pouvons citer : Vulnérabilité, Actif, Conséquence, Attaque, Attaquant, Contre-mesure, Outil d'attaque, etc.

L'ontologie que nous allons développer doit être en mesure de répondre aux questions suivantes :

- Quelles sont les différentes attaques contre les systèmes informatiques ?
- Qui effectue ces attaques ?
- Quel est l'outil utilisé pour effectuer ces attaques ?
- Quelles vulnérabilités sont exploitées pour effectuer ces attaques ?
- Quelles sont les conséquences de ces attaques sur l'entreprise ?
- Comment se protéger contre ces attaques ?

Etape 2 : Conceptualisation

Cette étape permet de créer plusieurs représentations intermédiaires par le biais de corpus et dictionnaires. Dans notre cas, nous utilisons les ressources de MEHARI présentées dans le point précédent. Une fois que la majorité des connaissances est acquises. Nous les organisons et les structurons en utilisant des représentations intermédiaires semi-formelles qui sont faciles à comprendre et indépendantes de tout langage d'implémentation. Dans cette phase, nous procédons à l'élaboration des glossaires et listes des termes, concepts, relations, attributs et instances de notre future ontologie.

Construction du glossaire des termes :

Ce glossaire contient la définition de tous les termes relatifs au domaine de la sécurité informatique et la gestion des risques. Ces termes serviront à construire les concepts de l'ontologie finale. Le tableau X fournit une liste des termes les plus importants qui seront plus tard repris dans l'ontologie.

Terme	Synonyme	Définition
Actif	Asset, Bien, Ressource	tout élément ayant de la valeur pour une entreprise et nécessitant une protection. Il existe deux types d'actif : actif primaire et actif secondaire.
Anti-virus	-	C'est un logiciel conçus pour identifier, neutraliser et éliminer des logiciels malveillants.
Attaquant	-	Personne qui effectue une attaque.
Attaque	Attack	activité malveillante qui exploite la vulnérabilité d'un système informatique à des fins non connues par les responsables du système et généralement préjudiciaires
Authentification	Identification	Procédé permettant de vérifier l'identité d'une entité.
Backdoor	Porte dérobée	Dans un logiciel, une porte dérobée est une fonctionnalité inconnue de l'utilisateur légitime, qui donne un accès secret au logiciel et permet de prendre le contrôle à distance.
Base de données	-	Une base de données est un ensemble structuré et organisé permettant de stockage de grandes quantités d'informations afin d'en faciliter l'exploitation.

Broadcast	Diffusion, Ping	Commande informatique permettant d'envoyer une requête d'un ordinateur local vers tous les autres ordinateurs du réseau.
Bombe logique	Fork Bomb	Logiciel malveillant capable de se déclencher suite à un évènement particulier.
Botnet	Réseau de Pc Zombies,	Réseau de PC zombies contrôlés à distance par un pirate informatique pour lancer des attaques de masse.
Buffer Overflow	Depassement de tampon	Attaque qui consiste à mettre plus d'informations en mémoire que celle-ci n'est disposée à en recevoir.
Canular	Hoax	un mail envoyé dont l'information est volontairement erronée pour gagner en popularité. Incitant les destinataires abusés à effectuer des opérations ou à prendre des initiatives inutiles, voire dommageables.
Cheval de Troie	Trojan, Troyen	Programme informatique ouvrant une porte dérobée dans un système pour y faire entrer un hacker ou d'autres programmes indésirables.
Chiffrement	-	Procédé grâce auquel on peut rendre la compréhension d'un message impossible à toute personne qui n'a pas la clé de chiffrement.
Cracker	-	Attaquant qui a comme but de créer des outils logiciels permettant d'attaquer des systèmes informatiques ou de casser les protections contre la copie des logiciels payants.
Confidentialité	-	Objectif de la sécurité informatique qui assure que l'information ne doit être accessible qu'aux personnes autorisées.
Conséquence	-	C'est le résultat de la réalisation d'un risque redouté suite à une attaque.
Contre-mesure	-	C'est l'ensemble des actions, matériels, procédures, logicielles mises en œuvre en prévention de la menace.
Contrôle d'accès	Access-Control	Dispositif qui valide les demandes d'accès à des ressources.
Dégradation	-	Diminution de la qualité d'un service ou d'un système.
Déni de service	-	Rendre indisponible pendant un temps indéterminé les services ou ressources d'une entreprise.
Disponibilité	-	la disponibilité vise à ce qu'un système soit capable d'assurer ses fonctions sans interruption ou dégradation, au moment même où la sollicitation en est faite.
Espionnage	-	Observer, surveiller sournoisement quelqu'un dans ses actions ou dans ses discours pour en faire un rapport.
Faille	-	Dysfonctionnement ou défaut de protection dans un logiciel pouvant être exploité manuellement ou par un programme. Elle permet une intrusion sur un ordinateur à distance, afin d'exécuter un programme malveillant ou provoquer une déconnexion d'Internet.
Firmwall	-	Logiciel permettant le fonctionnement d'un composant informatique. Il est stocké sur la mémoire de celui-ci. Il est principalement utilisé pour contrôler directement le matériel.
Firewall	Pare-feu	Dispositif de sécurité destiné à protéger un réseau interne informatique des attaques extérieures par un filtrage des informations provenant d'un réseau publique.
Hacker	-	Pirate informatique ou expert en informatique spécialiste du forçage des systèmes de sécurité et de l'intrusion dans les sites protégés.
Honeypot	-	Système informatique public volontairement vulnérable à une ou plusieurs failles connues visant à attirer les pirates. Le but est d'étudier leurs stratégies d'attaque pour mieux les comprendre et les anticiper. En ciblant un honeypot, le pirate croit que vous exécutez des services vulnérables qu'il peut utiliser pour arrêter ou s'introduire votre système.
Hôte	Host	Ordinateur ou machine qui dispose de ressources particulières, et qui est connecté à un réseau
Informatique	Computer Science	domaine d'activité scientifique, technique et industriel concernant le traitement automatique de l'information par des machines
Information	-	Interprétation d'une donnée en fonction de critères relatifs à un point de vue.
Intrusion	infiltration	Accès non autorisé à un système informatique .
Keylogger	Enregistreur de frappes	Dispositif chargé d'enregistrer les frappes de touches du clavier à l'insu de l'utilisateur.
Logiciel		Le logiciel est une série d'instructions interprétables par un ordinateur.
Malware	programme malveillant, malicien	Programme ou partie d'un programme destiné à perturber, altérer ou détruire tout ou partie des éléments logiciels indispensables au bon fonctionnement d'un système informatique
Malveillance		Intention de nuire
Matériel	Hardware	l'ensemble des pièces électroniques nécessaires au fonctionnement des appareils informatiques.
Menace	Threat	Source potentielle d'incident pouvant entrainer des changements indésirables sur un actif, un ensemble d'actifs, ou l'entreprise.
Méthode de	-	C'est une organisation méthodologique mise en œuvre afin de gérer les risques de

gestion des risques		manière efficace.
Mise à jour	-	Une mise à jour permet d'améliorer un programme informatique en corrigeant les défauts ou en modifiant certaines lignes du code source.
Mot de passe	Password, Code	Séquence de caractères utilisée par un usager pour valider son accès à des ressources personnelles
Objectif de sécurité	-	Un objectif de sécurité est le résultat que la sécurité veut atteindre
Ordinateur	PC	Appareil électronique capable en appliquant des instructions prédéfinies d'effectuer des traitements de données et d'interagir avec l'environnement grâce à des périphériques.
Outil	accessoire	Ce qui permet de faire une action ou un travail.
Personnel	-	Ensemble de personnes au service de l'entreprise.
Pharming	-	Technique de piratage informatique qui exploite des vulnérabilités DNS afin de récupérer les données d'une victime.
Phishing	Hameçonnage	Technique frauduleuse utilisée par des pirates informatiques pour récupérer des informations auprès des internautes.
Politique de sécurité	-	Document de référence définissant les objectifs poursuivis en matière de sécurité et les moyens mis en œuvre pour les assurer.
Protocole	-	Ensemble de conventions nécessaires pour faire coopérer des entités distantes, en particulier pour établir et entretenir des échanges d'informations entre ces entités.
Proxy	-	Serveur recevant des requêtes qui ne lui sont pas directement destinées et qui les transmet aux autres serveurs
Protection	Assurance	Dispositif de sécurité isolant les opérateurs d'un danger potentiel.
Réseau	-	C'est un ensemble d'hôtes reliés entre eux par des canaux de communications qui leur permettent d'échanger des informations.
Risque	-	C'est la possibilité qu'une menace exploite une vulnérabilité intrinsèque d'un actif et ainsi causer un préjudice à l'organisation
Script kiddies	Lammer, Crasher, Packet Monkey.	Adolescent utilisant des programmes trouvés sur internet pour effectuer des attaques aléatoirement.
Sécurité Informatique	-	ensemble de moyens techniques, organisationnels, juridiques et humains nécessaires et mis en place pour conserver, rétablir, et garantir sa sécurité.
Site	Local, Salle.	Situation du lieu ou du terrain où s'élèvent une ville, un village, une station, un monument, etc.
Site Web	-	Ensemble de pages Web accessibles via Internet sur un serveur identifié par une adresse.
Sniffing	Scanning, analyse du trafic	C'est l'interception massive de données qui transitent sur un réseau.
Sniffer	Analyseur du réseau.	C'est un outil ou logiciel qui surveille et identifie le trafic de paquets réseaux.
Sneaker		Personne qui tente de pénétrer dans un système dans le but de tester leur sécurité.
Spam	pollupostage, pourriel, junk mail, courrier indésirable.	Le spam est un courrier électronique non sollicité, envoyé de manière répétitive, souvent à une liste de destinataire obtenue de manière aléatoire.
Spoofing	-	Usurper l'identité d'un utilisateur
Système	-	Ensemble d'élément en interaction dynamique, organisés en fonction d'un but.
Système d'exploitation	-	C'est un logiciel permettant le fonctionnement d'un ordinateur, pour installer des programmes et d'assurer leur fonctionnement.
Téléchargement	Download	Le **téléchargement** est une opération consistant à transférer des fichiers, d'un ordinateur à un autre via un canal de transmission.
Utilisateur	-	Personne qui utilise un système informatique.
Ver	Worm	C'est un programme malveillant qui se copie d'ordinateur en ordinateur.
Virus	-	Tout programme capable d'infecter un autre programme d'ordinateur en le modifiant de façon à ce qu'il puisse à son tour se reproduire.
Vulnérabilité	-	Faille dans un actif ou dans une mesure de sécurité qui peut être exploitée par une menace.
Web	-	Partie de l'Internet qui est composée des pages Web stockées sur les serveurs et affichées par les clients appelés navigateurs.

Tableau 3.6 : Glossaire des termes relatifs à la sécurité informatique.

Construction du diagramme de classification de concepts :

Dans cette étape, nous construisons le diagramme de classification des concepts. La hiérarchie de classification de concepts démontre l'organisation des concepts de l'ontologie dans un ordre hiérarchique qui exprime les relations sous-classe. Un concept universel « Thing », qui généralise tous les concepts racines des différentes hiérarchies de concepts est utilisé pour former une seule hiérarchie globale. Pour construire la taxonomie des concepts, Methontology propose d'utiliser les quatre relations : Subclass-Of, Disjoint-Décomposition, Exhaustive-Décomposition, et Partition.

- Un concept C1 est une sous-classe de concept C2 si et seulement si toute instance de C1 est une instance de C2.
- Une Disjoint-Décomposition d'un concept C est un ensemble de sous-classes de C qui ne couvrent pas C et n'ont pas des instances communes.
- Une Exhaustive-Décomposition d'un concept C est un ensemble de sous-classes de C qui couvrent C et peuvent avoir des instances communes.
- Une partition d'un concept C est un ensemble de sous-classes de C qui couvrent C et n'ont aucune instance commune.
-

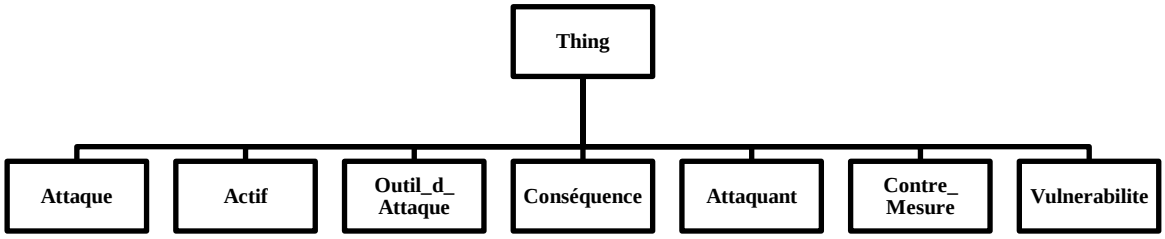


Figure 3.1 : Diagramme de classification des concepts

Chaque concept possède une hiérarchie extraire du dictionnaire des termes, en appliquant les règles imposées par Methontology. Dans ce qui suit, nous allons présenter les hiérarchies des concepts sous-concept de Thing. Qui sont : Attaque, Actif, Outil_d_attaque, Conséquence, Attaquant et Contre_Mesure.

1. La hiérarchie du concept Attaque

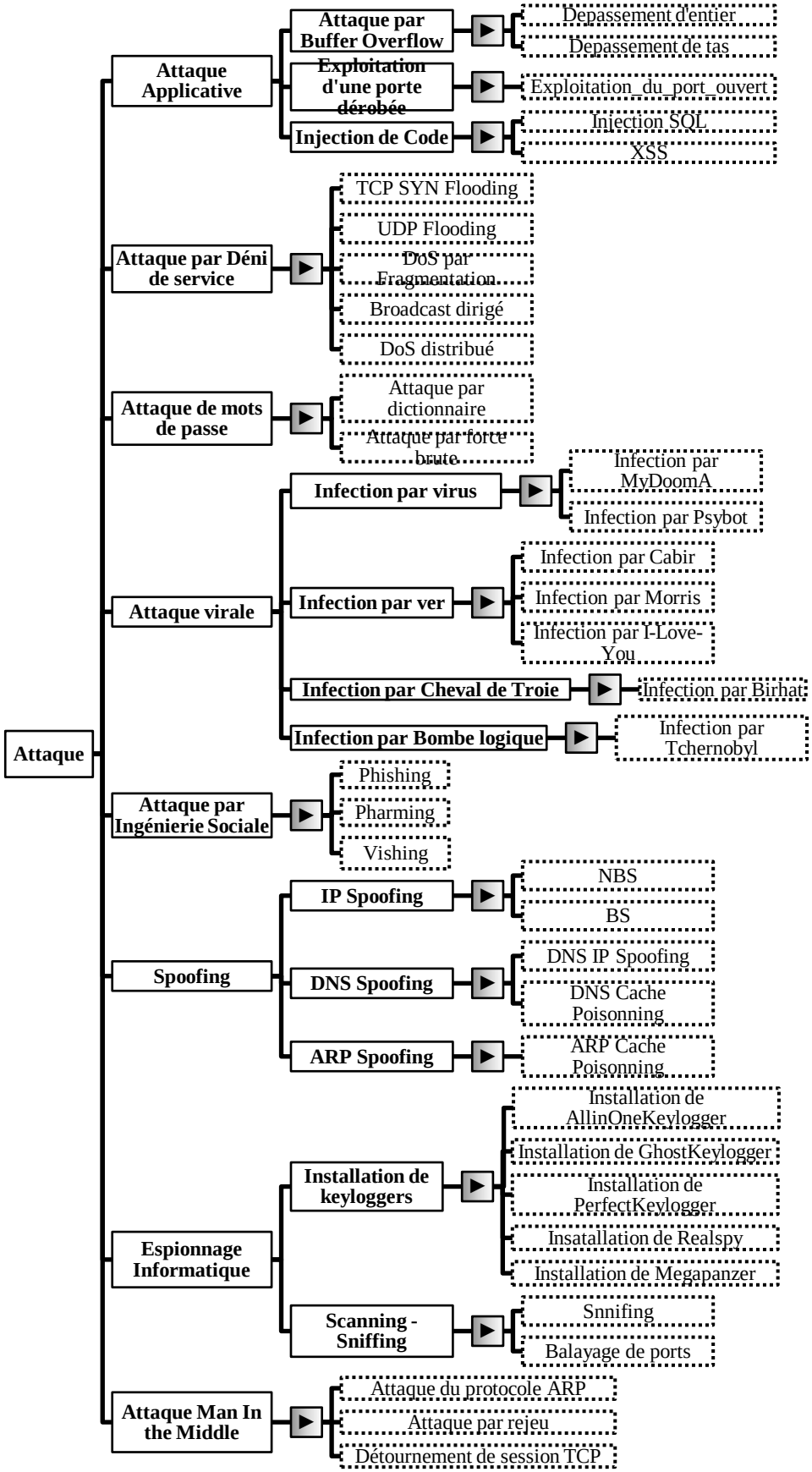


Figure 3.2 : Digramme de classification des concepts, Hiérarchie des Attaques

2. La hiérarchie du concept Actif

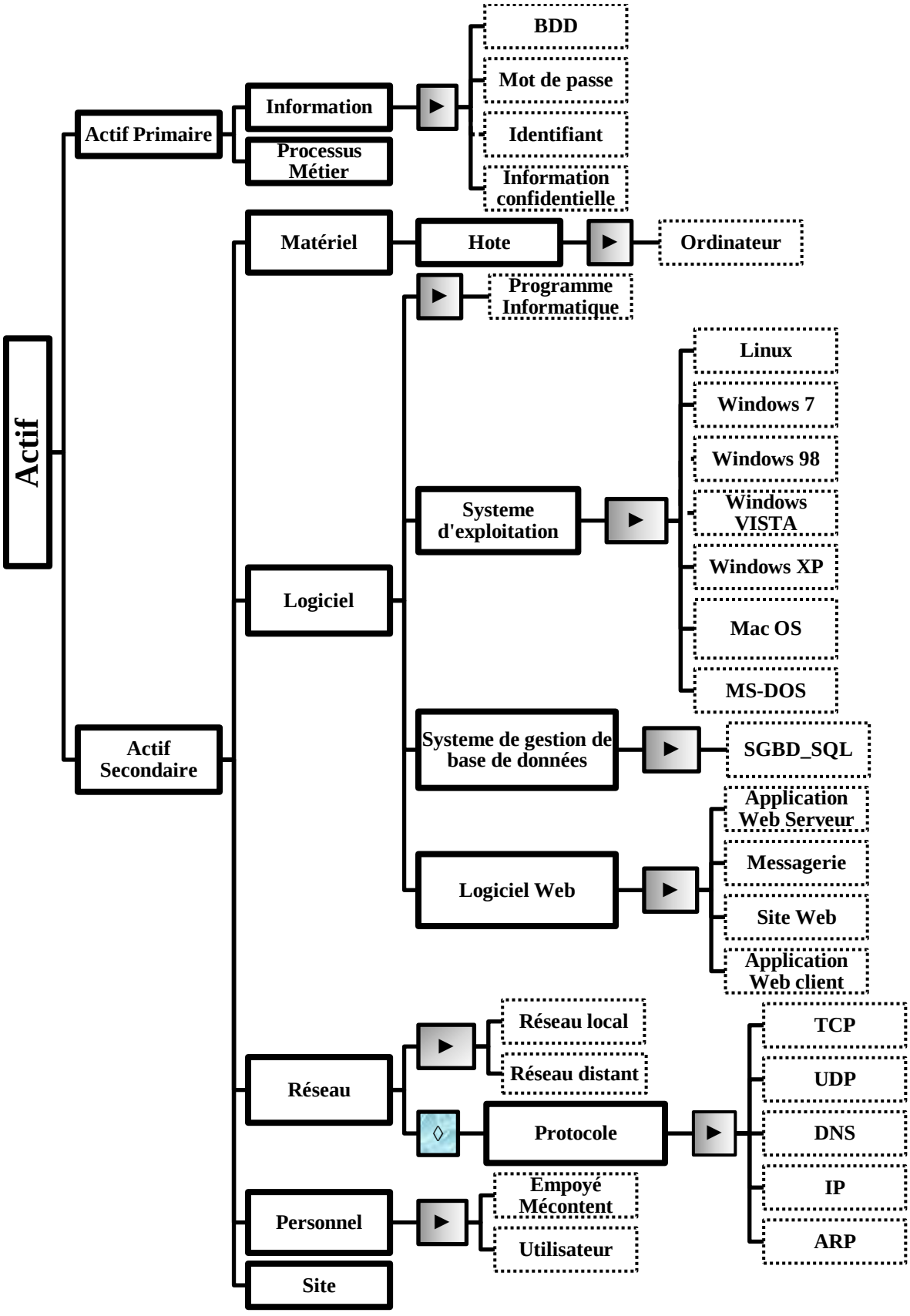


Figure 3.3 : Digramme de classification des concepts, Hiérarchie des Actifs

3. La hiérarchie du concept Vulnérabilité

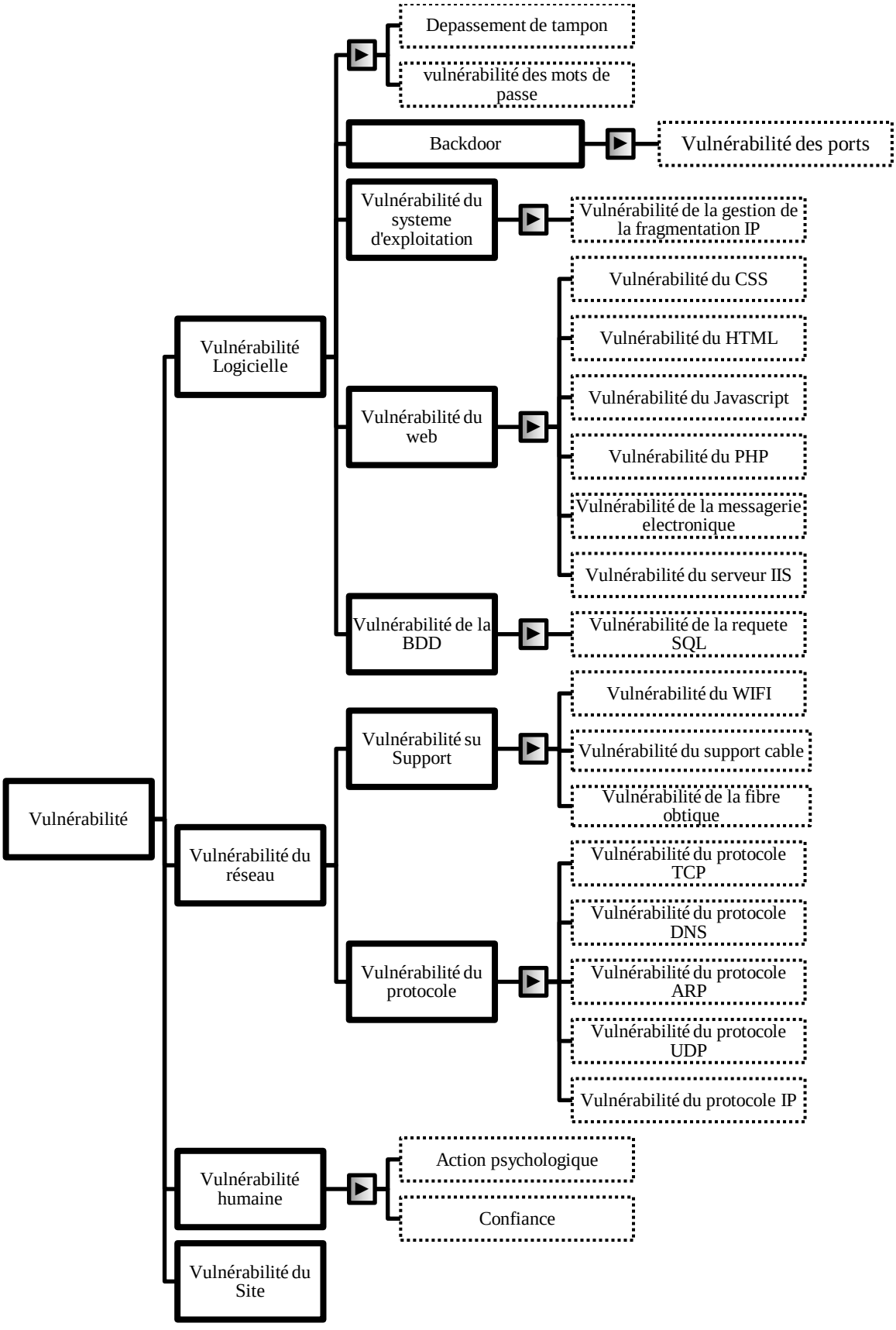


Figure 3.4 : Digramme de classification des concepts, Hiérarchie des Vulnérabilités

4. La hiérarchie du concept Conséquence

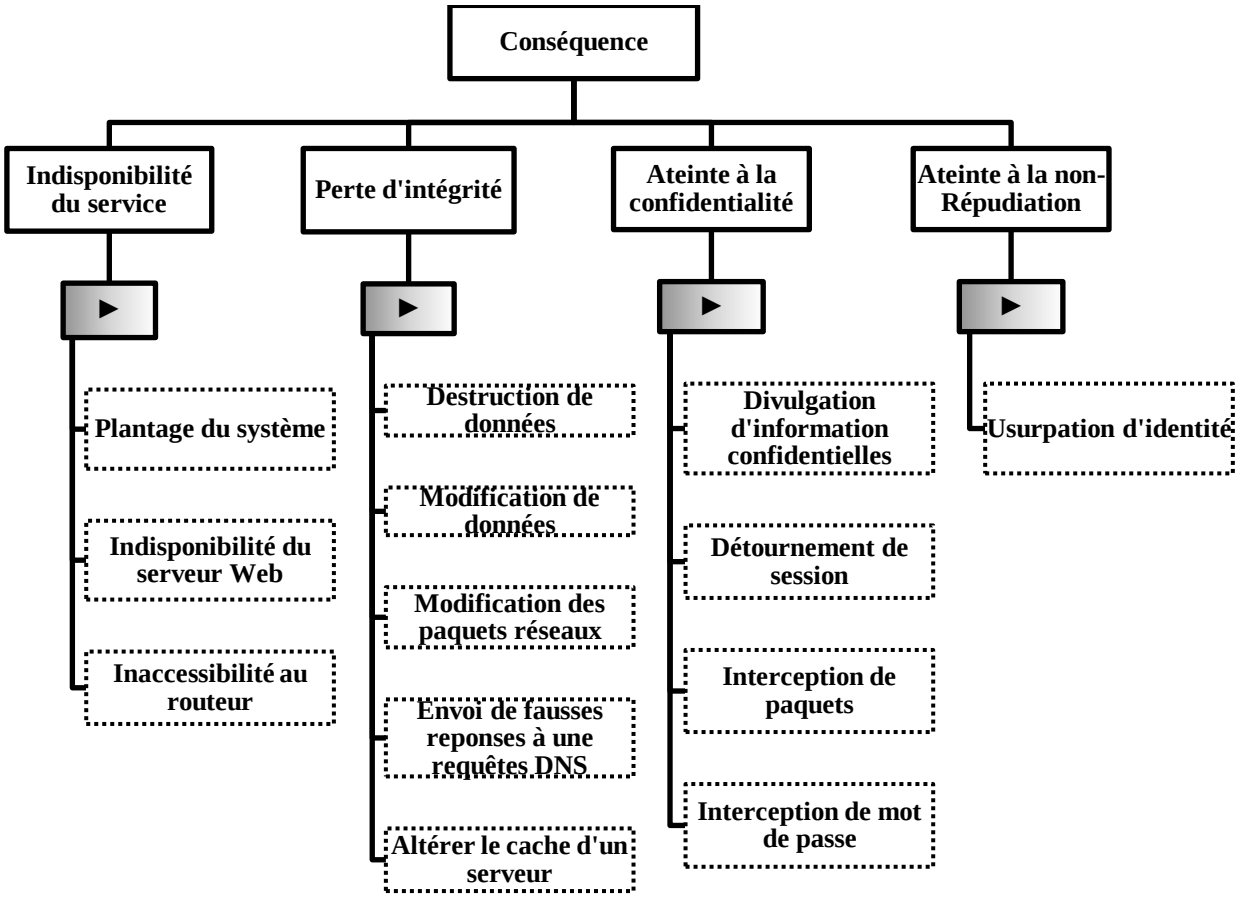


Figure 3.5 : Digramme de classification des concepts, Hiérarchie des Conséquences

5. La hiérarchie du concept Attaquant

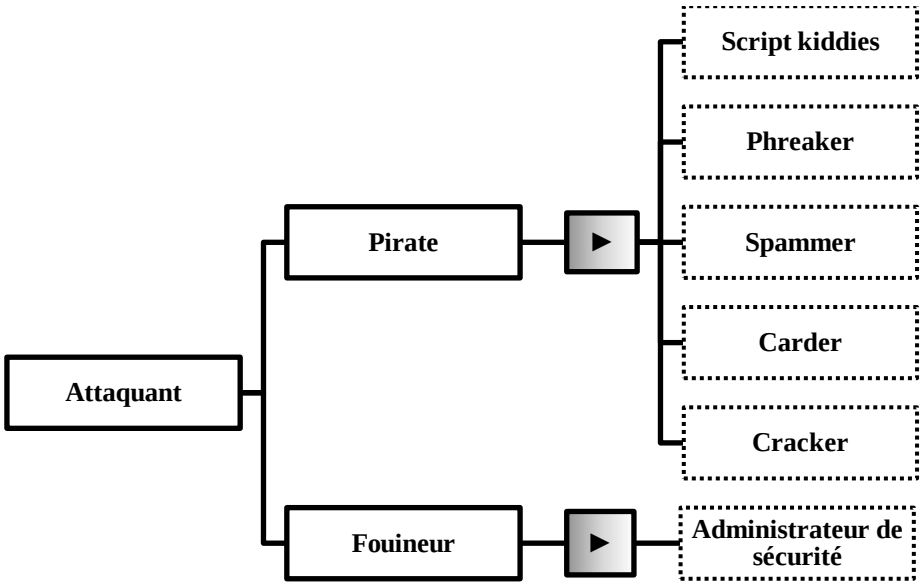


Figure 3.6 : Digramme de classification des concepts, Hiérarchie des Attaquant

6. La hiérarchie du concept Outil d'attaque

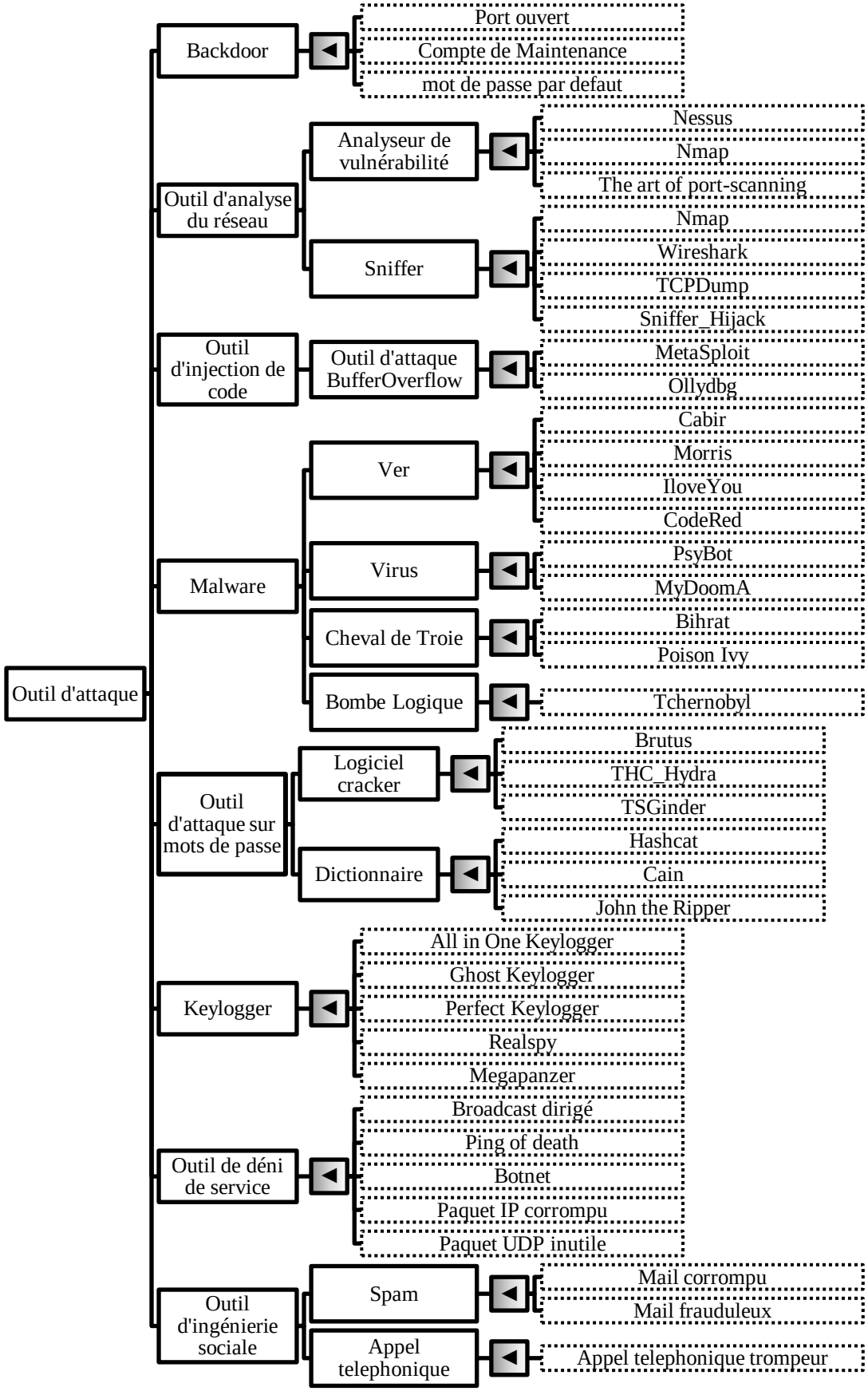


Figure 3.7 : Digramme de classification des concepts, Hiérarchie des Outils d'attaques

7. La hiérarchie du concept Contre-Mesure

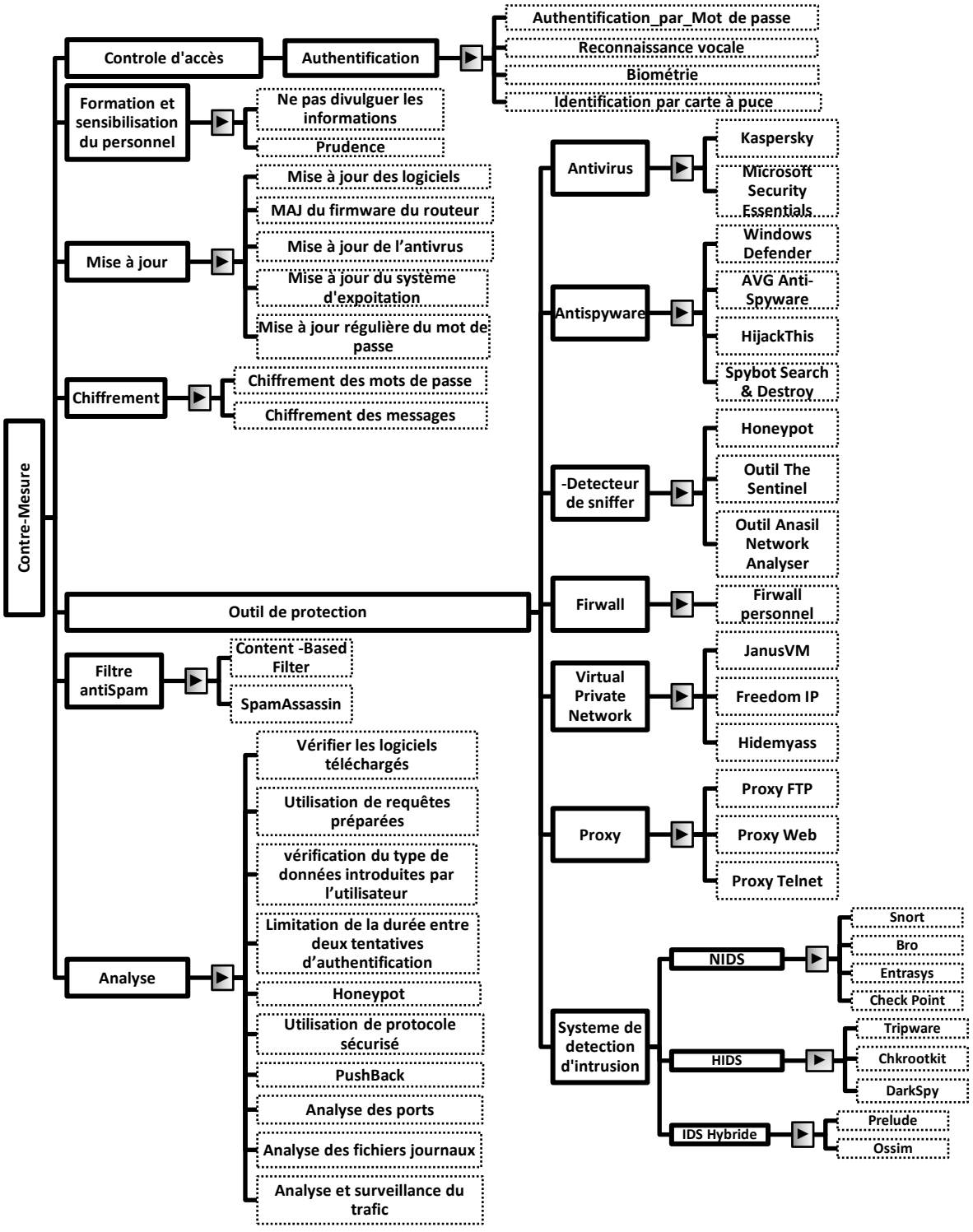


Figure 3.8 : Digramme de classification des concepts, Hiérarchie des Contre-mesures

Dictionnaire de concepts

Concept	Instances	Attributs de classe	Relations
Actif (Asset)	-	Categorie = Actif Type_Actif	Presente.Vulnerabilite Subi.Consequence
Actif Primaire	-	Type_Actif = Primaire	-
Information	- Base de données - Mot de passe - Identifiant - Information personnelle	Type=Information Niveau_confidentialité	Presente.Vulnerabilite_De_La_BDD Presente.Vulnerabilite_des_mots_de_passe
Processus	- Processus métier	Type=Processus Niveau_Intégrité	-
Actif Secondaire	-	Type_Actif = Seconaire	-
Matériel	- Routeur - Ordinateur - Switch	Type = Matériel Nom_du_constructeur Type_matériel Fonction	Presente.Vulnerabilite_materielle
Logiciel	- Programme informatique	-Fabricant -Version	Presente.Vulnerabilite_logicielle
Système d'exploitation	- Linux - Windows Seven - Windows Vista - Windows 98 - Windows XP - Système d'exploitation X	-Fonction=Système d'exploitation	Presente.Vulnerabilite_du_Systeme_d_exploitation
Système de gestion de base de données	- SGBD SQL	-Fonction=SGBD -Langage de requete -Type BDD	Presente.Vulnerabilite_de_la_BDD
Logiciel Web	- Serveur Web - Messagerie - Site Web - Application Web	-	Presente.Vulnerabilite_du_Web
Réseau	- Réseau Local - Réseau Distant	-Support : [Wifi, Paire torsadée Ethernet, Fibre optique]	Presente.Vulnerabilite_Du_reseau
Protocole	-	-Nom -Type -Fonction	-
Protocole Réseau	- TCP - UDP - DNS - IP - ARP	-Niveau_Sécurité -Numero_Port	Presente.Vulnerabilite_Protocole
Hote	- Ordinateur	Numero_Hote Fonction	Presente.Vulnerabilite_Materille Presente.Vulnerabilite_Logicielle
Personnel	- Employe_mécontent - Utilisateur	Identifiant	Presente.Vulnerabilite_Humaine
Site	-	-Type=Site -Localisation	Presente.Vulnerabilite_Du_Site
Attaque	-	Categorie = Attaque -But -Source :internet/ Externe -Type -Mode : active/passive	Cible.Actif Engendre.Consequence Est_Realisé_Avec.Outil_d_Attaque
Attaque Applicative	-	-	Cible.Logiciel Exploite.Vulnerabilite_Logicielle
Attaque par Buffer_ OverFlow	- Attaque par Buffer Overflow	-	
Exploitation d'une porte dérobée	- Backdoor	-	Est_Realisé_Avec.Outil_BufferOverflow
Injection de code	- Injection_SQL - XSS	-	Est_Realisé_Avec.Outil_Injection_de_code

Attaque_Par_Déni_De_Service	- TCP_SYN_Flooding - UDP_Flooding - DoS_Par_Fragmentation - Broadcast_Dirigé - Dos_Distribué - Attaque_par_Dictionnaire - Attaque_Par_Force_Brute	-	Est_Realisé_Avec_ Outil_Deni_de_service Engendre.Indisponibilite
Attaque_Virale	-	-	Est_Realisé_Avec.Malware
Infection_par_Virus	- Infection_par_MyDoom.A - Infection_par_Psybot	-	Est_Realisé_Avec.Virus
Infection_par_Ver	- Infection_par_Cabir - Infection_par_Morris - Infection_par_Iloveyou - Infection_par_CodeRed	-	Est_Realisé_Avec.Ver
Infection_par_Cheval_de_Troie	- Infection_par_Birhat	-	Est_Realisé_Avec.Cheval_de_Troie
Infection_par_Bombe_logique	- Infection_par_Tchernobyl	-	Est_Realisé_Avec.Bombe_logique
Attaque_par_Ingénierie_Sociale	- Phishing - Pharming - Scam - Vishing	-	Est_Realisé_Avec.Outil_integierie_Sociale Exploite.Vulnerabilite_Humaine
Spoofing	-	-	Exploite.Vulnerabilite_Protocole
IP_Spoofing	- NBS - BS	-	-
DNS Spoofing	- DNS_IP_Spoofing - DNS_Cache_Poisonning	-	-
ARP_Spoofing	- ARP_Spoofing	-	-
Espionnage_Informatique	-	-	-
Installation_de_Keylogger	- Envoi_de_Keylogger		Est_Realisé_Avec.Keylogger
Scanning	- Snnifing - Balayage_de_ports		Est_Realisé_Avec.Outil_d_analyse
Attaque_Man_In_The_Middle	- Attaque_du_protocole_ARP - Attaque_par_rejeu - Détournement_de_Session_TCP	-	-
Contre_mesure	-	-	Protégé.Actif
Controle_d_acces	-	-	
Authentification	- Mots_de_passe - Reconnaissance_Vocale - Biométrie - Identification_par_carte_a_puce	-	-
Formation_et_sensibilisation_du_personnel	- Ne_pas_divulguer_les_informations - Prudence	Acquisition= parfaite/imparfaite	Protege.Personnel
Mise_a_jour	- Correction_des_failles - Mise_a_jour_des_logiciels - Mise_a_jour_du_firme_du_ routeur - Mise_a_jour_de_l_antivirus - Mise_a_jour_du_systeme_d_ exploitation - Mise_a_jour_reguliere_du_mot_ de_passe	-	Reduit.Vulnerabilite
Outil_de_protection	-	-Type= Outil de protection -Fonction -Licence (Booleen) -Fabricant -Version	Protege.Logiciel
Antivirus	- Kaspersky - Microsof_Security_Essentials - Avast	-Constructeur -Version	Reduit.Attaque_Virale Reduit.Attaque_par_mot_de_passe
AntiSpyware	- Windows_Defender	-Constructeur	Detecte.Malware

	- AVG_Anti_Spyware - HijackThis - Spybot_Search_and_Destroy	-Version	
Detecteur_de_sniffer	- The_sebtinel - Anasil_Network_Analyser	-	Protege.Reseau
Parefeu		Support utilisé= routeur/switch	-
VPN	- JanusVM - Freedom_IP - Hidmyass	-Constructeur -Version	-
Proxy	- Proxy_FTP - Proxy_Web - Proxy_Telnet	-	-
Système_de_detection_d_intrusion	-	-	-Previet_de.Attttaque_par_deni_de_service -Prevent_de.Attaque_Man_in_the_middle
Filtre_AntiSpam	- Content_Based_Filter - SpamAssassin	-	-
Chiffrement	- Chiffrer_les_mots_de_passe - Chiffrer_les_messages	-Alogorithme utilisé -type symetrique/asymétrique	-
Analyse	- PushBack - Analyse_des_ports - Analyse_des_Vulnerabilités - Analyse_et_surveillance_du_trafic - Utilisation_de_protocole_de_controle_de_transmission - Analyse_des_vulnérabilités_des_protocoles - Verifier les logiciels telecharges - Utilisation de requetes preparees - Verification_du_type_de_donnees_introduit_par_l_utilisateur - Honeypot	-	-Reduit.Vulnerabilite_du_protocole -Reduit.Vulnerabilite_du_support
Attaquant	-	intention :mauvaise/ bonne	Effectue.Attaque
Fouineur	-	-intention= bonne	Effectue.Attaque
Pirate	-Phreaker -Spammer -Carder -Cracker -Hacker	-intention= mauvaise	Effectue.Attaque
Consequence	-	Categorie = Consequence -Objectif_visé (Service_Corrompu) =Confidentialité	Est_engendre_par.Attaque
Atteinte_a_la_confidentialité	- Divulgation_d_information_confidentielles -Detournement_de_session -Intercetion_de_paquets -Interception_de_mot_de_passe	-Objectif_visé (Service_Corrompu) =Confidentialité	-
Indisponibilité_du_Service	-Plantage_du_systeme -Indidponibilite_du_serveur_Web -Inaccessibilite_au_routeur	-Objectif_visé (Service_Corrompu) =Disponibilité	-
Personne	-	-Nom -Prenom -Adresse -Date_de_naissance	-
Outil_d_attaque	-	Type	-
Backdoor	- Port_Ouvert - Compte_de_Maintenance - Mot_de_passe_par_Defaut	Type= attaque applicative	-
Outil_Analyse_Reseau	-	Type= attaque applicative	-

Outil_Injection_Code	-	Type= attaque applicative	Utilise_pour.Injection_de_code
Malware	-Virus -Ver -Bombe logique -Cheval de Troie	-	- Utilise_pour_Infection_virale - Utilise_pour_Infection_par_ver - Utilise_pour_Infection_Bombe_logique - Utilise_pour_Infection_Cheval_de_Troie
Outil_attaque_sur_mot_de_passe	-	Type= attaque_sur_mot_de passe	-
Keylogger	-All_in_One_Keylogger -Ghost_Keylogger -Perfect_Keylogger	Type= attaque_par_ecoute	-
Outil_Deni_de_Service	-Broadcast_dirigé -Ping_of_death -Botnet -Paquet_IP_corrompu -Paquet_UDP_inutile	Type= Dos	-
Outil_Ingénierie_Sociale	-	Type= attaque_pa_ingenierie_sociale	-
Ver	-Cabir -Code-Red -Morris -I-Love-You	-	-
Virus	-MyDoom.A -Psybot	-	-
Cheval_de_Troie	-Birhat -Poison Ivi	-	-
Bombe_Logique	Tchernobyl	-	-
Logiciel_Cracker	-Brutus -THC_Hydra -TSGinder	-	-
Vulnérabilité	-	-	-
Spam	-	-	-
Appel_Telephonique	-	-	-
Vulnérabilité_Humaine	- Action_psychologique - Confiance	-	-
Vulnérabilité_du_Web	- Vulnerabilité_du_HTML - Vulnerabilité_du_CSS - Vulnerabilité_du_PHP - Vulnerabilité_du_Javascript - Vulnerabilité_de_la_Messagerie - Vulnerabilité_du_Serveur_IIS	-	-
Vulnerabilité_de_la_BDD	- Vulnérabilité_de_la_requete_SQL	-	-
Vulnérabilité_Du_Systeme_d_Exploitation	- Vulnérabilité_de_la_Gestion_de_la_Fragmentation_IP	-	-
Vulnérabilité_Du_Support	- Vulnérabilité_Reseau_Sans_fil - Vulnérabilité_Reseau_Ethernet	-	-
Vulnérabilité_Du_Protocole	- Vulnérabilité_du_Protocole_TCP - Vulnérabilité_du_Protocole_IP - Vulnérabilité_du_Protocole_UDP - Vulnérabilité_du_Protocole_DNS - Vulnérabilité_du_Protocole_ARP	-	-

Tableau 3.7 : Dictionnaire des concepts

Construction du diagramme de relations binaires

Une fois que les concepts et leurs hiérarchies définies, il est nécessaire de relier tous les concepts qui ont des liens entre eux en utilisant les relations sémantiques existantes. Les relations qui existent entre nos concepts peuvent être définies de la manière suivante :

- Une attaque est réalisée par un attaquant.
- Une attaque cible un actif.
- L'attaque peut exploiter la vulnérabilité d'un actif pour l'attaquer ou attaquer un autre actif.
- Pour effectuer cette attaque, un attaquant utilise un outil d'attaque contre l'actif visé.
- Une attaque peut avoir des conséquences fâcheuses et qui compromettent la sécurité en violant l'un de ses objectifs.
- Plusieurs contre-mesures peuvent être mises en place afin de contrer ces attaques, protéger l'actif et réduire les vulnérabilités.

Nous allons reprendre ces relations sémantiques plus en détails dans la Figure 3.9.

Diagramme des relations binaires :

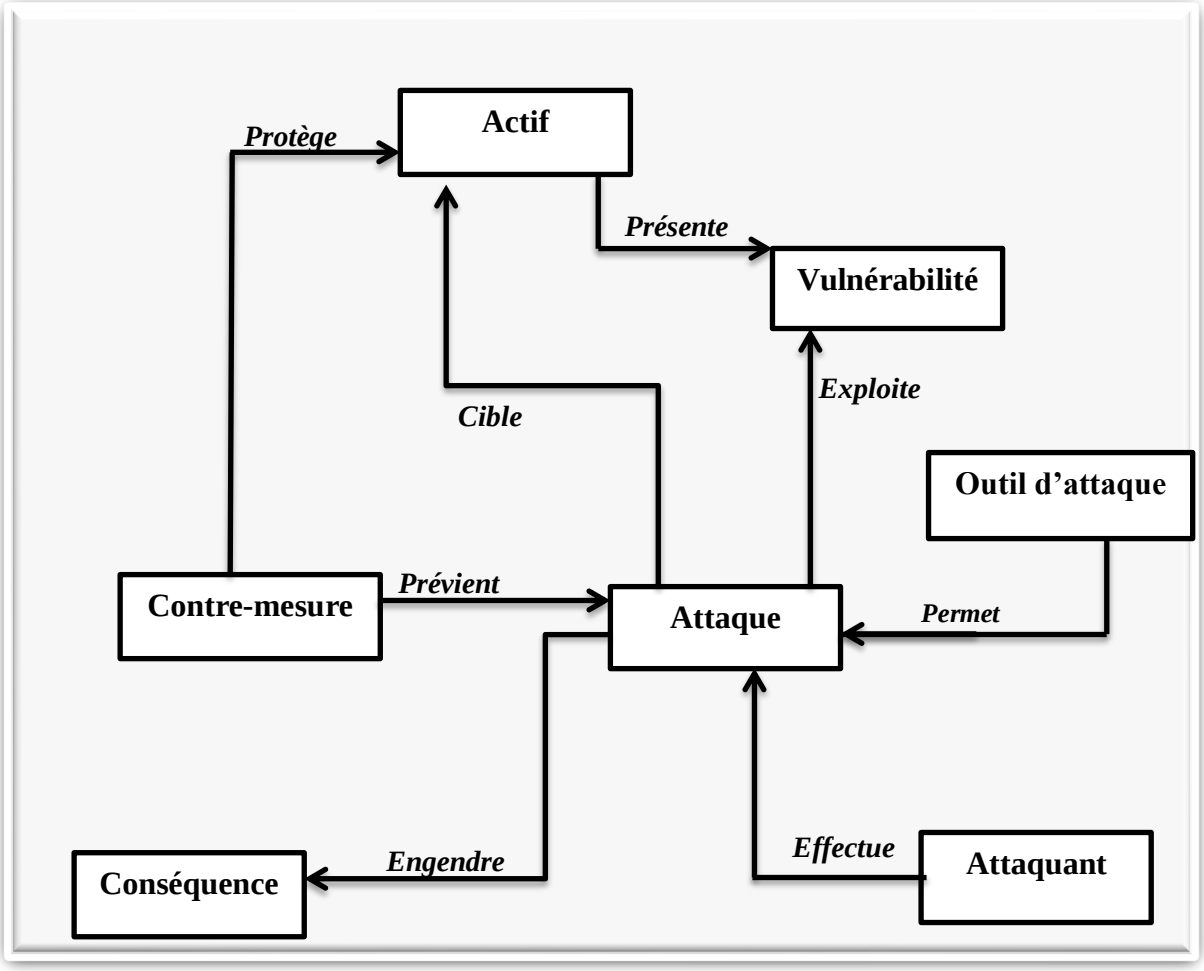


Figure 3.9 : Digramme de classification des concepts et des relations binaires

Tableau des relations binaires

Nom de la relation	Concept Source	Cardinalité Source (Max)	Concept Cible	Relation Inverse
Cible	Attaque	N	Actif	*Est_Ciblé_Par*
Effectue	Attaquant	N	Attaque	*Est_Effectué_Par*
Engendre	Menace	N	Conséquence	*Est_Engendré_Par*
Exploite	Attaque	N	Vulnérabilité	*Est_Exploitée_Par*
Est_Ciblé_Par	Actif	N	Attaque	*Cible*
Est_Effectué_Par	Attaque	1	Attaquant	*Effectue*
Est_Engendré_Par	Conséquence	N	Menace	*Engendre*
Est_Exploitée_Par	Vulnérabilité	N	Attaque	*Exploite*
Est_Présenté_Par	Vulnérabilité	1	Actif	*Présente*
Est_Prévenu_Par	Attaque	N	Contre_Mesure	*Prévient*
Est_Protégé_Par	Actif	N	Contre_Mesure	*Protège*
Est_Permis_Par	Attaque	N	Outil_d_Attaque	*Permet*
Présente	Actif	N	Vulnérabilité	*Est_Présenté_Par*
Prévient	Contre_Mesure	N	Attaque	*Est_Prévenu_Par*
Protège	Contre_Mesure	N	Actif	*Est_Protégé_Par*
Permet	Outil_d_Attaque	N	Attaque	*Est_Permis_Par*

Tableau 3.8 : Tableau des relations binaires.

Tableau des attributs

Nom d'attribut d'instance	Nom de concept	Type de valeur	Intervale de valeur
Categorie	-Actif	String	-
Type_actif	-Actif_primaire -Actif_secondaire	String	-
Type	Information	String	-
Type	Processus	String	-
-Type -Nom_du_ constructeur -Fonction	Materiel	String	-
-Fonction -Fabricant -Version	Logiciel	String	-
Fonction	Système_d_exploitation	String	-
-Fonction -Langage_de_requete -Type_BDD	Système_de_gestion_de_base_de_ donnees	String	-
-Support	Reseau	String	-
-Nom -Type -Fonction	Protocole	String	-
-Niveau_de_securite -Numero_Port	Protocole_Reseau	String	-
-Numero_Hote -Numero_Port	Hote	String	-
-Identifiant	Personnel	Entier	>0
-Type -Localisation	Site	String	-
-Categorie -But -Source -Type -Mode	Attaque	String	-
-Acquisition	Formation_et_sensibilisation_du_personnel	String	-
-Type -Licence	Outil_de_protection	String	-

-Fabricant -Version			
-Constructeur -Version	Antivirus	-string -entier	- -> 0
-Constructeur -Version	Antispyware	-string -entier	- -> 0
Support	Parefeu	String	-
-Constructeur -Version	VPN	-string -entier	- -> 0
- Algorithme_ utilisee -Type	Chiffrement	String	-
Intension	Attaquant	String	-
Categorie Objectif	Consequence	String	-
Objectif_vise	Atteinte_a_la_confidentialite	String	-
Objectif_vise	Indisponibilite_du_service	String	-
-Nom -Prenom -Adresse -Date_de_naissance	Personne	enregistrement	-
Type	Outil_d_attaque	String	-

Tableau : 3.9 : Tableau des attributs

Tableau des instances (Extrait des instances de notre ontologie)

Nom de l'instance	Nom du concept	Attributs	
		Nom	Valeur
Application_Web	Logiciel_Web	-	-
Information_confidentielle	Information	Niveau_confidentialité	'Elevé'
Mot_de_passe_Utilisateur	Information	-	-
Ordinateur	Hote	-	-
Programme_informatique	Logiciel	-	-
Windows_Seven	Systeme_d_exploitation	Constructeur Version	= 'Windows' = '7'
Messagerie	Logiciel_Web	-	-
Réseau Local	Reseau	Support	='Ethernet'
TCP	Protocole	-	-
IP	Protocole	-	-
Utilisateur_253	Personnel	Identifiant	='253'
Depassement_de_tas	Attaque_par_Buffer_Overflow	-	-
Injection_SQL	Injection_de_code	-	-
Deni_de_service_Distribué	Attaque_par_deni_de_service	-	-
Attaque_par_Dictionnaire	Attaque_sur_mot_de_passe	-	-
Phishing	Attaque_par_ingenierie_sociale	-	-
DNS_Cache_Poisonning	DNS_Spoofing	-	-
Sniffing	Ecoute_du_reseau	-	-
Attaque_du_protocole_ARP	Attaque_Man_in_the_middle	-	-
Exploitation_d_un_port_ouvert	Exploitation_d_une_porte_derobee	-	-
Port_Ouvert	Outil_pour_une_porte_derobee	-	-
Nessus	Analyseur_de_ports		-
Nmap	Sniffer		-
MetaSploit	Outil_d_attaque_par_Buffer_ Overflow	-	-
Code_Red	Ver	-	-
Psybot	Virus	-	-
Birhat	Cheval_de_Troie	-	-
Tchernobyl	Bombe_logique	-	-
Brutus	Logiciel_Cracker	-	-
Cain	Dictionnaire_mots_de_passe	-	-
Ghost_Keylogger	Keylogger	-	-
Botnet	Outil_de_deni_de_service	-	-
Mail_Corrompu	Spam	-	-
Spammer	Pirate	Intention	='Mauvaise'
Cracker	Pirate	Intention	='Mauvaise'

Carder	Pirate	Intention	=’Mauvaise’
Hacker	Pirate	Intention	=’Mauvaise’
Administrateur_de_securite	Fouineur	Intention	=’Bonne’
Indisponibilite_du_serveur_Web	Indisponibilite_du_service	-	-
Modification_de_donnees	Perte_d_integrite	-	-
Divulgation_d_information_confidentielle	Atteinte_a_la_confidentialite	-	-
Interception_de_paquets	Atteinte_a_la_confidentialite	-	-
Usurpation_d_identite	Atteinte_a_la_non_repudiation	-	-
Mise_a_jour_reguliere_du_mot_de_Passe	Mise_a_jour	-	-
Prudence	Formation_du_personnel	-	-
Authentification_par_mot_de_passe	Authentification	-	-
Kaspersky	Antivirus	Constructeur Version	=’Kaspersky’ =’2013’
Windows_Defender	Antispyware	Constructeur Version	=’Windows’ =’2008’
Honeypot	Detecteur_de_sniffer	-	-
Snort	NIDS	-	-
AIDE	HIDS	-	-
Analyse_de_fichier_journaux	Analyse_et_Test	-	-
JanusVM	VPN	-	-
Port_Ouvert	Vulnerabilite_porte_derobee	-	-
Mot_de_passe_Cassable	Vulnerabilite_Mot_de_passe	-	-
Depassement_de_Tampon	Depassement_de_Ressource	-	-
Action_Psychologique	Vulnerabilite_humaine	-	-
Vulnerabilité_du_CSS	Vulnerabilite_du_Web	-	-
Vulnerabilité_du_PHP	Vulnerabilite_du_Web	-	-
Vulnérabilité_du_protocole_TCP	Vulnerabilite_du_Protocole_reseau	-	-
Vulnérabilité_du_protocole_IP	Vulnerabilite_du_Protocole_reseau	-	-
Vulnérabilité_de_la_requete_SQL	Vulnerabilite_liee_a_la_BDD	-	-
Vulnérabilité_de_la_Gestion_de_la_Fragmentation_IP	Vulnerabilite_du_systeme_d_Exploitation	-	-

Tableau : 3.10 : Tableau des instances

Etape 3 : Formalisation

Dans cette étape, nous allons utiliser le formalisme des logiques de description afin de formaliser le modèle conceptuel que nous avons obtenue dans l’étape précédente de conceptualisation.

➔ **La construction de la TBox se fait :**

- Par la définition des concepts et des rôles en utilisant les constructeurs fournis par les logiques de descriptions. Par exemple, la définition d’un attaquant « Un attaquant est une personne qui effectue au moins une attaque » peut être écrite en logique de description comme :

$$\textbf{Attaquant} \equiv \exists \textbf{Effectue.X} \sqcap \textbf{Attaque(X)}$$

- Par la spécification des relations de subsomption qui existent entre les différents concepts/rôles.

$$\textbf{Pirate} \sqsubseteq \textbf{Attaquant}$$

Les définitions des différents concepts sont illustrées dans le tableau ci-dessous.

Concept	Définition	Relation de Subsomption
Actif	-	Actif ⊑ Thing
Actif Primaire	≡ (Type_Actif=Primaire)	Actif_Primaire ⊑ Actif
Information	-	Information ⊑ Actif_Primaire
Processus	-	Processus ⊑ Actif_Primaire
Actif Secondaire	≡ (Type_Actif=Primaire)	Actif_Secondaire ⊑ Actif
Matériel	-	Matériel ⊑ Actif_Secondaire
Logiciel	-	Logiciel ⊑ Actif_Secondaire
Système_d_exploitation	-	Système_d_exploitation ⊑ Logiciel
SGBD	-	SGBD ⊑ Logiciel
Logiciel Web	-	Logiciel Web ⊑ Logiciel
Réseau	-	Réseau ⊑ Actif_Secondaire
Protocole	-	Protocole ⊑ Thing
Protocole Réseau	-	Protocole_Réseau ⊑ Protocole
Hote	-	Hote ⊑ Matériel
Personnel	-	Personnel ⊑ Actif_Secondaire
Employé	-	Employé ⊑ Personnel
Site	-	Site ⊑ Actif_Secondaire
Attaque	-	Attaque ⊑ Thing
Attaque_Applicative	≡(∀**Exploite.**Vulnerabilité_Logicielle ⊔ ∀**Exploite.**Vulnerabilité_du_Reseau) ⊓ ∀**Cible.**Logiciel	Attaque_Applicative ⊑ Attaque
Attaque_par_Buffer_OverFlow	≡(∀**Engendre.**Indisponibilite ⊔ ∀**Engendre.**Atteinte_a_l_integrité) ⊓ ∀**Se_Fait_Avec.**Outil_BufferOverflow	Attaque_par_Buffer_OverFlow ⊑ Attaque_Applicative
Exploitation_d_une_porte_dérobée	-	Exploitation_d_une_porte_dérobée ⊑ Attaque_Applicative
Injection_de_code	≡(∀**Engendre.**Indisponibilite ⊔ ∀**Engendre.**Atteinte_a_la_confidentialité ⊔ ∀**Engendre.**Atteinte_a_l_integrité)	Injection_de_code ⊑ Attaque_Applicative
Attaque_Par_Déni_De_Service	≡ ∀**Engendre.**Indisponibilite	Attaque_Par_Déni_De_Service ⊑ Attaque
Attaque_Virale	≡ ∀**Se_Fait_Avec.**Malware	Attaque_Virale ⊑ Attaque
Infection_par_Virus	≡ ∀**Se_Fait_Avec.**Virus	Infection_par_Virus ⊑ Attaque_Virale
Infection_par_Ver	≡ ∀**Se_Fait_Avec.**Ver	Infection_par_Ver ⊑ Attaque_Virale
Infection_par_Cheval_de_Troie	≡ ∀**Se_Fait_Avec.**Cheval_de_troie	Infection_par_Cheval_de_Troie ⊑ Attaque_Virale
Infection_par_Bombe_logique	≡ ∀**Se_Fait_Avec.**Bombe_Logique	Infection_par_Bombe_logique ⊑ Attaque_Virale
Attaque_par_Ingénierie_Sociale	≡ ∀**Se_Fait_Avec.**Outil_ingenierie_sociale ⊓ ∀**Exploite.**Vulnerabilité_Humaine	Attaque_par_Ingénierie_Sociale⊑ Attaque
Spoofing	≡ ∀**Exploite.**Vulnerabilité_Protocole	Spoofing ⊑ Attaque
IP_Spoofing	≡ (Vulnerabilité_Protocole=IP)	IP_Spoofing ⊑ Spoofing
DNS Spoofing	≡ (Vulnerabilité_Protocole=DNS)	DNS Spoofing ⊑ Spoofing
ARP_Spoofing	≡ (Vulnerabilité_Protocole=ARP)	ARP_Spoofing ⊑ Spoofing
Espionnage_Informatique	≡ ∀**Engendre.**Atteinte_a_la_confidentialité	Espionnage_Informatique ⊑ Attaque
Installation_de_Keylogger	≡ ∀**Est_Permis_Par.**Keylogger	Installation_de_Keylogger ⊑ Espionnage_Informatique
Scanning	≡ ∀ **Est_Permis_Par.** Outil_analyse_du_reseau	Scanning ⊑ Espionnage_Informatique
Attaque_Man_In_The_Middle	-	Attaque_Man_In_The_Middle ⊑ Attaque
Contre_mesure	-	Contre_mesure ⊑ Thing
Controle_d_acces	-	Controle_d_acces ⊑ Contre_mesure
Authentification	-	Authentification ⊑ Controle_d_acces
Formation_et_sensibilisation_du_personnel	-	Formation_et_sensibilisation_du_personnel ⊑ Contre_mesure
Mise_a_jour	-	Mise_a_jour ⊑ Contre_mesure
Outil_de_protection	-	Outil_de_protection ⊑ Contre_mesure
Antivirus	≡ ∀Previent.Infection_Virale	Antivirus ⊑ Outil_de_protection
Detecteur_de_sniffer	≡ ∀Previent.Sniffing	Detecteur_de_sniffer ⊑

		Outil_de_protection
Parefeu	-	Parefeu ⊑ Outil_de_protection
VPN	-	VPN ⊑ Outil_de_protection
Proxy	-	Proxy ⊑ Outil_de_protection
Système_de_detection_d_intrusion	-	Système_de_detection_d_intrusion ⊑ Outil_de_protection
Filtre_AtiSpam	-	Filtre_AntiSpam ⊑ Outil_de_protection
Précaution	-	Précaution ⊑ Contre_mesure
Chiffrement	-	Chiffrement ⊑ Contre_mesure
Test_et_Analyse	-	Test_et_Analyse ⊑ Contre_mesure
Attaquant	≡∃Effectue.attaque	Attaquant ⊑ Personne
Chapeau_Blanc	≡ ∀ (Intention=Bonne)	Chapeau_Blanc ⊑ Attaquant
Pirate	≡ ∀ (Intention=Mauvaise)	Chapeau_Noir ⊑ Attaquant
Consequence	-	Consequence ⊑ Thing
Atteinte_a_la_confidentialité	-	Atteinte_a_la_confidentialité ⊑ Consequence
Indisponibilité_du_Service	-	Indisponibilité_du_Service ⊑ Consequence
Perte_d_intégrité	-	Perte_d_intégrité ⊑ Consequence
Perte_de_Controle	-	Perte_de_Controle ⊑ Consequence
Saturation_des_ressources_systemes	-	Saturation_des_ressources_systemes ⊑ Consequence
Outil_d_attaque	-	Outil_d_attaque ⊑ Thing
Vulnérabilité	-	Vulnérabilité ⊑ Thing
Vulnérabilité_Humaine	-	Vulnérabilité_Humaine ⊑ Vulnérabilité
Vulnérabilité_Logicielle	-	Vulnérabilité_Logicielle ⊑ Vulnérabilité

Tableau 3.11 : Tableau des concepts formalisés

➔ **Construction de la ABox :**

Nous décrivons les faits en utilisant le langage assertionnel, de la manière suivante :

- C(A) : Pour spécifier que A est une instance de la classe C.

  Par exemple : Ver(Cabir), Virus(PsyBot).

- R(A,B) : Pour spécifier que les deux individus A et B sont reliés par la relation R.

  Par exemple : Cible(Sniffing, Données), exploite(Pirate, Faille).

Dans les tableaux III.12 et III.13, nous définissons quelques assertions :

Nom du concept	Nom de l'instance
Logiciel_Web	Logiciel_Web(Application_Web) Logiciel_Web(Messagerie)
Information	Information(Information_confidentielle)
Hote	Hote(Ordinateur) Hote(Routeur)
Logiciel	Logiciel(Programme_informatique)
Systeme_d_exploitation	Systeme_d_exploitation(Windows_7) Systeme_d_exploitation(Windows_XP) Systeme_d_exploitation(Linux)
Reseau	Reseau(Reseau_Local) Reseau(Reseau_Distant)
Protocole_Reseau	Protocole_Reseau(TCP) Protocole_Reseau(IP)
Attaque_par_Buffer_Overflow	Attaque_par_Buffer_Overflow(Depassement_de_tas) Attaque_par_Buffer_Overflow(Depassement_d_entier)
Attaque_par_ingenierie_sociale	Attaque_par_ingenierie_sociale(Phishing) Attaque_par_ingenierie_sociale(Vishing) Attaque_par_ingenierie_sociale(Pharming)
Sniffer	Sniffer(Nmap) Sniffer(Sniffer_Hijack) Sniffer(TCPDump) Sniffer(Wireshark)
Ver	Ver(Code_Red) Ver(Cabir) Ver(ILoveyou) Ver(Morris)
Virus	Virus(Psybot) Virus(MyDoomA)
Logiciel_Cracker	Logiciel_Cracker(Brutus) Logiciel_Cracker(TSGinder) Logiciel_Cracker(THC_Hydra)
Keylogger	Keylogger(Ghost_Keylogger) Keylogger(All_in_one_Keylogger) Keylogger(Megapanzer) Keylogger(Perfect_Keylogger) Keylogger(RealSpy)
Pirate	Pirate(Spammer) Pirate(Cracker) Pirate(Carder) Pirate(Hacker)

Tableau 3.12 : Extrait de la Partie Assertionnelle des concepts

Nom de la relation	Définition
Présente	Présente(mots_de_passe,mot_de_passe_cassable) Présente(SGBD_SQL,Vulnerabilite_requete_SQL) Présente(TCP,Vulnerabilite_du_protocole_TCP) Présente(Utilisateur,Action_Psychologique)
Exploite	Exploite(Vishing,Action_Psychologique) Exploite(TCP_SYN_Flooding,Vulnerabilite_du_protocole_TCP) Exploite(Injection_SQL,Vulnerabilite_requete_SQL) Exploite(Attaque_par_mot_de_passe,Mot_de_passe_cassable)
Previent_de	Previent_de(Kaspersky,Psybot) Previent_de(Outil_The_Sentinel,Sniffing)

Tableau 3.13 : Extrait de la Partie Assertionnelle des relations

Conclusion

La modélisation des informations caractérisant la sécurité informatique s'avère à l'ère actuelle très importante. Dans la mesure où une simple erreur peut provoquer une conséquence presqu'inestimable. Cela demande une connaissance du domaine et une méthodologie adéquate afin de pallier au problème d'incompréhension,

Dans ce chapitre, nous avons spécifié le but de notre travail qui consiste à développer une ontologie qui décrira le domaine de la sécurité informatique. Après la présentation de la méthodologie employée pour la construction de l'ontologie, nous avons établi une hiérarchie de chaque concept utilisé, à commencer des classes aux instances. Puis, nous avons défini les relations pouvant exister entre les classes.

Le chapitre qui suit mettra en exergue l'implémentation de notre ontologie, en présentant les outils et langages utilisés pour la construction et l'interrogation de notre ontologie.

Chapitre IV

Implémentation et Exploitation

4.1. Introduction

Dans le chapitre précèdent, nous avons obtenu une ontologie formelle en utilisant le processus de construction d'ontologies Methontology. Nous allons consacrer ce chapitre à l'implémentation de l'ontologie formelle obtenue précédemment. Puis nous testerons la consistance de cette ontologie afin de corriger les erreurs d'implémentation. Ensuite nous établirons les règles d'inférence SWRL qui permettront de déduire de nouvelles connaissances. Enfin nous effectuerons des requêtes SPARQL afin d'interroger l'ontologie construite.

4.2. Implémentation de l'ontologie :

Pour l'implémentation de l'ontologie, nous utiliserons l'éditeur Protégé 3.5 qui sera présenté en Annexe C. Cette implémentation se fait de la manière suivante :

- **Définition des Concepts Héritant de Thing et les relations qui existent entre eux.**

Ces concepts sont ceux obtenus au chapitre III et sont : Actif, Attaque, Attaquant, Outil_d_Attaque, Vulnérabilité, Contre_Mesure, Conséquence. La figure 4.1 donne un aperçu de protégé et de la définition du concept Actif.

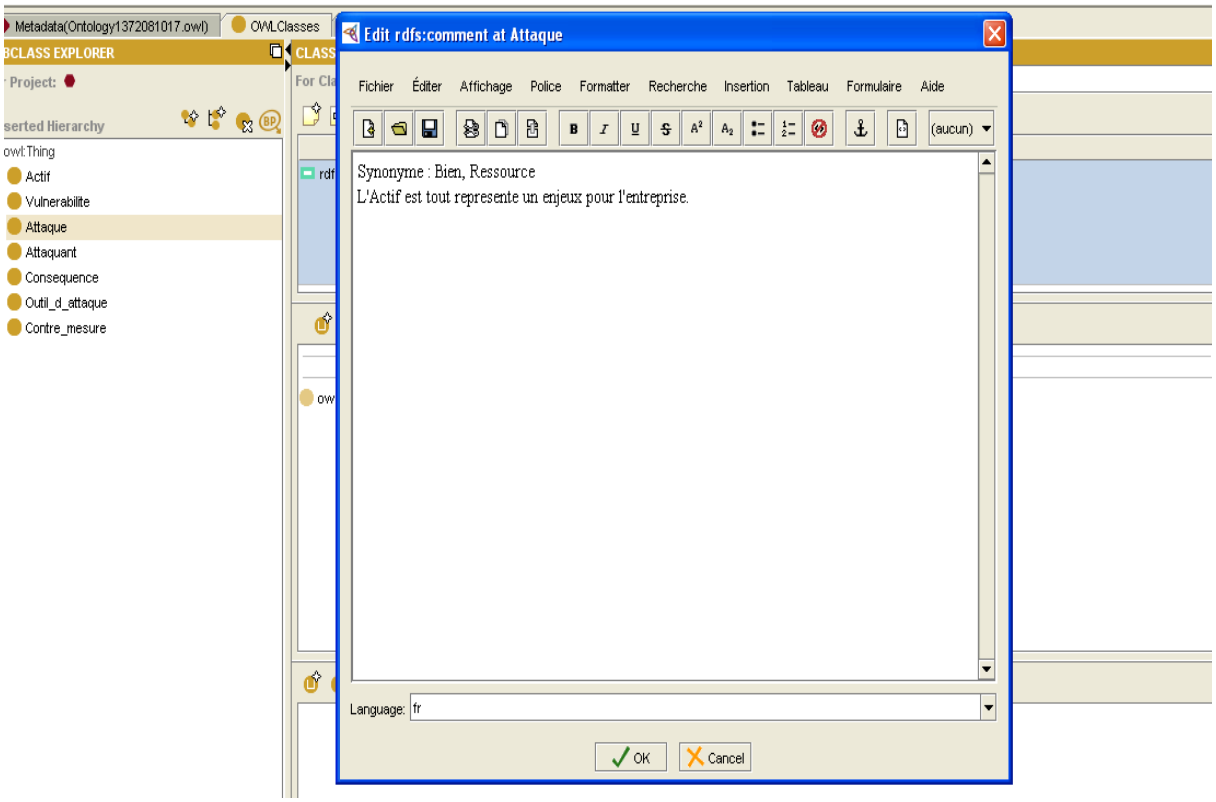


Figure 4.1. Définition du concept « Attaque »

Apres la définition de ces principaux concepts, nous passons à leurs hiérarchies. Chaque concept est relié à son concept père en utilisant la relation de subsomption « is-a ». protégé permet de définir les sous classes comme le montre la figure 4.2.

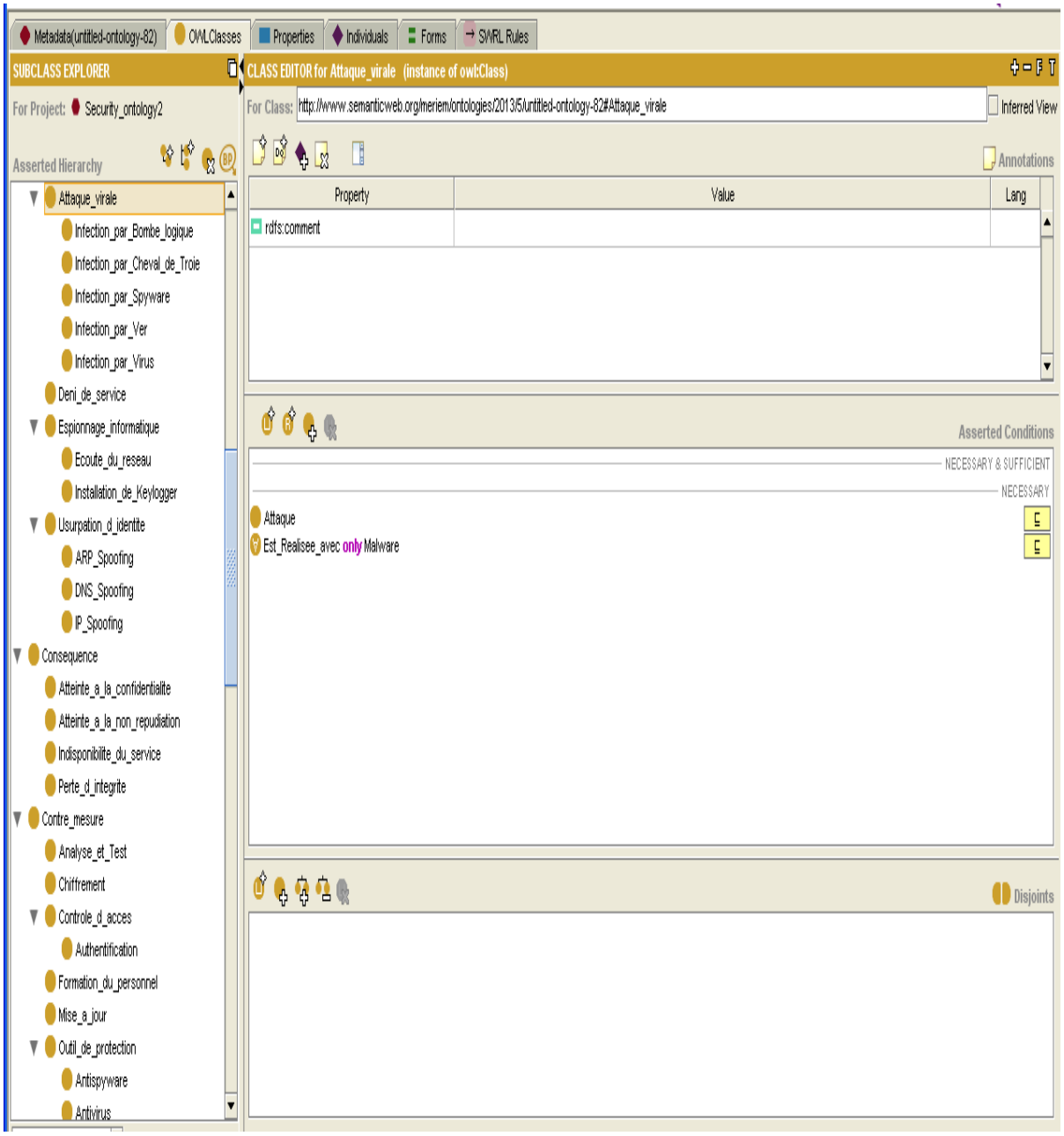


Figure 4.2. Définition des Hiérarchies des concepts de l'ontologie.

Une fois que nous avons implémenté tous les concepts de l'ontologie. Nous passons aux attributs de chaque concept. Par exemple, pour le concept attaque, nous pouvons ajouter un attribut « Type_Attaque » qui peut prendre une des deux valeurs possibles « Active –Passive », comme illustré dans la figure 4.3

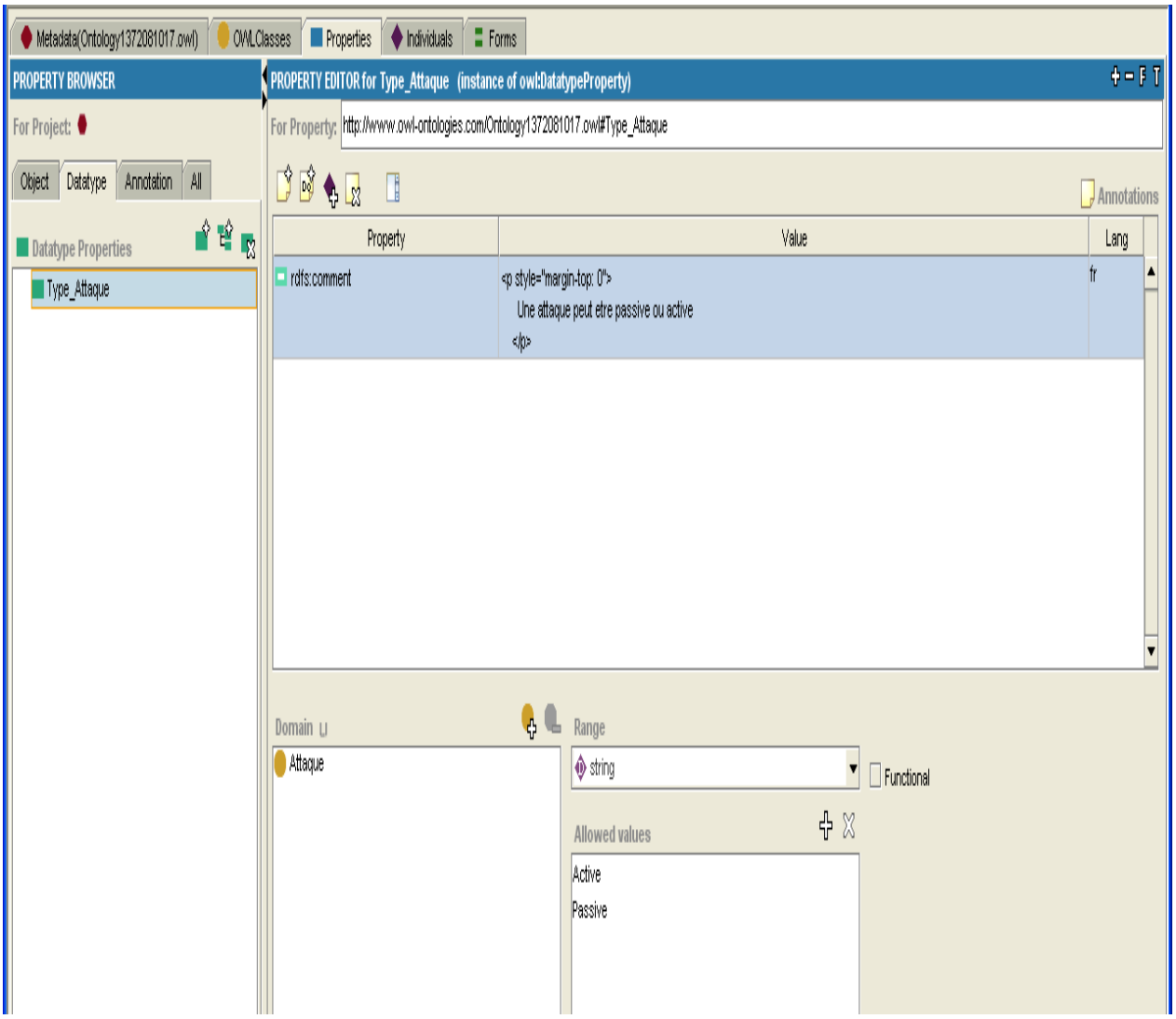


Figure 4.3. Définition de l'attribut « Type_Attaque » du concept « Attaque ».

- **Définition des relations binaires qui vont relier les concepts entre eux.**

Les relations constituent un des éléments les plus importants dans une ontologie. Nous allons à présent implémenter les relations obtenues dans la formalisation. Dans protégé, les relations peuvent être définies en tant que ObjectProperty. Leur définition se fait en spécifiant le domaine « Domain » qui contient le concept source de la relation, et le domaine « Range » qui contient le concept destination de la relation.la figure 4.4 donne un exemple de la définition de la relation Cible (Une attaque cible un actif).

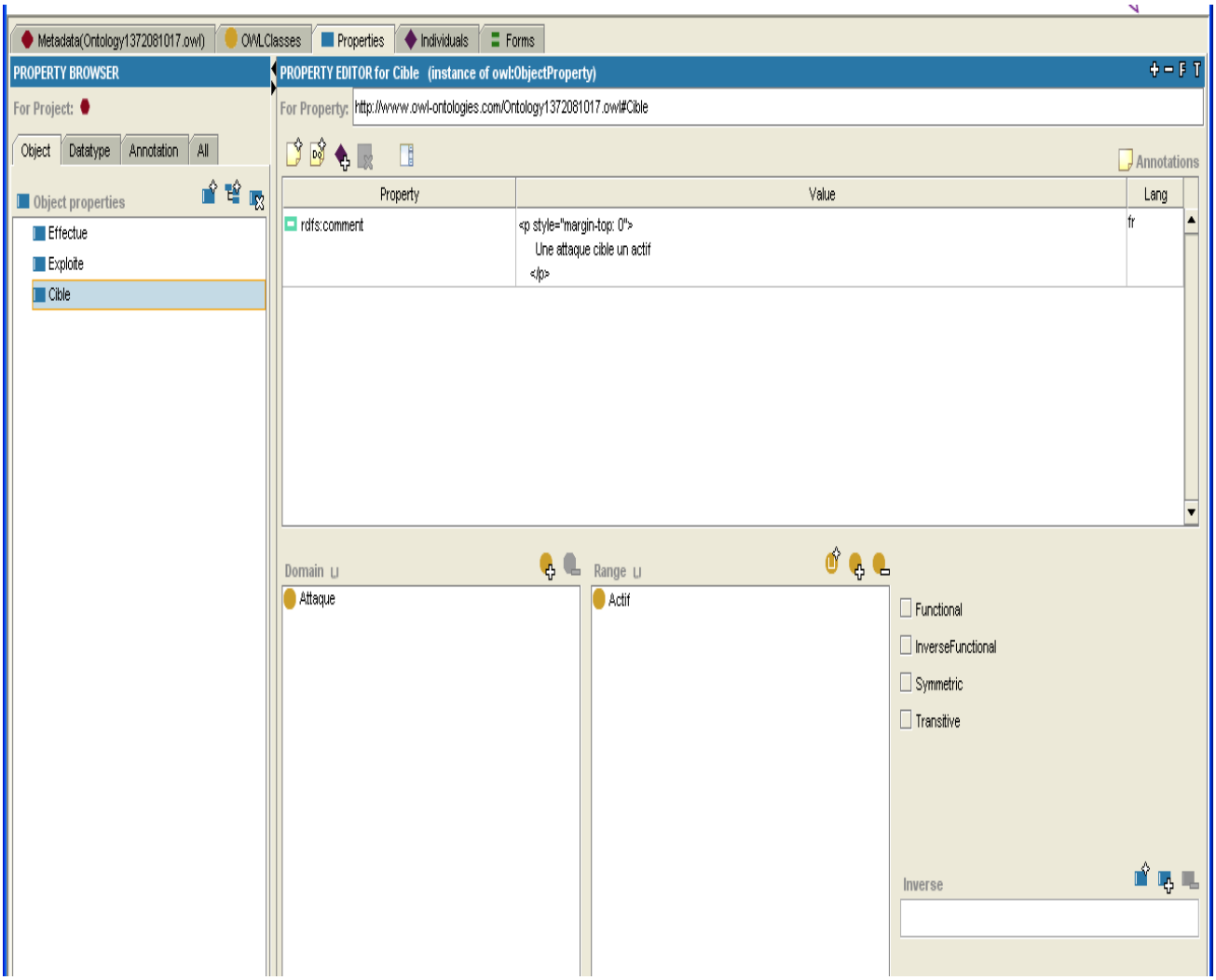


Figure 4.4. Définition de la relation « Cible » entre « Attaque » et « Actif ».

Pour avoir une ontologie consistante. Il est nécessaire d'utiliser les restrictions sur les relations reliant les concepts. Cela permet d'effectuer des raisonnements sur le niveau terminologique TBox. Comme montré dans la figure 4.5.

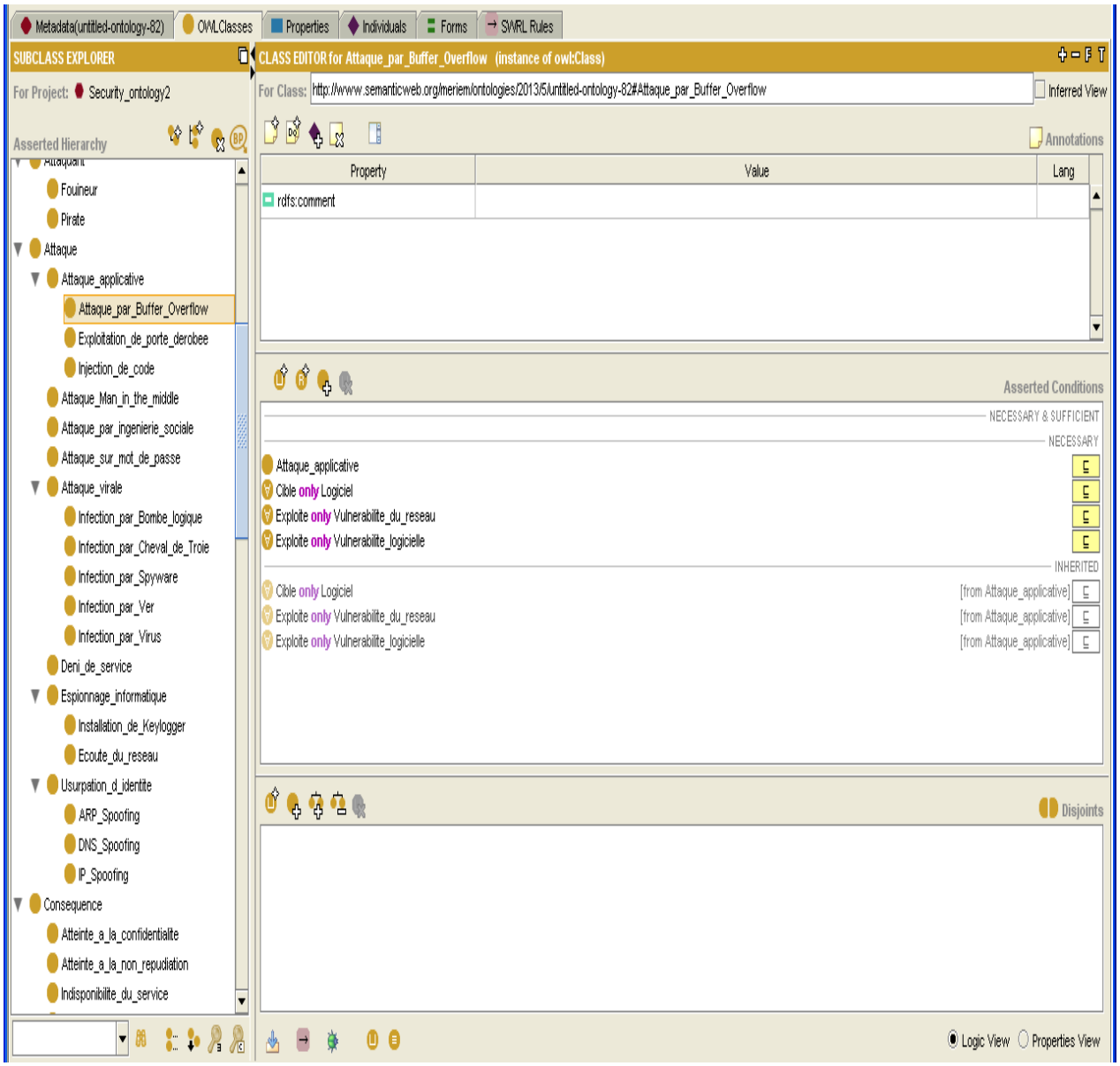


Figure 4.5. Définition des restrictions sur les relations.

Nous avons défini les concepts, les relations et les attributs. Nous donc défini la TBox de notre ontologie. Nous allons passer à la construction de la ABox qui est constitué des instances de concepts, les relations entre ces instances et les valeurs de leurs attributs.

Les instances peuvent être définies dans Protégé en utilisant l'onglet Individu, comme le montre la figure 4.6.

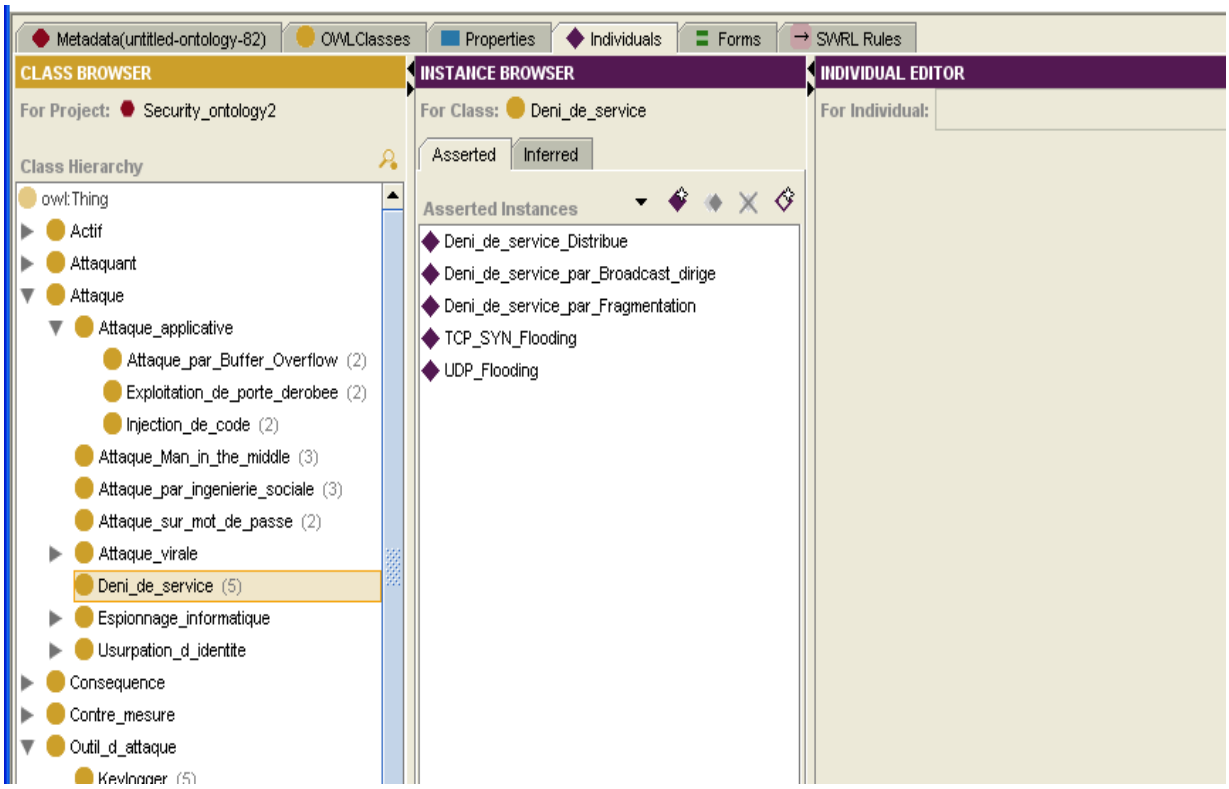


Figure 4.6. Définition des instances de concepts.

Apres la définition des instances, nous attribuons des valeurs à leurs attributs en respectant le type de données spécifié pour chaque attribut. La figure 4.7 donne un exemple d'attribut.

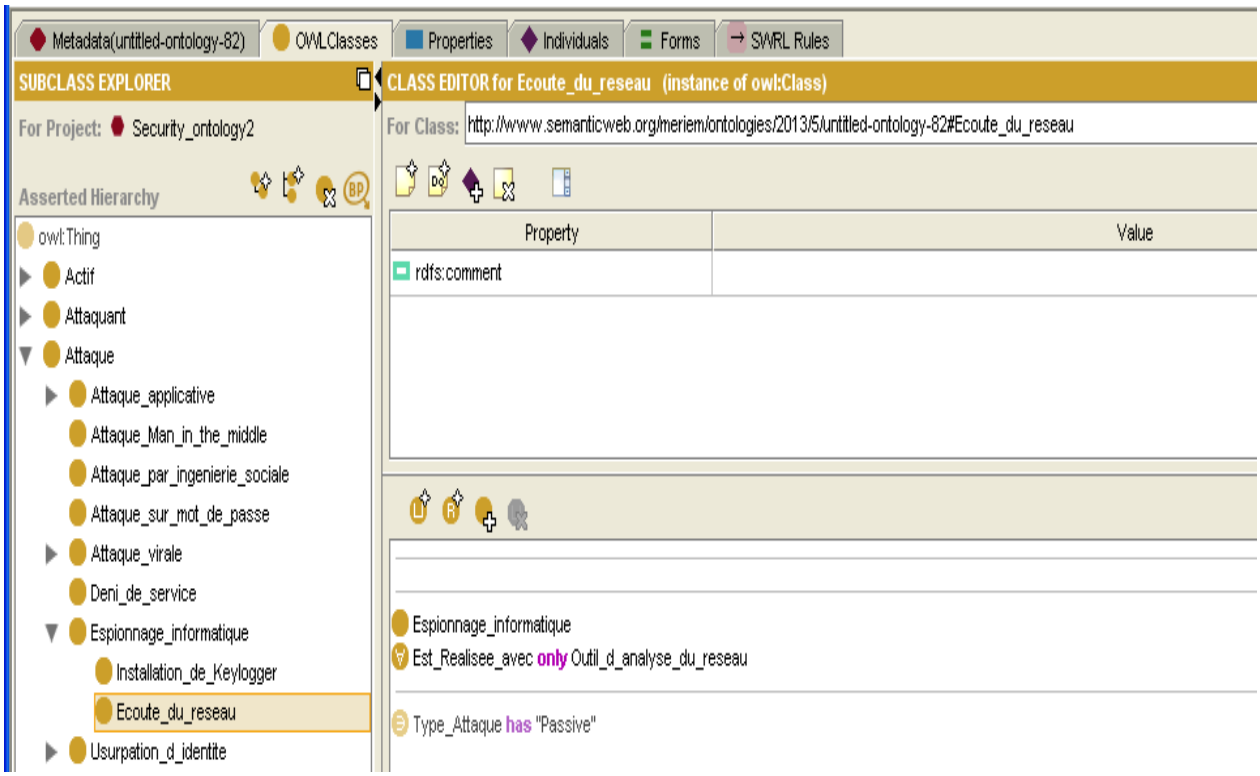


Figure 4.7. Définition des valeurs des attributs, Exemple « Type_Attaque » has « Passive »

Nous passons maintenant à l'instanciation des relations existantes entre chaque deux concepts, afin de relier les instances entre elles, tout en respectant les restrictions faites au niveau conceptuel.

La figure 4.8 donne un aperçu des relations instanciées entre des individus (instances).

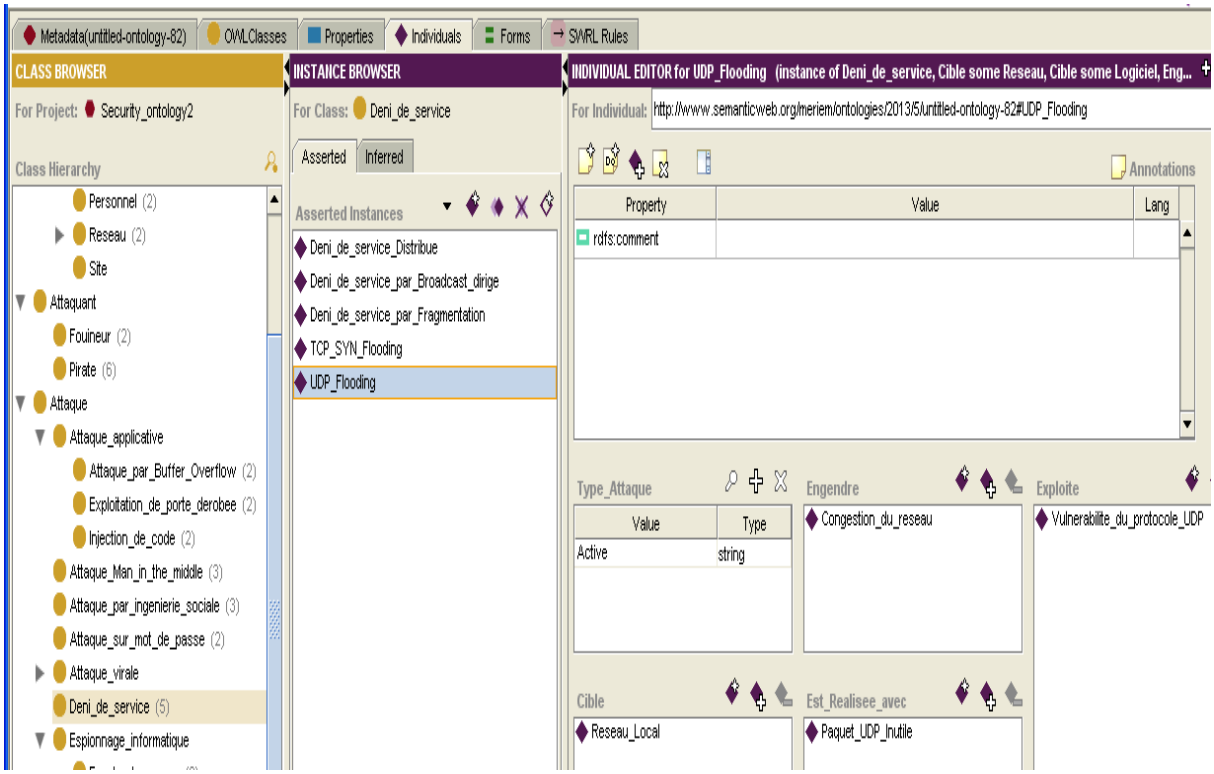


Figure 4.8. Instanciation des relations de la TBox dans la ABox.

Nous obtenons à présent une ontologie implémentée sur Protégé au format OWL. Avant de l'exploiter, il est nécessaire de la tester afin de vérifier sa consistance et de corriger ses erreurs.

4.3. Test de la consistance de l'ontologie

Une fois l'implémentation faite, il est nécessaire de la tester et de vérifier qu'il ne contient pas d'erreur. Pour vérifier la consistance de l'ontologie, nous utilisons le raisonneur Pellet 1.5.2 inclut dans Protégé 3.5. Son utilisation de fait à partir de l'interface protégé en lançant la commande Check Consistency comme montré dans la figure 4.9.

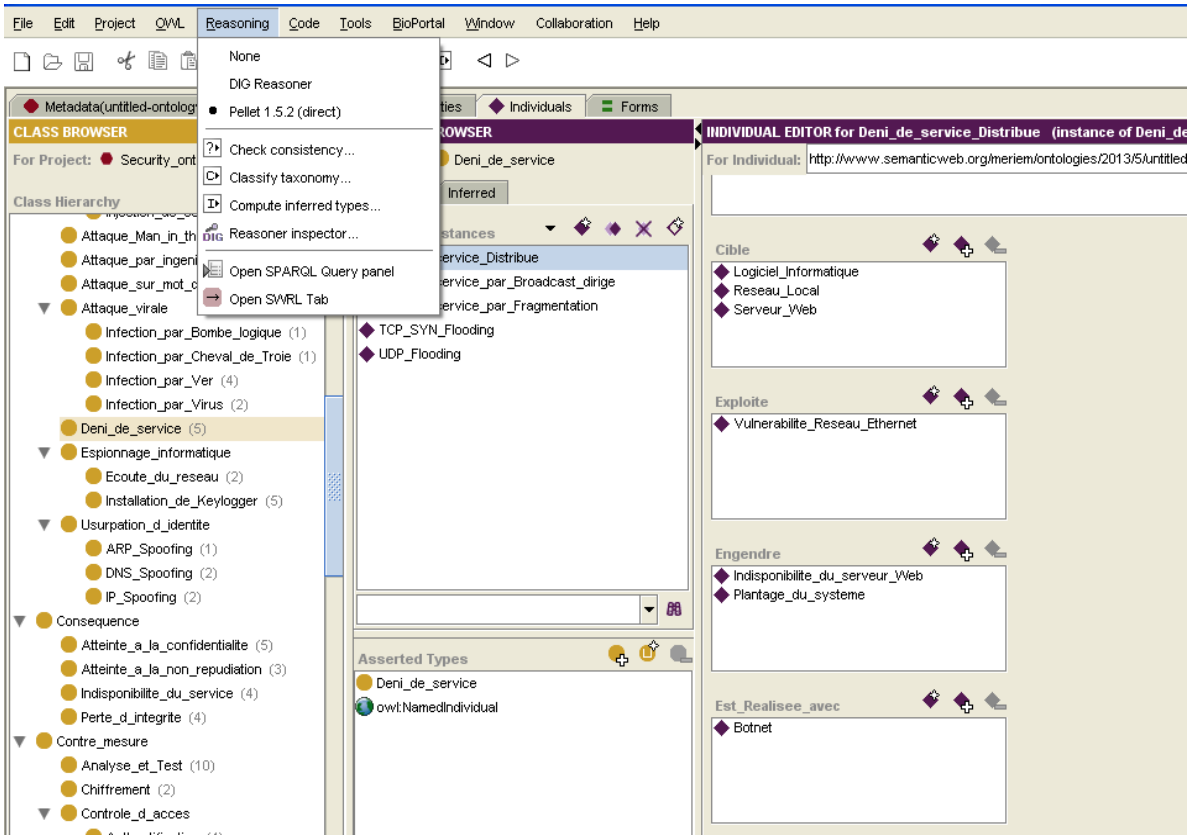


Figure 4.9. Lancement du raisonneur Pellet 1.5.2.

Pour effectuer le test, nous utilisant la commande « Check Consistency ». Une fois le test fait, cette commande ne retourne aucune erreur, donc notre ontologie est bien consistante. La figure 4.10 donne un aperçu du résultat du test de consistance.

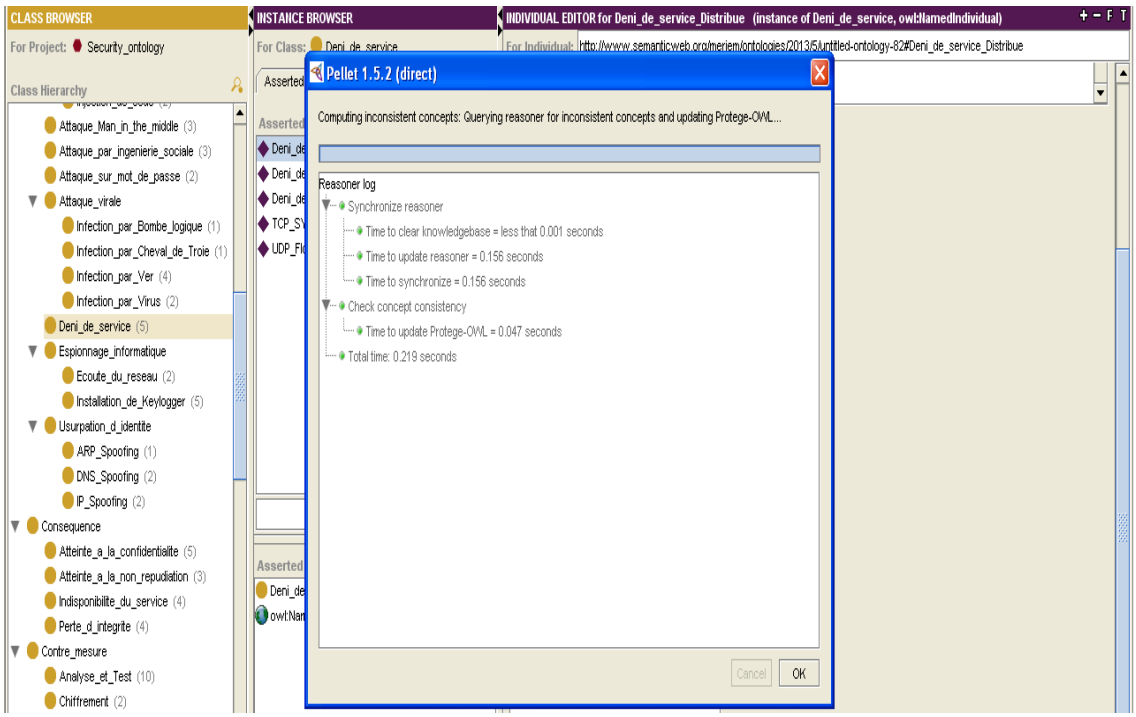


Figure 4.9. Résultat du test de consistance effectué avec le raisonneur Pellet 1.5.2.

4.5. Visualisation de l'ontologie

Nous pouvons visualiser l'ontologie en utilisant deux outils : OWLViz et OntoGraf.

- **Visualisation de l'ontologie en utilisant OntoGraf :**

OntoGraf fournit un schéma d'une partie de l'ontologie. Voici un extrait de l'ontologie, visualisé avec OntoGraf.

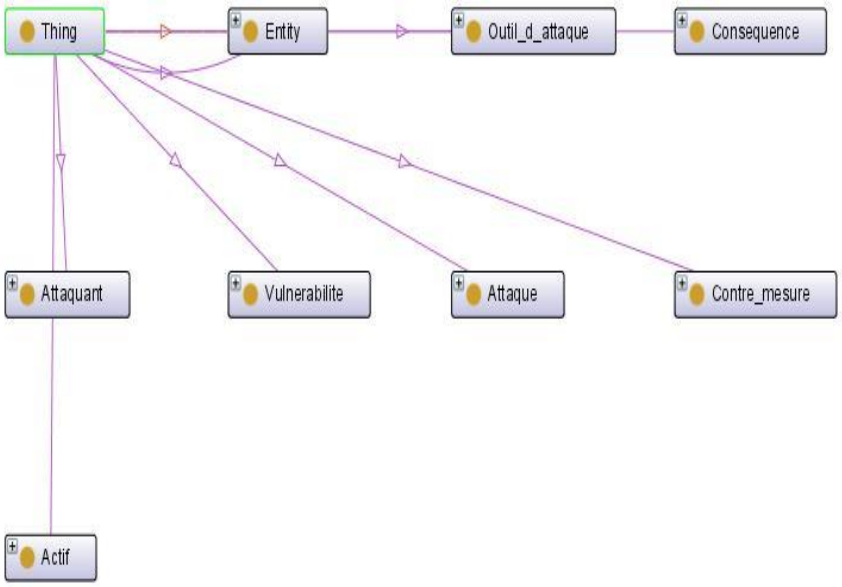


Figure 4.10. Visualisation d'un extrait de l'ontologie avec OntoGraf.

OntoGraf permet aussi de visualiser les concepts, les instances et les relations à la fois. La figure 4.10 donne un exemple ou nous voulons visualiser l'instance « Carder » du concept « Pirate ».

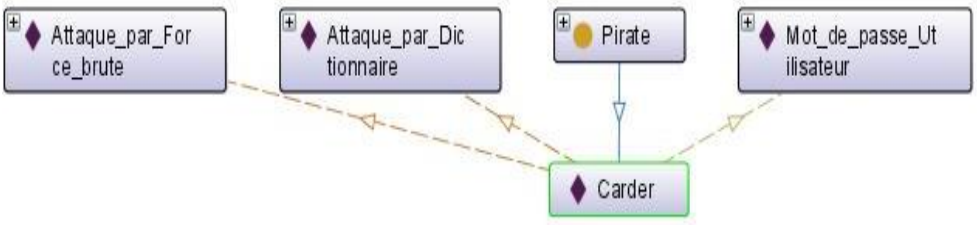


Figure 4.10. Visualisation de l'instance « Carder » avec OntoGraf.

- **Visualisation de l'ontologie en utilisant OWLViz :**

OWLViz fournit un schéma représentant l'ontologie complète ou d'une partie. Ce schéma est un arbre qui a comme nœud père « Thing » et comme nœuds fils les concepts principaux. Puis pour chaque concept, une hiérarchie lui sera reliée. Les figures 4.11 et 4.12 montrent la visualisation de l'ontologie en utilisant OWLViz.

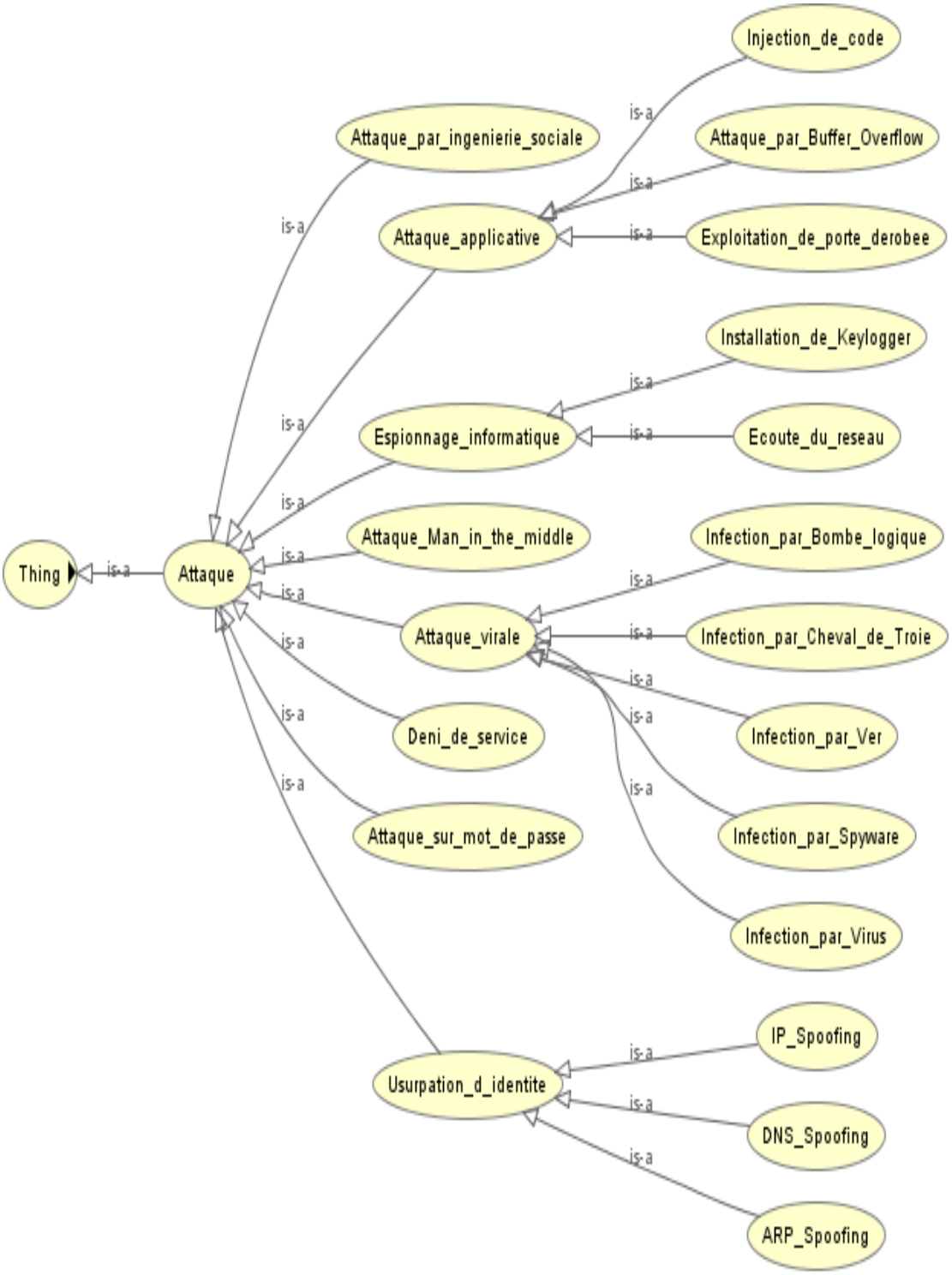


Figure 4.11. Visualisation d'une partie de l'ontologie avec OWLViz.

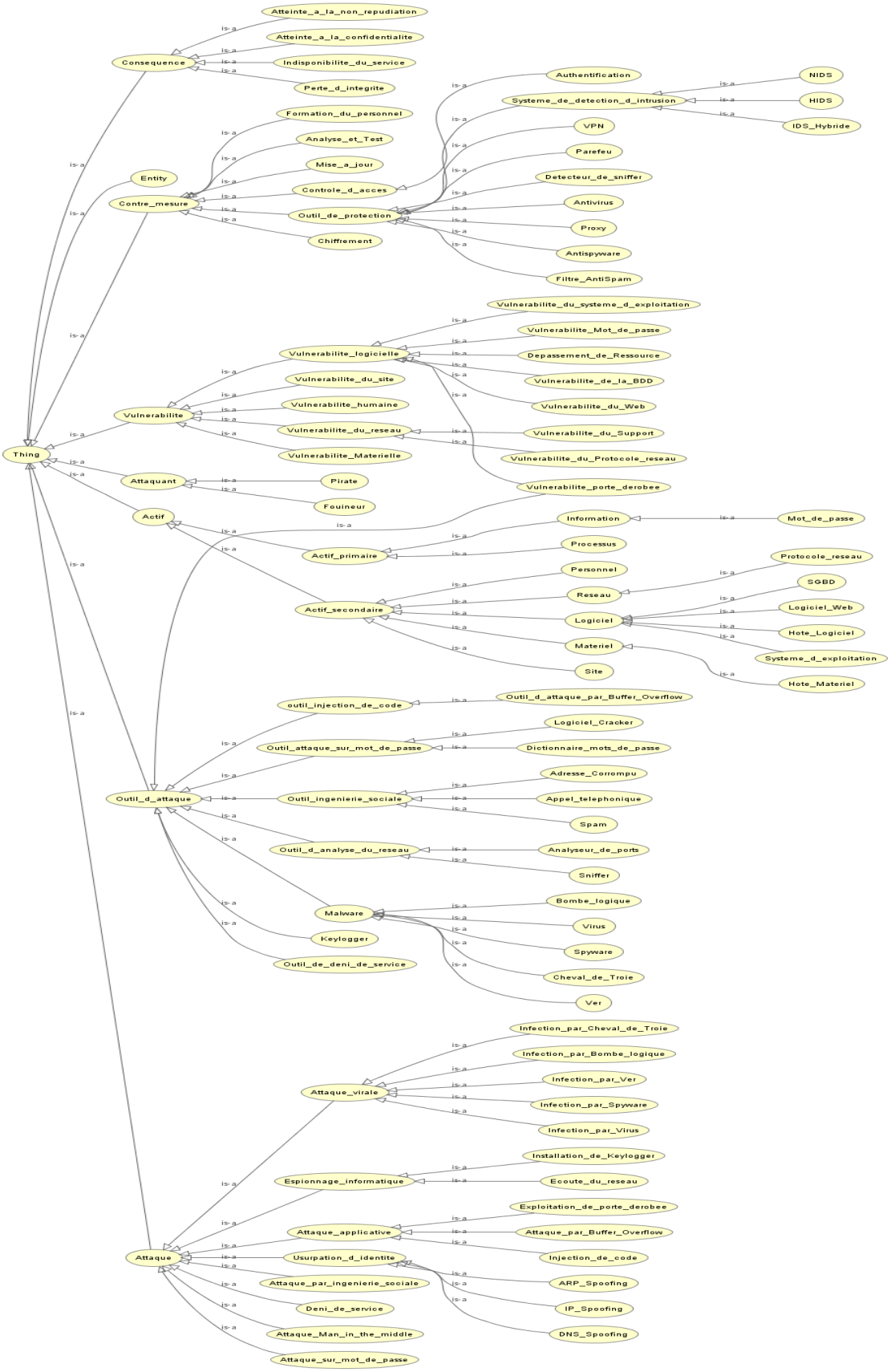


Figure 4.12. Visualisation de l'ontologie complète avec OWLViz.

4.4. Les règles SWRL et les inférences

Puisque nous avons maintenant une ontologie consistante, nous allons implémenter des règles SWRL afin de pouvoir faire des inférences, et déduire de nouvelles relations à partir de celles que l'on a. Pour cela, nous allons créer de nouvelles relations qui vont être déduite. La figure 4.12 présente le nouveau diagramme des relations binaires qui contient les relations qui seront déduites.

- **Diagramme des relations binaires :**

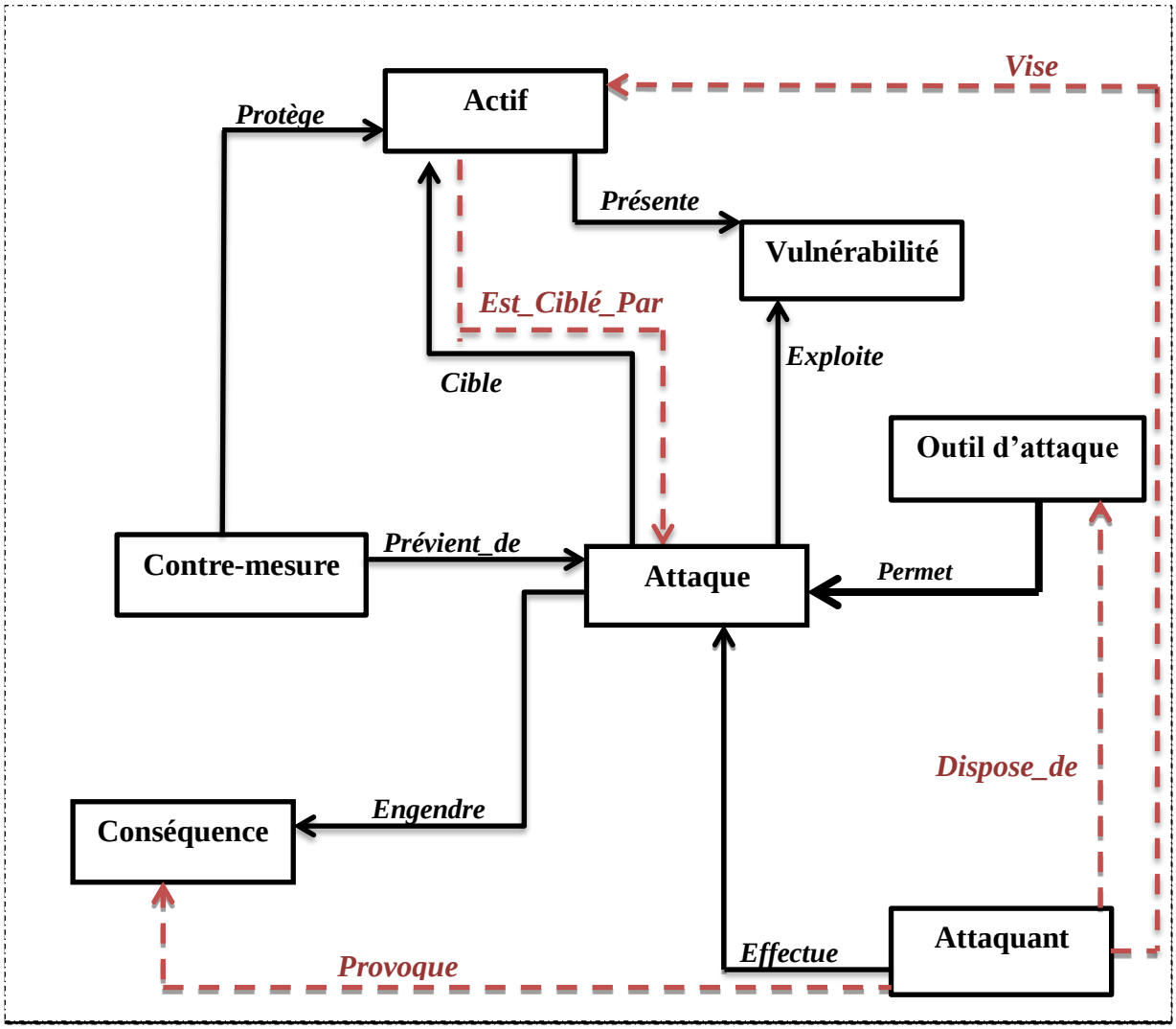


Figure 4.12: Digramme de classification des concepts et des relations binaires.

Pour déduire ces relations entre les instances, il est nécessaire de déclarer les règles correspondantes.

Règle 1 : si un attaquant effectue une attaque, et si cette attaque engendre une conséquence alors cet attaquant a bien provoqué cette conséquence. Se traduit en langage des règles SWRL par :

Attaquant (?x) /\ Attaque (?y) /\ Conséquence (?z) /\ Effectue(?x , ?y) /\ Engendre (?y , ?z)
→ Provoque (? x , ? z).

Règle 2 : Si un attaquant effectue une attaque, et que cette attaque est effectuée en utilisant un outil alors l'attaquant dispose de l'outil d'attaque. Cette règle se traduit par :

Attaquant (?a) /\ Attaque (?b) /\ Outil_d_attaque (?c) /\ Effectue(?a , ?b) /\ Est_realisé_Avec(?b , ?c) → Dispose_de(?a , ?c).

Règle 3 : Si un attaquant effectue une attaque, et que cette attaque cible un actif alors on déduit que cet attaquant vise cet. La règle correspondante est :

Attaquant (?a) /\ Attaque (?b) /\ Actif(?d) /\ Effectue(?a , ?b) /\ Cible(?b , ?d) → Vise (?a , ?d).

Règle 4 :

Cette règle concerne un cas particulier qui est la relation inverse. La relation inverse signifie qui si un concept A est reliée à un concept B avec une relation R, la relation I reliera le concept B au concept A, avec R est Sémantiquement l'inverse de I. Par exemple, si une Attaque Cible un Actif, alors cet Actif sera Ciblé_par cette Attaque. Cette règle peut être traduite par :

Attaque (?x) /\ Actif (?y) /\ Cible (?x, ?y) → Est_Ciblé_Par (?y ,?x).

➢ Implémentation des règles sur Protégé :

L'écriture des règles dans protégé se fait au niveau de l'onglet SWRLTab, présenté dans la figure4.13.

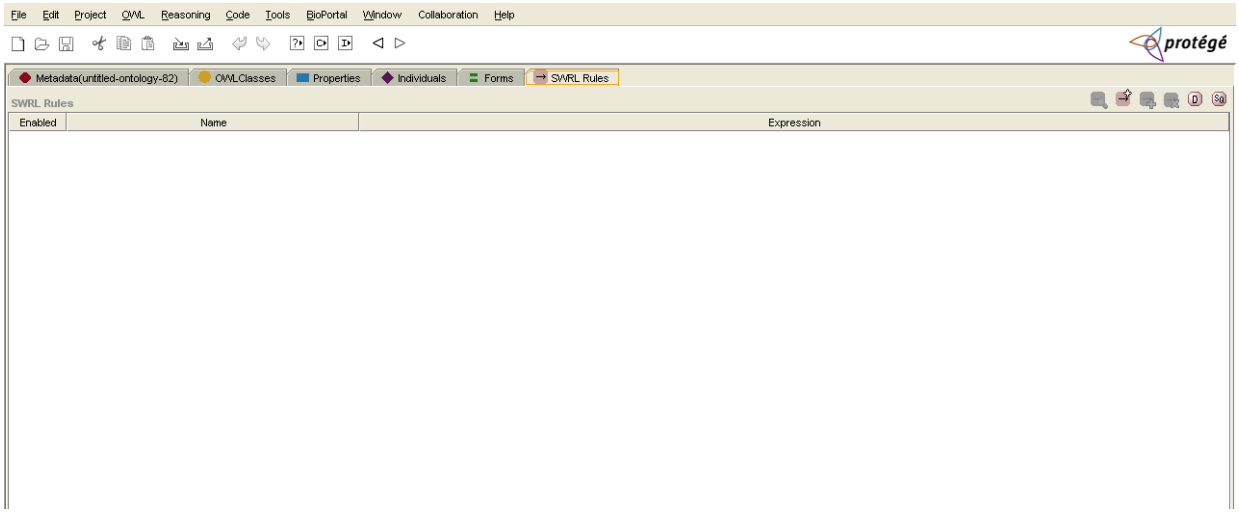


Figure 4.13. Présentation de l'onglet SWRLTab pour les règles SWRL sur Protégé.

Pour ajouter une nouvelle règle, utiliser le bouton « Create new Rule » , qui se trouve dans l'onglet SWRLTab. Nous allons implémenter les règles définies sur cette interface. Nous commençons d'abord par la règle 1 comme le montre la figure 4.14.

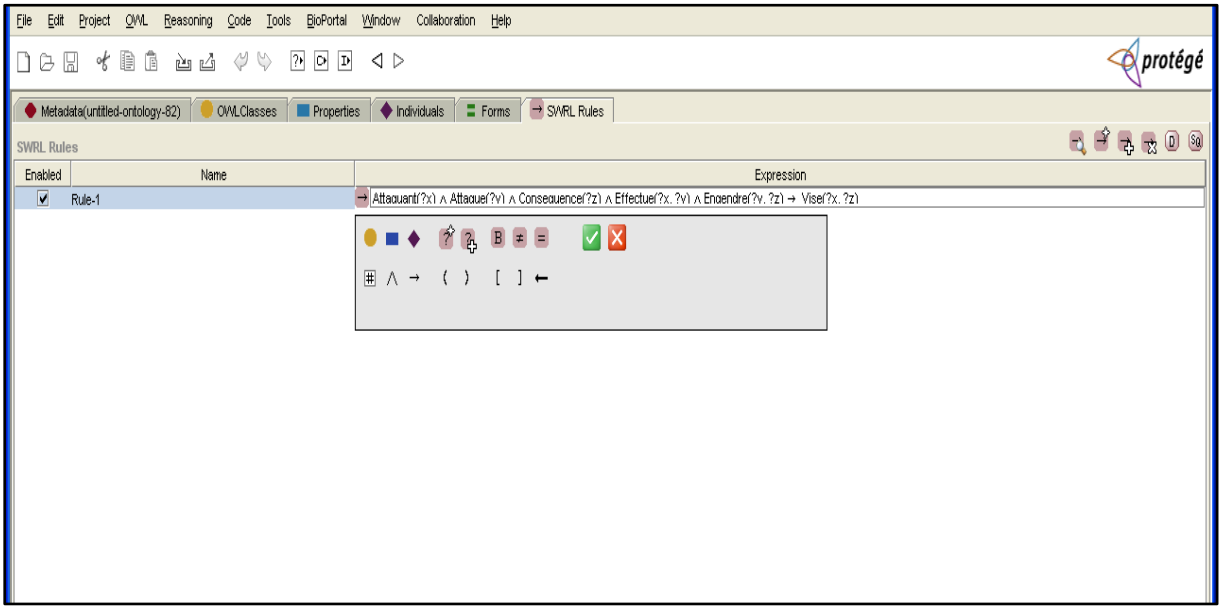


Figure 4.14. Ecriture de la règle 1 dans Protégé.

Pour Exécuter cette règle, nous utilisons le bouton « Activate/Desactivate DroolsTab » qui permet de préparer les requêtes, de les charger et de les exécuter comme le montre la figure 4.15.

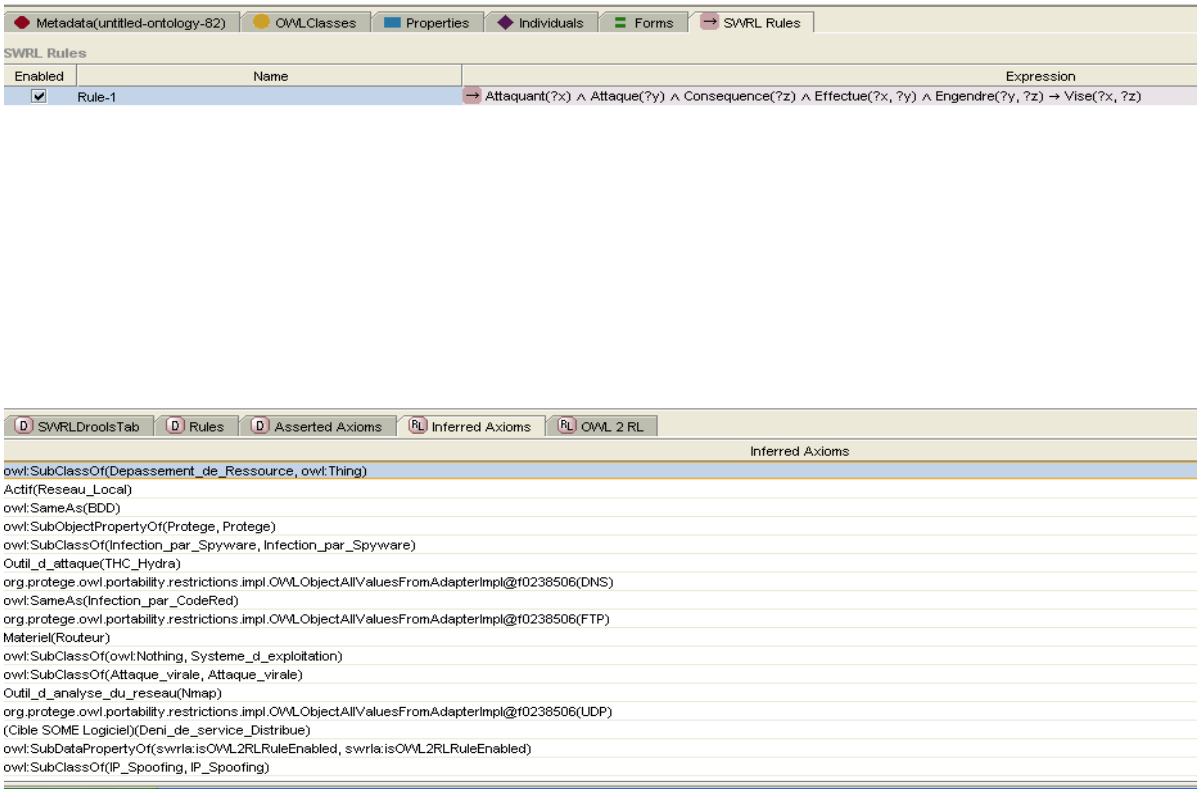


Figure 4.15. Exécution de la règle 1.

Apres l'exécution de la règle, nous allons vérifier ses résultats dans les relations que nous avons ajouté, et qui doivent être crées avec les inférences de cette règle. Nous allons voir les instances du concept « Attaquant » pour voir si de nouvelles relations ont été créés. Par exemple, pour l'instance « Cracker », nous remarquons que la relation « Vise » a été établie avec des instances d'actifs comme : « Application_Web », « Linux », « Mot_de_passe_Utilisateur » ,…etc. Effectivement, cette relation existe bel et bien entre ces instances. Nous concluons que notre règle a bien été définie et effectue des inférences.

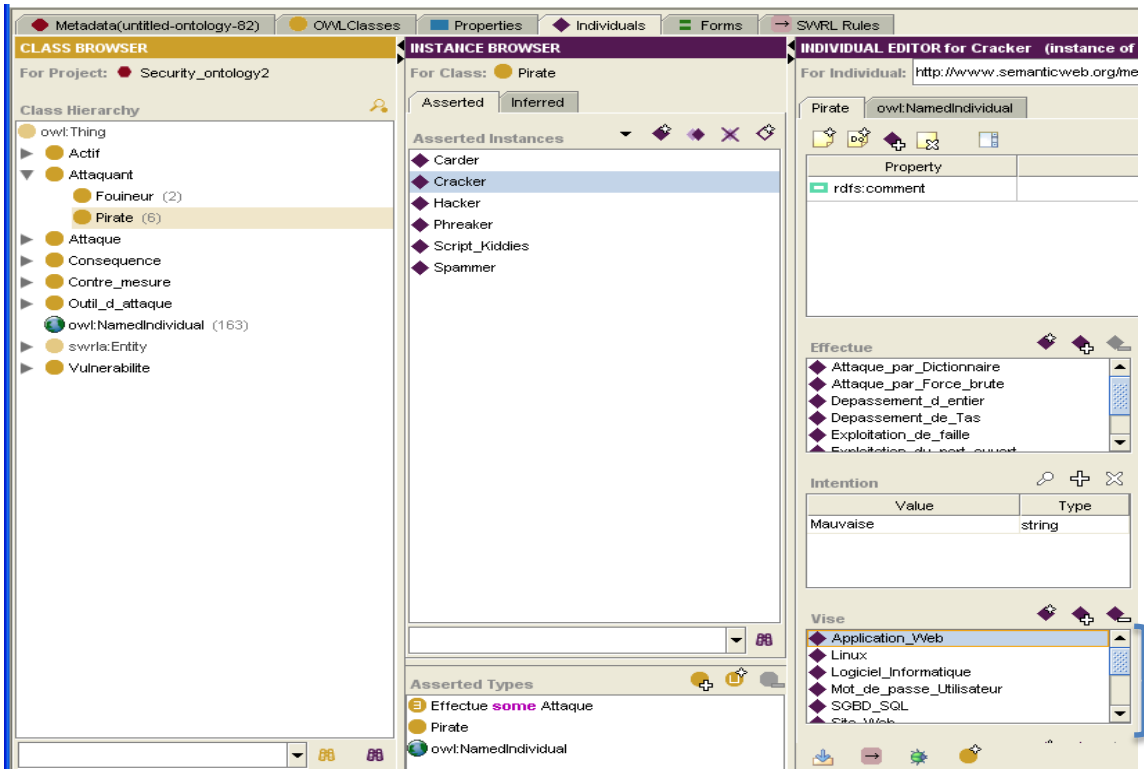


Figure 4.16. Résultat de l'exécution de la Règle 1.

Nous allons implémenter les autres règles sur Protégé afin de créer de nouvelles relations entre les instances de l'ontologie. La figure 4.17 donne un aperçu des règles implémentées sur Protégé.

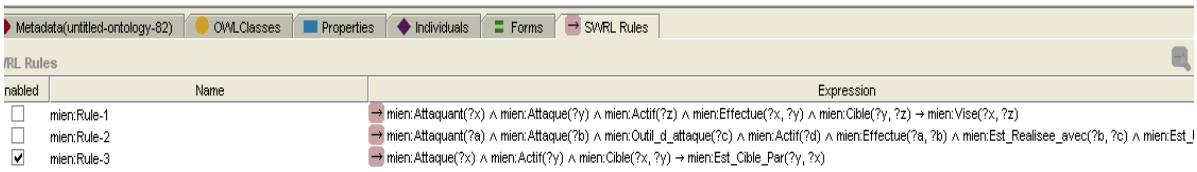


Figure 4.17. Implémentation des Règles SWRL.

4.5. Interrogation de l'ontologie

Nous pouvons interroger notre ontologie de deux manières.

- **Effectuer des requêtes en utilisant le Plugin Queries**

Une manière pour effectuer le requetage dans Protégé est d'utiliser le plugin Queries. Ce Plugin est limité aux requêtes de type « Contient » ou « Ne contient pas ». Néanmoins, dans une ontologie ayant un grand nombre d'individus, cela reste un moyen efficace d'obtenir des réponses rapidement. Il permet également de nommer les requêtes afin de pouvoir les réutiliser ultérieurement. Dans l'exemple ci-dessous, nous avons voulu connaitre les attaques qui exploitent la vulnérabilité du protocole UDP. Nous avons eu comme résultat l'attaque « UDP Flooding ».

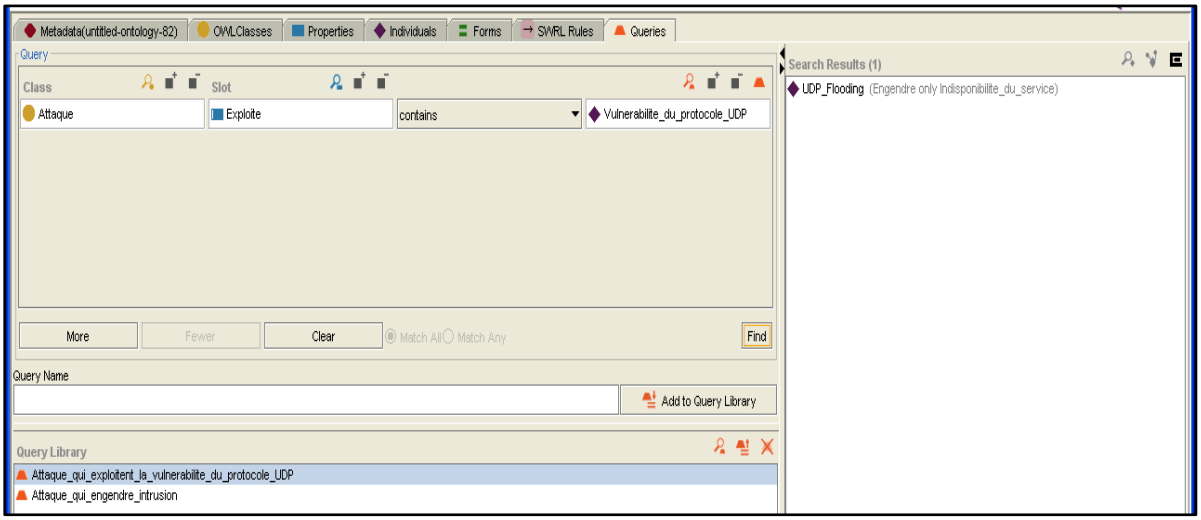


Figure 4.18. Exemple 1 de requête avec le Plugin Queries.

Nous allons effectuer une deuxième requête où nous sélectionnerons toutes les attaques qui ont comme conséquence l'intrusion de l'attaquant dans le système. Le résultat de la requête est illustré dans la figure 4.19.

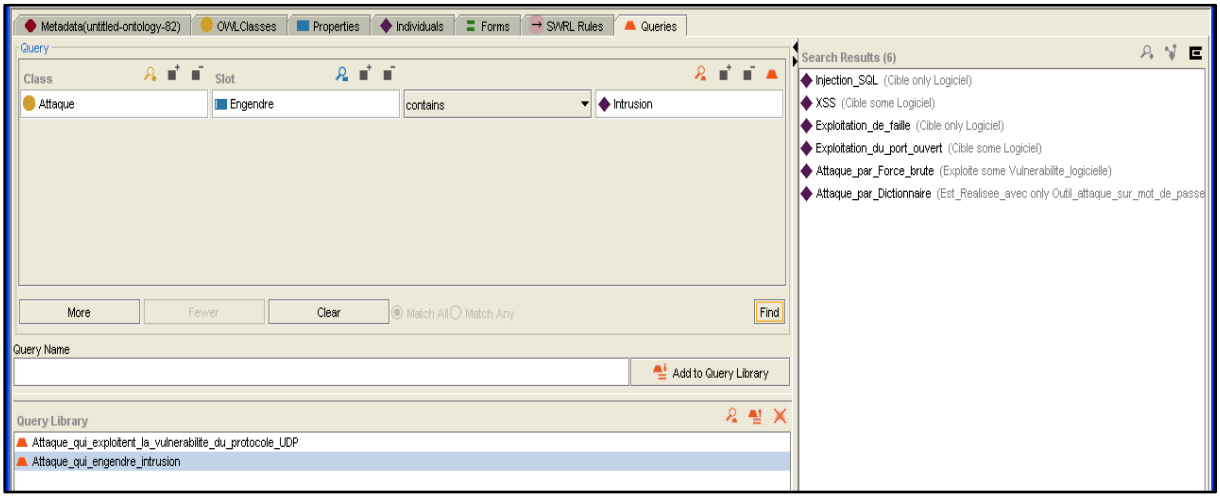


Figure 4.19. Exemple 2 de requête avec le Plugin Queries.

- **Interroger l'ontologie en utilisant le langage SPARQL sur SPARQL Query Panel**

Protégé propose un Plugin qui permet d'exécuter des requêtes SPARQL directement sur l'ontologie. C'est « SPARQL Query Panel ». Nous allons exécuter des requêtes en utilisant le langage SPARQL.

Requête 1 : Cette requête permet de récupérer tous les concepts qui sont reliés par la relation « SubClasseOf », la relation de subsomption ou d'héritage.

```
SELECT ?subject ?object
WHERE { ?subject rdfs:subClassOf ?object }
```

Le résultat de la requête est illustré dans la figure 4.20.

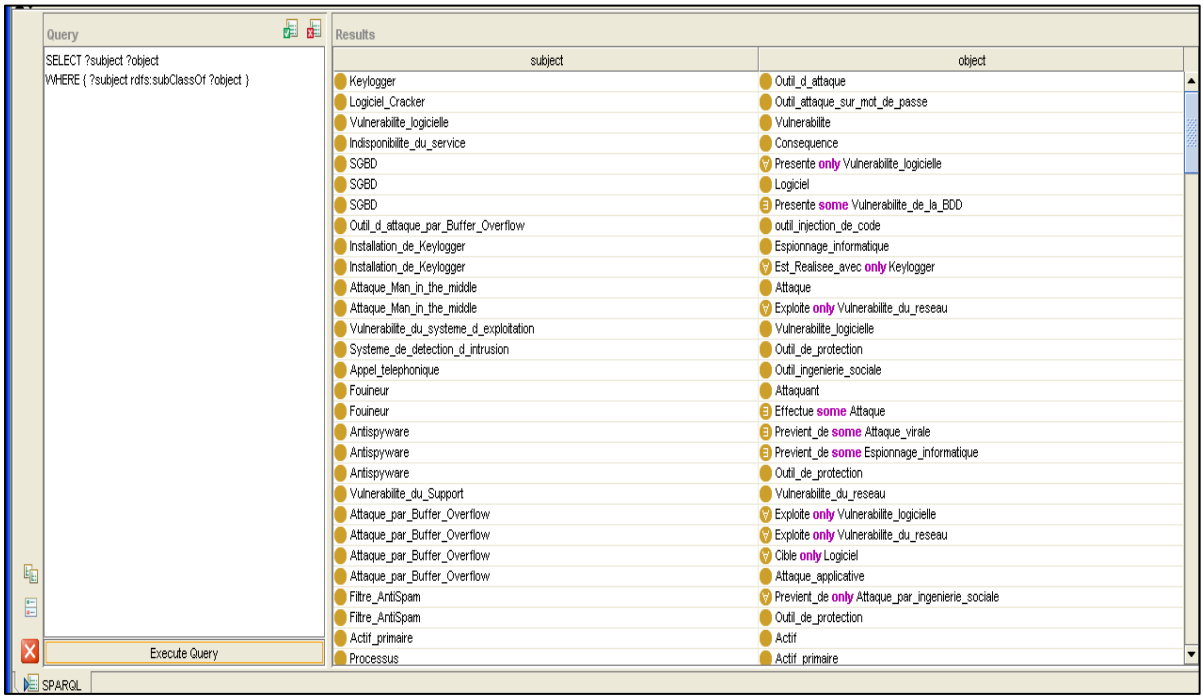


Figure 4.20. Exemple 1 de l'exécution d'une requête SPARQL sur Protégé.

Requête 2 : Cette requête permet de récupérer toutes les instances qui sont reliés par la relation «Cible».

```
SELECT ?subject ?object
WHERE { ?subject :Cible ?object }
```

Le résultat de la requête est illustré dans la figure 4.21.

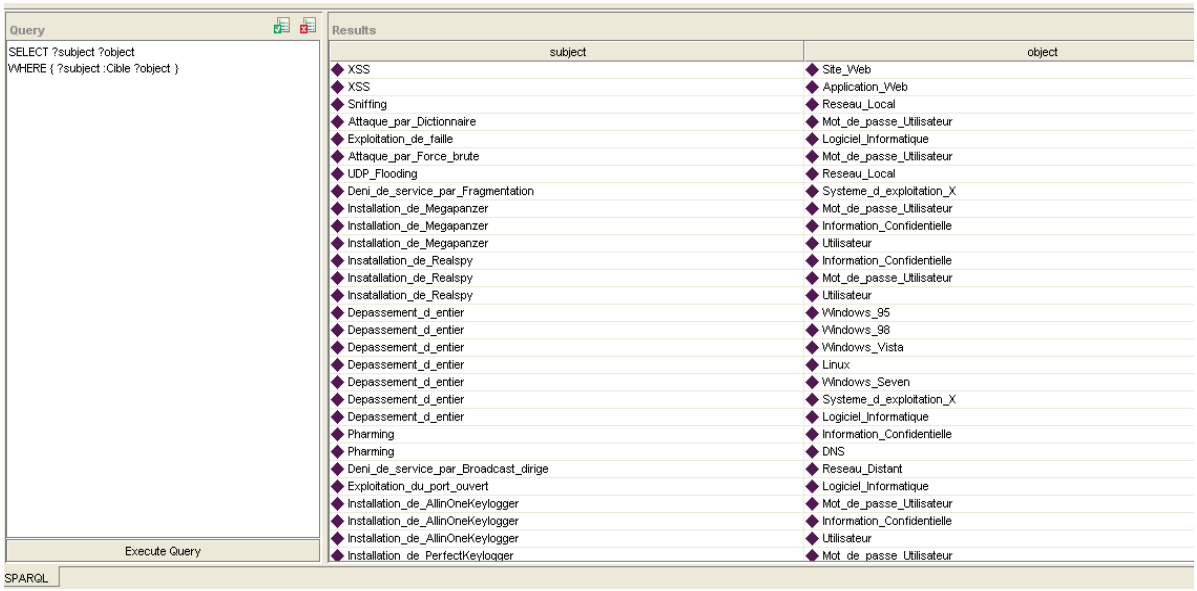


Figure 4.21. Exemple 2 de l'exécution d'une requête SPARQL sur Protégé.

Requête 3 : Cette requête permet de récupérer pour chaque attaque, l'actif visé et l'outil d'attaque utilisé. Le résultat de la requête est illustré dans la figure 4.21.

```
SELECT ?subject ?object ?outil
WHERE { ?subject :Cible ?object.
        ?subject :Est_Realisee_avec ?outil.
}
```

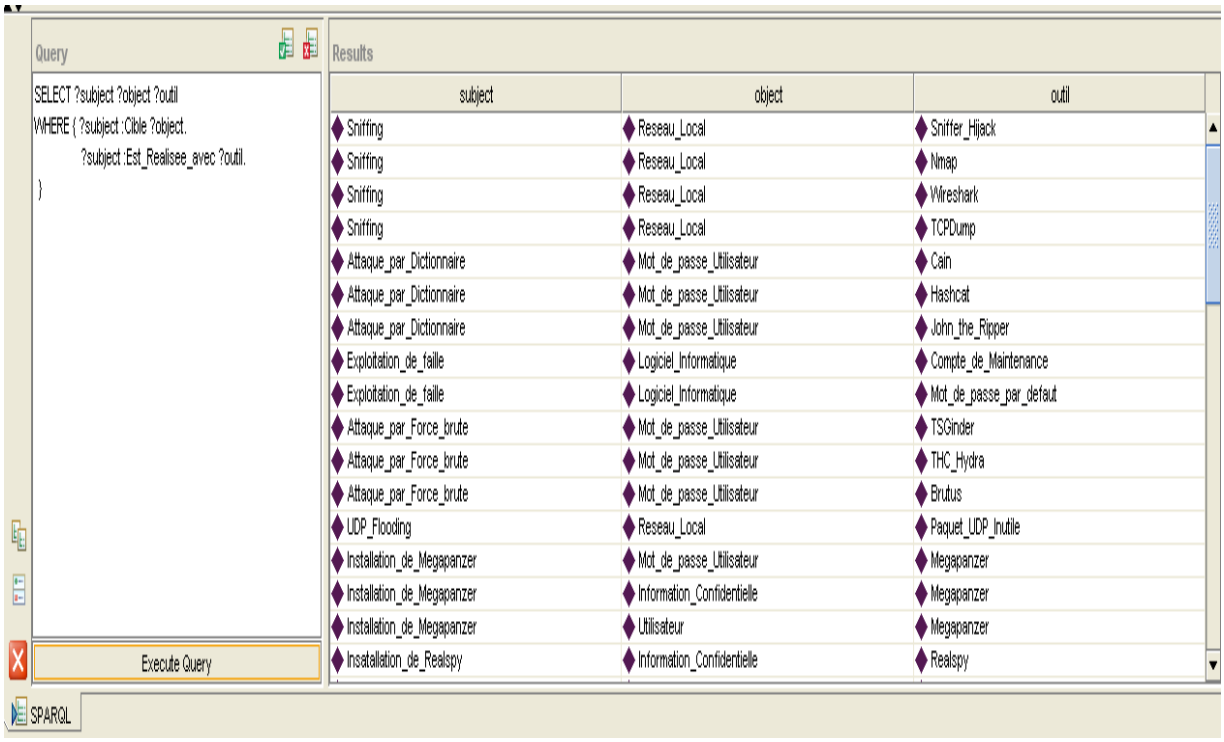


Figure 4.22. Exemple 3 de l'exécution d'une requête SPARQL sur Protégé.

Requête 4 : Cette requête permet de donner les contre-mesures qui préviennent de chaque attaque.

```
SELECT  ?object ?subject
WHERE { ?subject :Previent_de ?object }
```

Le résultat de l'exécution de cette requête est présenté dans la figure 4.23.

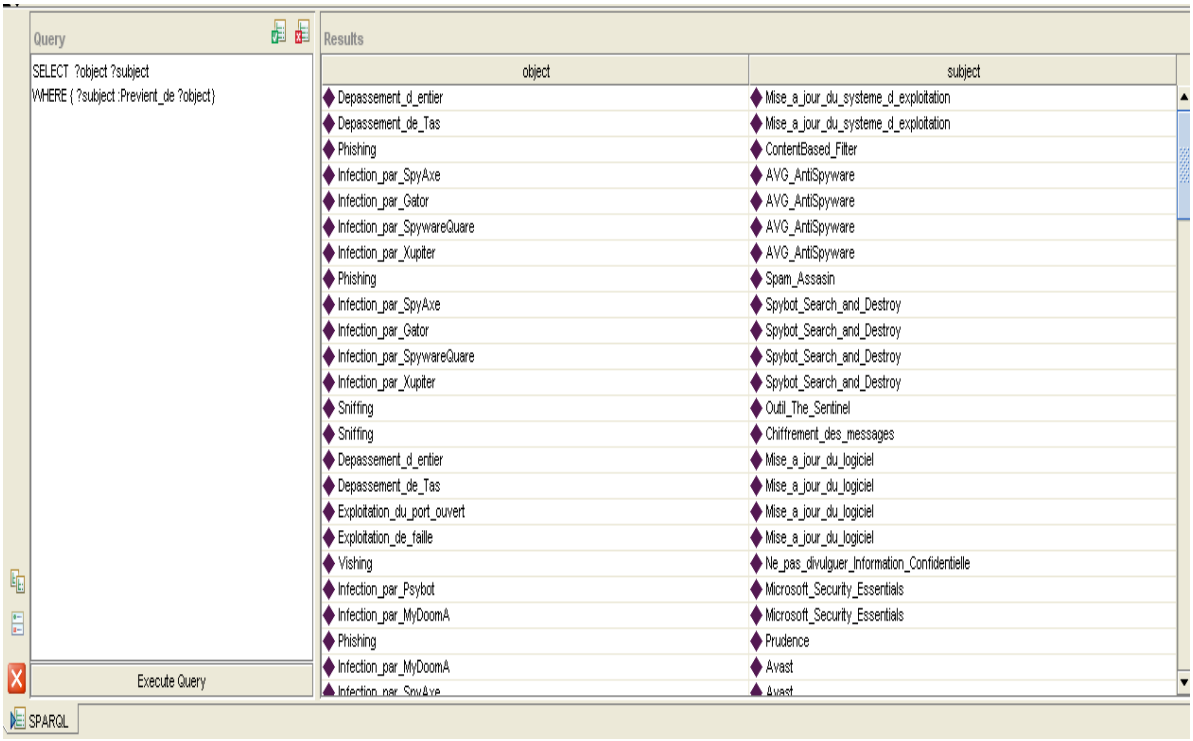


Figure 4.23. Exemple 4 de l'exécution d'une requête SPARQL sur Protégé.

4.6. Conclusion

Dans ce chapitre nous avons présenté en détail l'implémentation de notre ontologie. Nous sommes passés d'une ontologie formelle à une ontologie consistante. Et pour cela nous avons utilisé l'éditeur d'ontologie Protégé. Nous avons d'abord commencé par la définition des concepts, attributs et relations. Puis nous avons défini toutes les instances de notre domaine et nous les avons reliées entre elles en instanciant les relations définies précédemment. Puis, nous avons testé la consistance de l'ontologie obtenue en utilisant le raisonneur Pellet. Ensuite, nous avons définis des règles SWRL afin d'effectuer des inférences et déduire de nouvelles connaissances. Nous avons exécuté ces règles et les avons évalué leurs résultats. Enfin nous avons interrogé notre ontologie de différentes manières. D'abord, nous avons utilisé le Plugin « Querie » offert par Protégé. Puis, nous nous sommes intéressés aux requêtes SPARQL en utilisant l'onglet « SPARQL Query Panel ». Et enfin nous avons essayé l'interrogation en utilisant l'API Jena.

A la fin de ce travail, nous avons obtenu une ontologie consistante et nous l'avons testé en inférant de nouvelles connaissances et en l'interrogeant. L'ontologie que nous avons développée peut être à présent exploitée pour être intégrée dans les systèmes ou pour la recherche. Elle peut être enrichi avec de nouveaux concepts ou fusionnée avec d'autres ontologies pour peut-être faire partie du Web de demain.

Conclusion Générale

Conclusion générale

Au terme de notre travail, il convient de noter que l'utilisation de l'ontologie pour formaliser le domaine de la sécurité informatique s'avère très efficace. En effet les menaces informatiques connaissent une croissance considérable, ce qui implique que l'on doit porter une certaine attention à la stratégie employée pour leur mise en œuvre afin de pouvoir atténuer les conséquences.

Dans ce travail, nous avons commencé avec la définition de la sécurité informatique, ses caractéristiques, ses objectifs, les types d'attaques auxquelles elle devra faire face. Nous avons aussi énuméré la notion de la politique de sécurité et les méthodes de gestion de risques qui permettent de prévenir les risques. Ensuite, nous avons explicité la solution utilisée pour la modélisation de l'information collectée à partir d'un corpus composé de plusieurs documentations. Nous avons détaillé dans cette section, les composants, outils de développement, les éditeurs et bien d'autres choses. De plus, nous avons approfondi les recherches en décortiquant chaque concept, établi les relations entre eux et les individus pouvant être rattachés à chaque concept. Enfin, le dernier chapitre a développé l'aspect pratique proprement dit. Nous avons mis l'accent sur la réalisation, l'exploitation, les inférences, l'interrogation de l'ontologie par des requêtes.

A la fin de ce travail, nous avons obtenu une ontologie consistante et nous l'avons testé en inférant de nouvelles connaissances et en l'interrogeant. L'ontologie que nous avons développée peut être à présent exploitée pour être intégrée dans les systèmes ou pour la recherche. Elle peut être enrichi avec de nouveaux concepts ou fusionnée avec d'autres ontologies pour peut-être faire partie du Web de demain.

Bibliographie :

[01] : Guide de développement d'une base de connaissances d'analyse des risques MEHARI, MEHARI 2010, Mars 2010, CLUSIF.

[02] : Tout sur la sécurité informatique, CommentCaMarche.net, 2eme édition, Jean-François Pillou et Jean-Philippe Bay, Dunod 2009.

[03] : Livre Blanc, La sécurité des systèmes d'information industriels, EURIWARE, Révision Mai 2010, France.

[04] : Guide de la sécurité des systèmes d'information, Robet Longeon et Jean-Luc Archimbaud, Centre National de la Recherche Scientifique (CNRS),1999, Paris, France.

[05] : Base de connaissances MEHARI 2010, Edition 2-14, CLUSIF, Mars 2010, Paris, France.

[06] : Utilisation de la méthode EBIOS : de l'organisation projet aux composants du SMSI (Système de Management de la Sécurité de l'Information), Philippe TOURRON, Université de la Méditerranée, Marseille, et Matthieu GRALL, Agence Nationale de la Sécurité des Systèmes d'Information, 2009, Paris, France.

[07] : Livre Blanc, Le Risque Informatique, Sécurité informatique et devoirs des entreprises, Dr Patrice Guichard, 29/05/2006.

[08] : Gérer ses risques avec la norme ISO 27005 et MEHARI, Annonce de MEHARI 2010, Jean-Philippe Jouas, CLUSIF, 27 Janvier 2010, Paris, France.

[09] : Sécurité Informatique : Principes et méthode, Laurent Bloch, Christophe Wolfhugel, EYROLLES, 15/05/2009.

[10] : Sécurité des réseaux de l'information : Proposition pour une approche politique européenne, Communication de la Commission européenne du 06 Juin 2001.

[11] : Vers une culture de la sécurité, Lignes directrices de l'OCDE (Organisation de Coopération et de Développement Economiques) régissant la sécurité des systèmes et réseaux d'information, 25/07/2002.

[12] : MEHARI 2010, Présentation générale de MEHARI 2010, CLUSIF, Janvier 2010, Paris, France.

[13] : MEHARI 2010, Guide de l'analyse et du traitement des risques, CLUSIF, Janvier 2010, Paris, France.

[14] : Evaluation quantitative de la sécurité des systèmes d'information, Rodolphe ORTALO, Laboratoire d'Analyse et d'Architecture des Systèmes du Centre National de la Recherche Scientifique. Docteur de l'Institut National Polytechnique de Toulouse, 19/05/1998, Toulouse, France.

[15] : MEHARI 2010, Guide de la démarche d'analyse et du traitement des risques, CLUSIF, Janvier 2010, Paris, France.

[16] : MEHARI 2010, Manuel de référence de la base de connaissance MEHARI 2010, CLUSIF, Janvier 2010, Paris, France.

[17] : MEHARI 2010, Principes fondamentaux et spécifications fonctionnelles, CLUSIF, Janvier 2010, Paris, France.

[18] : RFC(Requests For Comments) 2196 Site Security Handbook (Développée en Juillet 1991).

[19] : Menaces sur les systèmes d'information, CLUSIF, 20/11/2003, Paris, France.

[20] : CCNA Exploration 4.0, Accès aux réseaux étendus.

[21] : Bachimont Bruno, L'intelligence artificielle comme écriture dynamique: de la raison graphique à la raison computationnelle, Paris , 1999.

[22] : BERNERS Lee The Semantic Web, 2001. Volume : Scientific American.

[23] : Borst W N Construction of engineering ontologies, 1997. University of Tweenty, Enschede, Centre for Telematica and Information Technology.

[24] : Bouarab Farida, These de doctorat, Modélisation basée ontologies pour l'apprentissage interactif - Application à l'évaluation des connaissances de l'apprenant, 2010, Tizi Ouzou.

[25] : Dieng et Autres R Méthodes et outils pour la gestion des connaissances, Une approche pluridisciplinaire du Knowledge Management, II^{ème} édition, Dunod, 2001.

[26] : Fernandez Lopez Methonlology : from ontological art toward ontological engineering, 1997, USA.

[27] : Gomez-Perez Ontological Engineering : A state of the art. Expert , 1999.

[28] : Gruber T Translation approach to portable ontology specification, 1993.

[30] : Kayser D ,La représentation des connaissances, Hermes, 1997.

[31] : Lekhchine Riad Construction d'une ontologie pour le domaine de la sécurité: Application aux agents mobiles, 2009. Constantine, Algerie.

[32] : Minsky A Framework for Representing Knowledge. Vol. The psychology of computer vision, 1975, New York, USA.

[33] : Natalya F Deborah L et Noy Ontology Development 101 : A Guide to Creating Your First Ontology" Stanford Knowledge Systems Laboratory Technical Report, 2001.

[34] : Neches al et R Enabling technology for knowledge sharing, AI Magazine, 1991.

[35] : Oberle D. An extensible ontology software environment, 2004, Vol. Handbook on Ontologies.

[36] : Quillian M Semantic memory,Vol. Semantic Information Processing, MIT Press, 1968.

[37] : Troncy A Issac et R DOE : une mise en oeuvre d'une méthode de structuration différentielle pour les ontologies. Rouen, Grenoble : Presses universitaires de Grenoble, 2002.

[38] : Uschold King et Towards a methodology for building ontologies. In workshop on Basic Ontological Issues in Knowledge Sharing. International Joint Conference on Artificil Intelligence, 1995.

[39] : Uschold M Gruninger et M Ontologies Principes, Methods and Application. Knowledge Engineering Review, 1996.

[40] : Uschold M.Gruninger et M. Creating semantically integrated communities on the World Wide Web, 2002.

[41] : WELTY Barry SITH et Christopher Formal Ontology in Information System: Proceedings of the international conference, 2001, New York, USA.

[42] : Xavier Lacot - Introduction à OWL, un langage XML d'ontologies Web. xavier@lacot.org. Juin 2005, rédigé dans le cadre d'un projet à l'Ecole Nationale Supérieure des Télécommunications.

[43] : Yannick Prié, Ingénierie ontologique. UFR Informatique Université Claude Bernard Lyon 1, Aout 2009, France.

[44] : Nilda Ruimy, Silvia Piccini, Emiliano Giovannetti, Andrea Bellandi, LA BASE DE CONNAISSANCE FERDINAND DE SAUSSURE SUR L'ÉDITEUR D'ONTOLOGIES *PROTÉGÉ*

[45] : OntoEdit : Collaborative Ontology Development for the Semantic Web. York Sure, Michael Erdmann, Juergen Angele, Steffen Staab, Rudi Studer, et Dirk Wenke, 2002, Institut AIFB, Université de Karlsruhe, Allemagne.

[46] : Quering the semantic web with Racer + nRQL, Volker Haarslev, Ralf Moller, Michael Wessel, 2004.

[47] : Michel Gagnon, Logique descriptive et OWL, 2012, Polytechnique de Montréal.

[48] : De l'analyse des risques à l'expression des exigences de sécurité des systèmes d'information. Manuel Vasquez, Nadira Lammari, Isabelle Comyn-Wattiau, Jacky Akoka. Laboratoire CEDRIC, CNAM, Paris, France.

[49] : Modélisation et classification automatique des informations de sécurité, Fatiha Benali, Stéphane Ubéda and Véronique Legrand, Jacques Saraydaryan, Gauthier Jombart, Véronique Legrand and Stéphane Ubéda. Ecole doctorale en informatique et information pour la santé. Thèse préparée au Centre d'Innovation en Télécommunications et Intégration de Services (CITI), INSA de Lyon - INRIA Rhône-Alpes et au sein de l'équipe R&D Exaprotect à Villeurbanne, France.

[50]: J. Howard and T.Longstaff. A common language for computer security incidents. Sandia International Laboratories, 1998.

[51] : Jeffrey L Undercoffer, Anupam Joshi, and John Pinkston. Modeling computer attacks an ontology for intrusion detections. The Sixth International Symposium on Recent Advances in Intrusion Detection. Springer, September 2003.

[52] : Le Web Sémantique, En quoi le Web Sémantique permet-il d'aborder le sens ? Nicolas BRULET & Xuan TRUANG, 15 Juin 2010, Université de Compiègne, France.

[53] : Kevin S. Killourhy, Roy A. Maxion, et Kymie M. C. Tan. A defense-centric taxonomy based on attack manifestations. In DSN '04 : Proceedings of the 2004 International Conference on Dependable System and Networks, page 102, Washington, DC, USA, 2004.

Sites Web Visités

[54] : http://fr.wikipedia.org/wiki/Ontologie_(informatique)

[55] : http://protegewiki.stanford.edu/wiki/Protege_Ontology_Library

[56] : http://swoogle.umbc.edu/

[57] : http://www.journaldunet.com/

[58] : https://www.cases.lu/fr/

[59] : http://www.fidens.fr/articles/qu-est-ce-que-la-famille-iso-27000-54.html

[60] : http://www.ysosecure.com/

[61] : http://www.clusif.asso.fr/

[62] : http://www.wikipedia.org/

[63] : http://www-igm.univ-mlv.fr/~dr/XPOSE2009/Le%20Web%203.0/technologies.html

[64] :http://www.memoireonline.com/01/13/6662/m_Conception-d-un-systeme-auteur-pour-lacreation -et-la-manipulation-d-une-base-de-ressources-pedag13.html

[65] : http://protegewiki.stanford.edu/index.php?title=Protege4Views&oldid=5931

[66] : https://interstices.info/jcms/c_17672/ontologies-informatiques

[67] : http://clarkparsia.com/pellet/

[68] : http://www.racer-systems.com/

[69] : http://www.w3.org/

[70] : http://www.w3.org/XML/

[71] : http://www.w3.org/RDF/

[72] : http://www.daml.org/language/

[73] : http://www.w3.org/2004/OWL/

[74] : http://www.xml.com/pub/r/861

[75] : http://www.ksl.stanford.edu/software/ontolingua/

[76] : http://mayor2.dia.fi.upm.es/oeg-upm/index.php/en/technologies/60-webode

[77] : https://code.google.com/p/swoop/

[78] : http://jena.sourceforge.net/tutorial/RDQL/

[79] : http://web-semantique.developpez.com/tutoriels/jena/arq/introduction-sparql/

Annexes

Annexe A : Les logiques de description *[Michel Gagnon, Logique descriptive et OWL,]*

Les logiques descriptives sont une famille de formalismes utilisées pour représenter une base de connaissances d'un domaine d'application. Plus spécifiquement, ces logiques permettent de représenter des concepts (aussi appelés *classes*) d'un domaine et les relations (aussi appelées *rôles*) qui peuvent être établies entre les instances de ces classes. Par exemple, on pourrait les utiliser pour représenter les concepts : *humain*, *femme* et *mère*, et spécifier que toute mère est une femme qui est le parent d'au moins un humain. Comme toute logique, des mécanismes d'inférence y sont associés, permettant ainsi de déduire de nouveaux faits à partir d'une base de connaissances.
Dans une base de connaissances en logique descriptive, on distingue deux composantes : la *TBox* et la *ABox*. La première contient tous les axiomes définissant les concepts du domaine, comme la définition du concept de mère présenté au paragraphe précédent.

La ABox contient des assertions sur des individus, en spécifiant leur classe et leurs attributs. C'est dans la ABox qu'on indiquerait que Marie est une femme et qu'elle a deux enfants. Le type d'inférence réalisé avec la TBox diffère de celui réalisé avec la ABox.

Dans la TBox, on est généralement intéressé à savoir si tous les concepts définis sont consistants, c'est-à-dire si, pour chaque concept, il peut exister au moins un individu membre de cette classe. Par exemple, si on définit une classe comme étant à la fois une sous-classe des classes *homme* et *femme* et que la TBox spécifie aussi que ces deux classes sont disjointes (c'est-à-dire qu'aucune entité ne peut à la fois être un homme et une femme), on se retrouve alors avec un concept inconsistant. Un autre type d'inférence réalisé avec la TBox est la *subsomption*, qui consiste à déduire qu'une classe est une sous-classe d'une autre classe, même si cela n'est pas déclaré explicitement dans la base de connaissances.

Dans la ABox, on retrouve les assertions sur les individus. En d'autres mots, on y spécifie quelles sont les entités du monde et à quelle classe elles appartiennent. La ABox contient aussi des énoncés spécifiant les relations qui existent entre les individus.

Les inférences avec la ABox visent normalement à déterminer si un ensemble d'assertions est *consistant*, c'est-à-dire si un individu déclaré comme instance d'une classe peut réellement être une instance de cette classe et, similairement, si une relation déclarée entre deux individus est réellement possible.

1. Langage de base AL

Les langages de la logique descriptive sont déterminés par la forme des énoncés qui sont permis. La plupart des langages utilisés découlent du langage AL (Attributive Language), dont l'expressivité est plutôt limitée.

1.1. Syntaxe du langage AL

Dans ce langage, les axiomes sont construits à partir d'un ensemble de concepts. Ainsi, pour déclarer qu'un humain est un animal, on utiliserait les concepts atomiques Humain et Animal et on déclarerait l'axiome suivant :

Humain ⊑ Animal

Mais ceci n'est pas très informatif. En fait, cet axiome ne fait que déclarer que les humains forment un sous-ensemble des animaux. Cela ne définit pas vraiment ce qu'est un humain. Pour ce faire, il faut citer d'autres caractéristiques. On sait par exemple qu'un humain est un animal qui raisonne. En définissant le concept Raisonnable, nous pourrions donc définir le concept Humain de la manière suivante :

Humain ≡ Animal ⊓ Raisonnable

Ici nous avons vraiment un énoncé plus précis, qui établit l'équivalence entre deux concepts : d'un côté le concept Humain qui représente l'ensemble des humains, et de l'autre le concept Animal ⊓ Raisonnable qui représente l'ensemble des individus appartenant à la fois à la classe Animal et à la classe Raisonnable.

Tout énoncé de la forme C ≡ D est un concept atomique, il est appelé ***définition***. C'est ce type d'énoncé qu'on utilise pour créer une TBox. En plus des concepts atomiques, que nous définissons nous-mêmes lorsque nous construisons une base de connaissances, le langage AL contient deux classes spéciales : le concept universel ⊤ qui représente tous les individus du monde représenté, et le concept impossible ⊥. Par définition, pour tout concept **C**, on a l'axiome suivant :

C ⊑ ⊤

Soit maintenant un concept **C** qui est impossible, c'est-à-dire qu'aucun individu ne peut appartenir à ce concept. Pour représenter cette situation, on utilise l'axiome suivant :

C ⊑ ⊥

Le langage AL contient la négation, qui ne peut être appliquée qu'à un concept atomique. Ainsi, on peut représenter la classe des non humains, en écrivant ¬**Humain**, mais on ne peut pas représenter la classe des individus qui ne sont pas des animaux raisonnables, c'est-à-dire la classe

¬(Animal ⊓ Raisonnable)

Le langage AL permet aussi de définir un concept par des restrictions sur les relations. On peut par exemple définir la classe des individus dont tous les enfants sont des femmes, en utilisant l'énoncé:

∀ aEnfant.Femme

On peut définir la classe des individus qui ont au moins un enfant, en utilisant la formule suivante :

∃ aEnfant.⊤

Notons que le langage AL ne permet pas de spécifier un concept avec le quantificateur existentiel. Par exemple, on ne peut pas définir la classe des individus qui ont au moins une fille, qui exigerait un énoncée de la forme suivante :

∃aEnfant.Femme

Seul le concept universel est permis avec le quantificateur existentiel. Voici un exemple plus complexe, utilisant des restrictions sur les relations, qui définit le concept d'un père qui n'a que des filles :

Humain ⊓ ∃aEnfant.⊤ ⊓ ∀aEnfant.Femme

Littéralement, ce concept représente l'ensemble des individus qui sont des humains ayant au moins un enfant et dont tous les enfants sont des femmes. À noter que l'énoncé suivant ne serait pas une bonne représentation :

Humain ⊓ ∀aEnfant.Femme

Le problème, c'est qu'un humain qui n'a pas d'enfants serait aussi une instance de ce concept. Supposons par exemple l'existence d'une personne qui n'a pas d'enfants. On ne peut pas dire qu'elle ne respecte pas la restriction. Pour ne pas la respecter, il faudrait trouver au moins un de ses enfants qui n'est pas une fille. Comme cette personne n'a pas d'enfants, on ne trouvera pas un tel contre-exemple, et la restriction est donc respectée.

Pour représenter une personne qui n'a pas d'enfants, nous devons restreindre la valeur de la relation aEnfant au concept impossible :

∀aEnfant.⊥

Pour appartenir à ce concept, un individu doit avoir tous ses enfants appartenant au concept impossible. Il ne peut donc pas avoir d'enfants.

En résumé, les descriptions possibles dans le langage AL sont les suivantes (on suppose que A est un concept atomique et C et que D sont des concepts atomiques ou complexes) [Schmidt-Schauss91] :

Le tableau X illustre les constructeurs offerts par AL

Constructeur	Description
A	concept atomique
⊤	concept universel
⊥	concept impossible
¬A	négation atomique
C ⊓ D	intersection de concepts
∀R.C	restriction de valeur (Universelle)
∃R.⊤	quantification existentielle limitée

Tableau A.1 : Les constructeurs selon AL

Les extensions d'AL

Le tableau ci-dessous illustre des exemples de constructeurs pour augmenter AL [Baader, 2003]. La première colonne contient la lettre qui désigne le constructeur, la deuxième sa syntaxe d'utilisation et la dernière sa sémantique. La nomenclature des LD dicte que pour chaque constructeur ajouté, il faut

agglutiner la lettre correspondante au nom de la logique originale. Par exemple, la logique AL, enrichie l'union (U) et de la quantification existentielle complète ($\mathcal{E}$).

$[\mathcal{O}]$	$\{a_1, a_2, ..., a_n\}$	$\{a_1^{\mathcal{I}}, a_2^{\mathcal{I}}, ..., a_n^{\mathcal{I}}\}$
$[\mathcal{U}]$	$C \sqcup D$	$C^{\mathcal{I}} \cup D^{\mathcal{I}}$
$[\mathcal{E}]$	$\exists R.C$	$\{a \in \triangle^{\mathcal{I}} \mid \{\exists b.(a,b) \in R^{\mathcal{I}}\} \wedge b \in C^{\mathcal{I}}\}$
$[\mathcal{C}]$	$\neg C$	$\triangle^{\mathcal{I}} \backslash C^{\mathcal{I}}$
$[\mathcal{I}]$	R_1^{-1}	$\{(y,x) \mid (x,y) \in R_1^{\mathcal{I}}\}$
$[\mathcal{H}]$	$R_1 \sqsubseteq R_2$	$R_1^{\mathcal{I}} \subseteq R_2^{\mathcal{I}}$
$[\mathcal{F}]$	$= 1R$	$\{x \in \triangle^{\mathcal{I}} \mid \mid \{y \in \triangle^{\mathcal{I}} \mid (x,y) \in R^{\mathcal{I}}\}\mid = 1\}$
	$\geq 2R$	$\{x \in \triangle^{\mathcal{I}} \mid \mid \{y \in \triangle^{\mathcal{I}} \mid (x,y) \in R^{\mathcal{I}}\}\mid \geq 2\}$
$[\mathcal{N}]$	$\geq nR$	$\{a,b \in \triangle^{\mathcal{I}} \mid \mid (a,b) \in R^{\mathcal{I}}\}\mid \geq n\}$
	$\leq nR$	$\{a,b \in \triangle^{\mathcal{I}} \mid \mid (a,b) \in R^{\mathcal{I}}\}\mid \leq n\}$
	$= nR$	$\{a,b \in \triangle^{\mathcal{I}} \mid \mid (a,b) \in R^{\mathcal{I}}\}\mid = n\}$
$[\mathcal{Q}]$	$\geq nR.C$	$\{a,b \in \triangle^{\mathcal{I}} \mid \mid (a,b) \in R^{\mathcal{I}} \wedge b \in C^{\mathcal{I}}\}\mid \geq n\}$
	$\leq nR.C$	$\{a,b \in \triangle^{\mathcal{I}} \mid \mid (a,b) \in R^{\mathcal{I}} \wedge b \in C^{\mathcal{I}}\}\mid \leq n\}$
	$= nR.C$	$\{a,b \in \triangle^{\mathcal{I}} \mid \mid (a,b) \in R^{\mathcal{I}} \wedge b \in C^{\mathcal{I}}\}\mid = n\}$

Tableau A.2 : Exemple de constructeurs de rôles et concepts pour étendre AL.

Les logiques de description qui existent sont des combinaisons des différents éléments du tableau X. par exemple, si on rajoute la négation complète C à la logique AL, on obtient la logique ACL.

Constructeur	Définition
$\mathcal{O}$	Permet la description de concepts par l'énumération d'individus nommés
$\mathcal{U}$	Désigne l'union de concepts arbitraires
$\mathcal{E}$	désigne la quantification existentielle complète
$\mathcal{C}$	désigne la négation complète
$\mathcal{I}$	désigne les rôles inverses
$\mathcal{H}$	Désigne l'inclusion entre rôles
$\mathcal{F}, \mathcal{Q}, \mathcal{N}$	Sont des variantes de la contrainte de cardinalité sur rôle

Tableau A.3 : Les constructeurs des Logiques de description

Annexe B : Le langage OWL

Présentation du Langage d'Ontologie Web (OWL)

OWL est un langage fondé sur la syntaxe RDF/XML et hérite des travaux de DAML+OIL. OWL introduit l'aspect sémantique qui manque à RDF, et offre, par ses primitives riches, à la machine une capacité d'interprétation plus grande que celle de RDF et RDFS. OWL se compose de trois sous-langages OWL-Lite, OWL-DL et OWL-Full, qui offrent des capacités d'expression croissantes, chacun est une extension par rapport à son prédécesseur plus simple.

Les différents niveaux d'OWL

a. OWL LITE

Ce sous-langage appartient à la famille SHIF et correspond à la version la plus simple du langage OWL. Il permet d'établir une hiérarchie de concepts simples, contraintes simples. OWL-Lite est le sous langage le plus simple, il répond à des besoins de hiérarchie de classification et de fonctionnalités de contraintes simples. Le tableau X ci-dessous présente les différents constructeurs de ce langage.

Catégorie	Constructeurs	Exemple
RDF Schema	Class, rdfs :subClassOf, rdf :Property, rdfs :subPropertyOf, rdfs :domain, rdfs :range, Individual	Personne : subClassOf(Thing) Femme : subClassOf(Personne) Enfant : subClasseOf(Personne)
In(Equalité)	equivalentClass, equivalentProperty, sameAs, differentFrom, AllDifferent, distinctMembers	Fille : equivalentClass(intersectionOf(Femme, Enfant))
Restrictions	onProperty, allValuesFrom, someValuesFrom	Parent : intersectionOf (Personne, restriction(minCardinality(1), onProperty(aEnfant))
Cardinalités (0 ou 1)	minCardinality, maxCardinality, cardinality	
Intersection	intersectionOf	
Propriétés	SymmetricProperty,FunctionalProperty, ObjectProperty, DatatypeProperty, inverseOf, TransitiveProperty, InverseFunctionalProperty	aParent : ObjectProperty (Enfant, Personne) aEnfant : inverseOf (aParent)

Tableau B.1 : les constructeurs de OWL Lite

OWL restreint davantage encore les possibilités du langage. Par exemple, OWL Lite exclut les cardinalités autre que 0 et 1. Le langage est plus facile à comprendre pour les utilisateurs et plus facile à mettre en œuvre pour les développeurs. L'inconvénient est que l'expressivité est d'autant plus limitée [Nicolas Brunet, Xuan Truong VU, Le Web sémantique].

b. OWL DL

OWL DL (pour Logique de Description) est un sous-langage d'OWL Full qui restreint la façon dont les constructeurs d'OWL et de RDF peuvent être utilisés. Le langage OWL DL possède un vocabulaire partitionné et un typage explicite. Le tableau X montre la liste des constructeurs ajoutés par OWL DL par rapport à OWL Lite.

Catégorie	Constructeurs	Exemple
Axiome de Classe	oneOf (enumération), dataRange, disjointWith	Gender : oneOf(Male, Female)
Expressions booléennes	unionOf, complementOf	Tante : intersectionOf (Femme, unionOf (aNeuve, aNiece))
Cardinalité (0, n)	minCardinality, maxCardinality, cardinality	
Individu cible d'une propriété	hasValue	Homme : intersectionOf (Personne, hasValue(sexe,Male))

Tableau B.2 : les constructeurs de OWL DL

Un document RDF doit en général être étendu à certains égards, et limité dans d'autres pour respecter les contraintes des documents OWL DL [Nicolas Brunet, Xuan Truong VU, Le Web sémantique].

c. OWL FULL

L'intégralité du langage OWL est appelé OWL Full. Il utilise toutes les primitives d'OWL. Il permet également la combinaison de ces primitives de manière arbitraire avec RDF et RDF Schéma. Cela inclut la possibilité de changer le sens de la primitive prédéfinie en appliquant les primitives d'un langage à un autre. Par exemple, dans OWL Full, on pourrait imposer une contrainte de cardinalité sur la classe mère de toutes les classes, pour limiter le nombre de classes qui peuvent être décrites dans une ontologie. L'avantage d'OWL Full est qu'il garanti une compatibilité ascendante avec RDF, à la fois syntaxiquement et sémantiquement. Ainsi, tout document valide RDF est également valide OWL Full. L'inconvénient d'OWL Full est que le langage est devenu si puissant qu'il est indécidable [Nicolas Brunet, Xuan Truong VU, Le Web sémantique].

Annexe C : Outil d'aide d'ontologie Protégé

Présentation de Protégé

Protégé est un outil libre et open source développé par l'université de Stanford. A l'origine, il a été développé pour le domaine biomédical et de la médecine, mais aujourd'hui son utilité à surpassé ce domaine pour être un éditeur d'ontologie. Il permet de représenter le vocabulaire du domaine étudié par l'intermédiaire d'une ontologie. Le logiciel Protégé supporte les langages RDF, XML et OWL

Présentation de l'interface de Protégé

Le choix de Protégé comme outil d'édition d'ontologie, s'est fait à cause de sa simplicité dans la classification des classes, relations et attributs. Il permet aussi la génération d'ontologie en fichier OWL en lui intégrant des règles d'inférences exploitable à partir d'interface graphique (Web). Pour notre travail, on a opté pour la version 3.5 de Protégé.

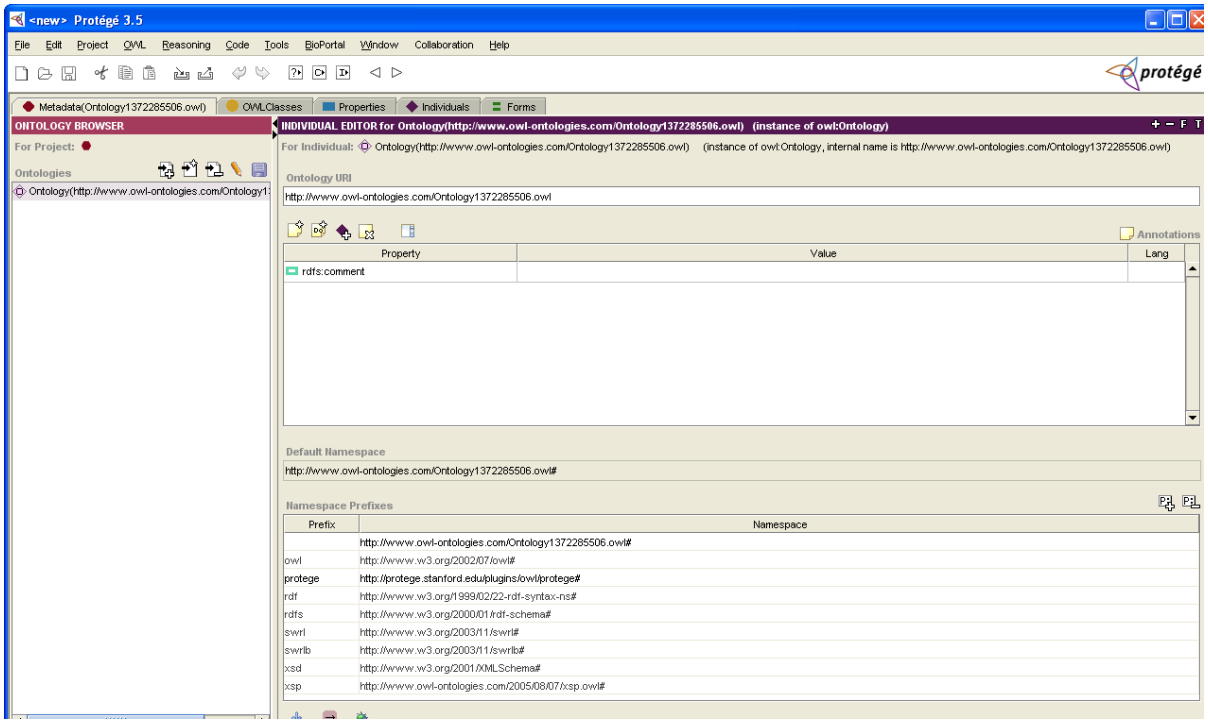


Figure C.1 : interface de Protégé 3.5

Cette page contient par défaut huit (8) onglets dont chacun effectue une tache spécifique :

- **Active Ontology :** affiche les informations sur l'ontologie en cours.
- **Entities :** affiche des informations sur les classes et les propriétés de l'ontologie.
- **Classes :** permet la création de classes et permet aussi de gérer une arborescence de classes ainsi que d'en créer de nouvelles.

- **Object Properties** : pour la gestion des relations entre classes (relation entre concepts). Cet onglet permet aussi de décrire de nouvelles relations entre classes.
- **Data Properties :** défini les relations entre les classes et les concepts simples. Il défini les attributs.
- **Individuals :** pour la définition et la gestion de toutes les instances.
- **OWLViz :** permet une visualisation graphique l'ontologie en cours.
- **DL Query :** autorise l'utilisateur d'interroger l'ontologie.

Création d'une classe

Pour créer une classe, on doit au préalable être dans l'onglet **Classes**. Ensuite sélectionner la classe mère **Thing** dans la fenêtre **Class hierarchy** et passer à la **« création de sous-classe ».**

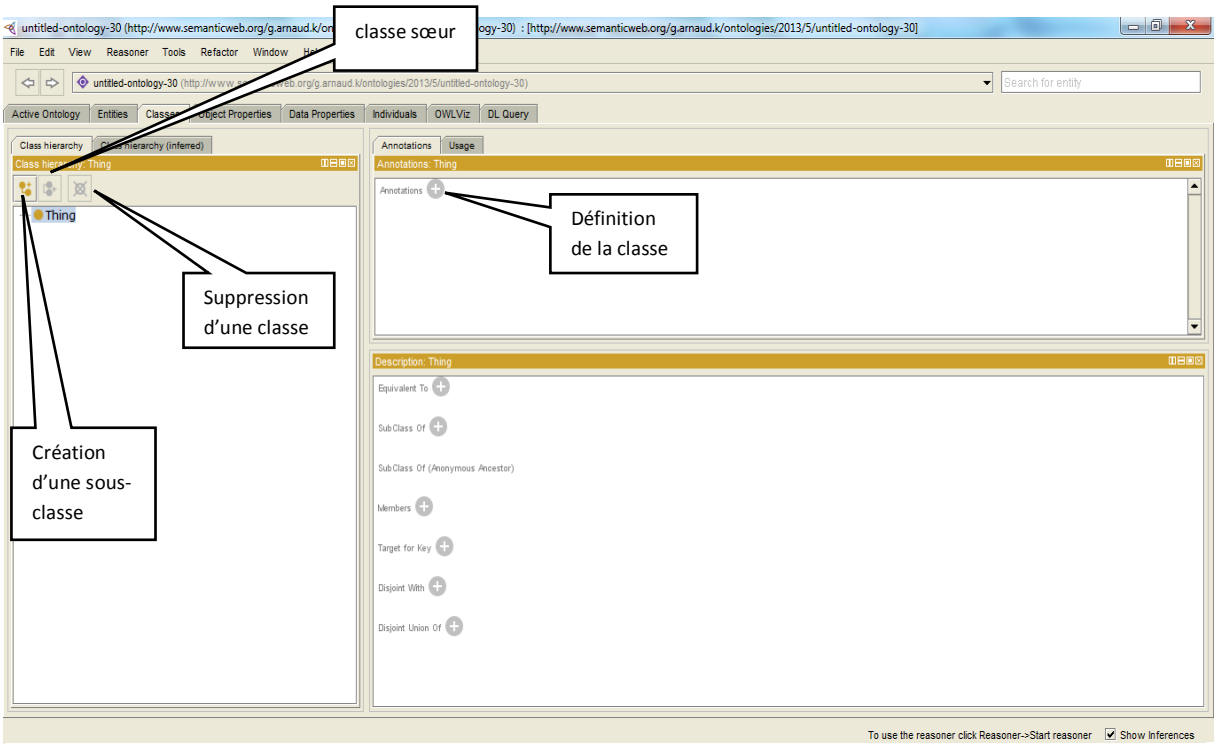


Figure C.2 : Onglet permettant la création d'une classe

Création d'une relation

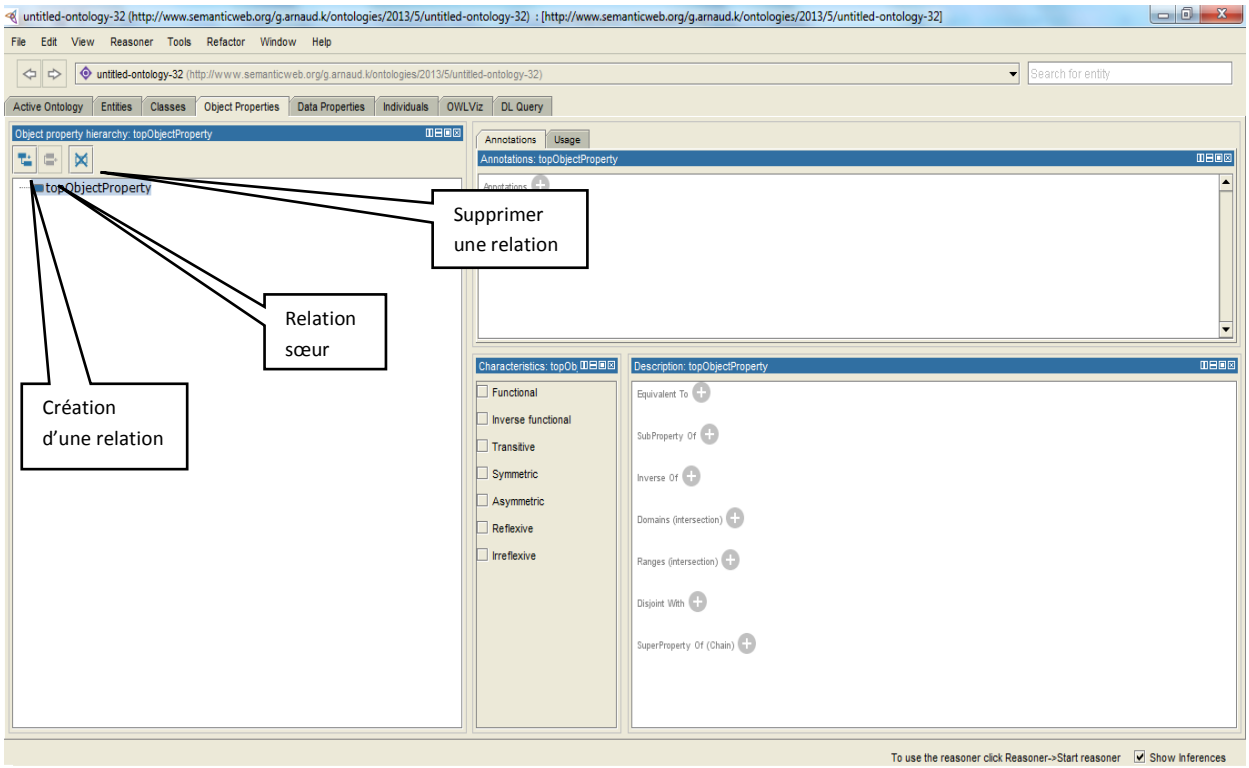


Figure C.3 : Onglet permettant la création d'une instance

En ce qui concerne la relation de composition, elle est représentée par la relation **PartOf**. On a la possibilité de définir la composition dans Classes et lui associer la relation PartOf dans Object Properties en spécifiant la classe à laquelle elle est rattachée.

Création d'une instance ou individu

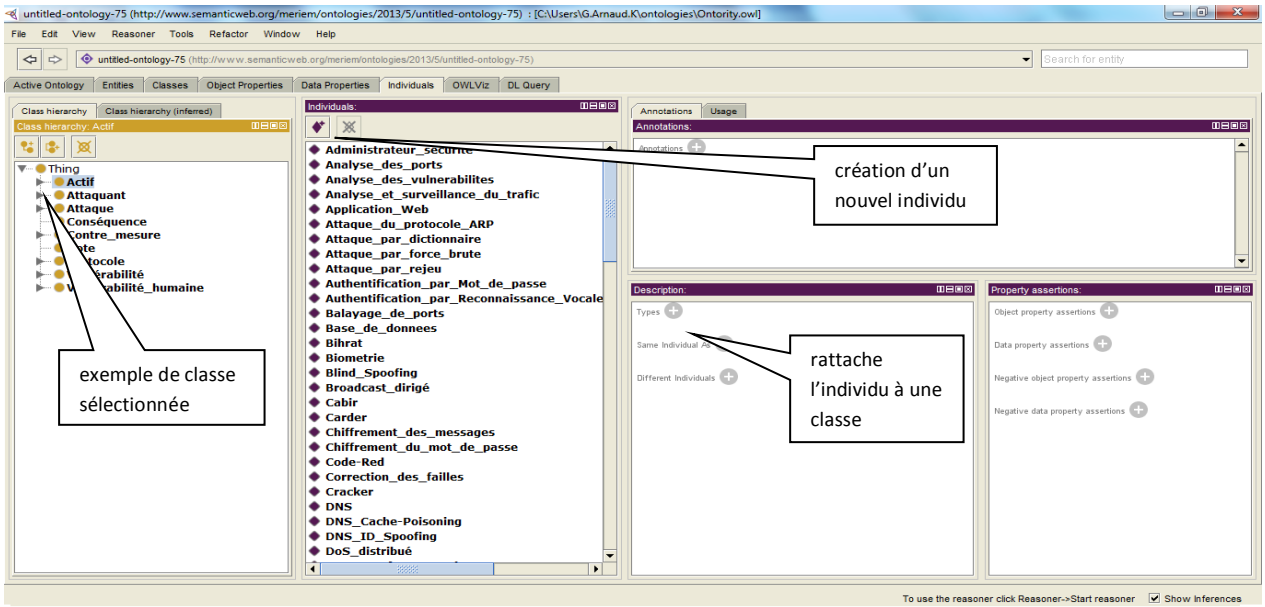


Figure C.4 : création d'une instance.

Visualisation de l'ontologie

La visualisation du graphe de l'ontologie s'effectue grâce au plugin OWLViz qu'il faut charger dans Protégé. Pour cela, il faudrait le télécharger à partir de http://www.graphviz.org**.** Après l'installation du plugin, on passe à la configuration dans Protégé :

- allez à Fichier -> Préférences -> OWL Viz
- défini le point Chemin de l'application dans dot.exe dans le dossier que Graphviz a été installé (exemple: C:\Program Files\Graphviz2.31\bin\dot.exe)

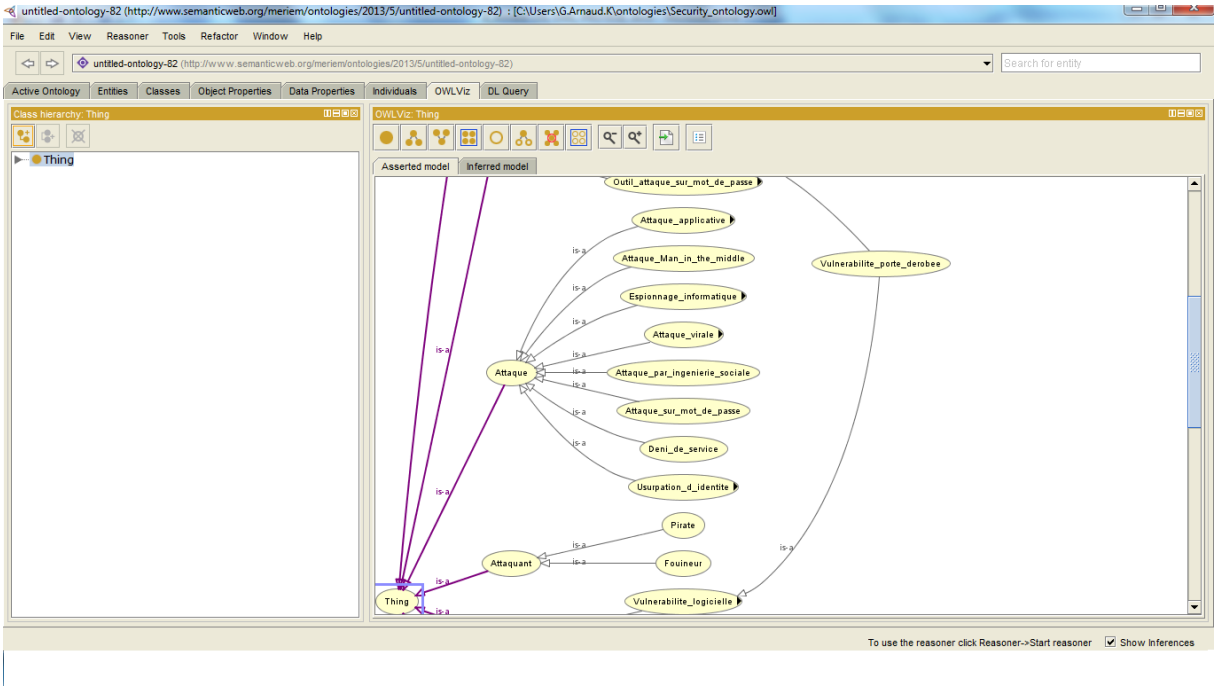
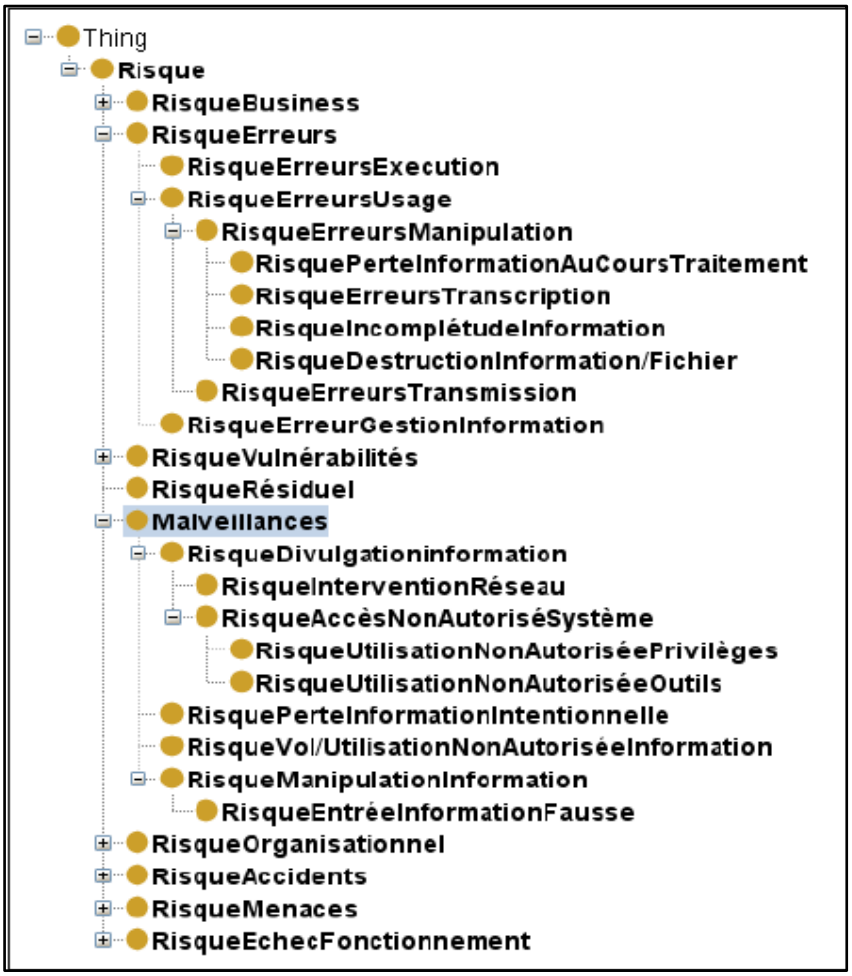


Figure C.5 : graphe de notre ontologie.

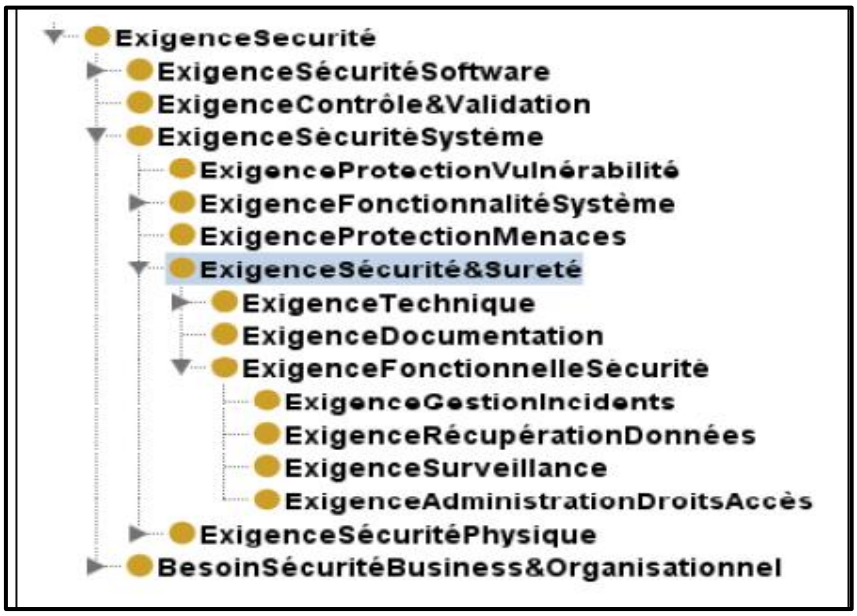
Pour visualiser le graphe, il faut cliquer sur le bouton 'Show Class' et définir la profondeur du graphe que l'on souhaite visualiser (Radius class). **[C. Pierot & JC. Desconnents & T. Libourel, TP Ontologie et Protégé]**

Annexe D : Ontologies abordées en état de l'art

D.1. Extrait de l'ontologie des risques [48]



D.2. Extrait de l'ontologie des exigences de sécurité [48].



D.3. Exemple de liste, de taxonomie et d'ontologie à partir de la littérature abordée par Benali Fatiha [49]. tiré de [53].

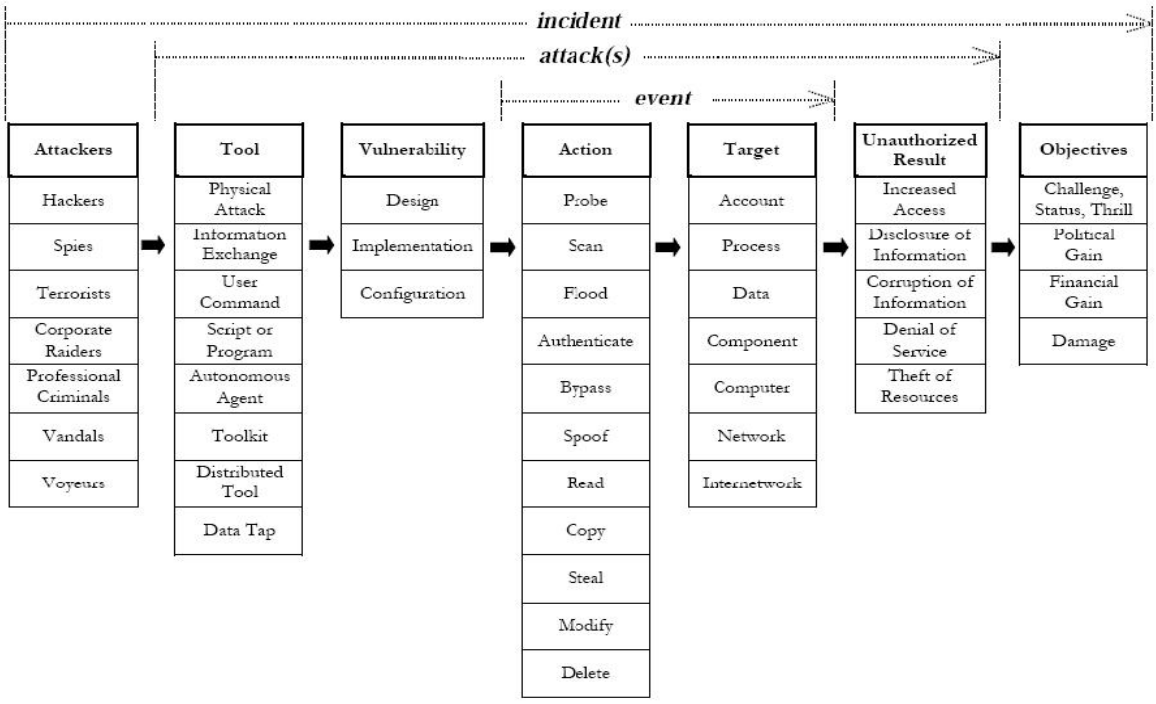


Figure : *Taxonomie d'Howard et Longstaff* **[50]**

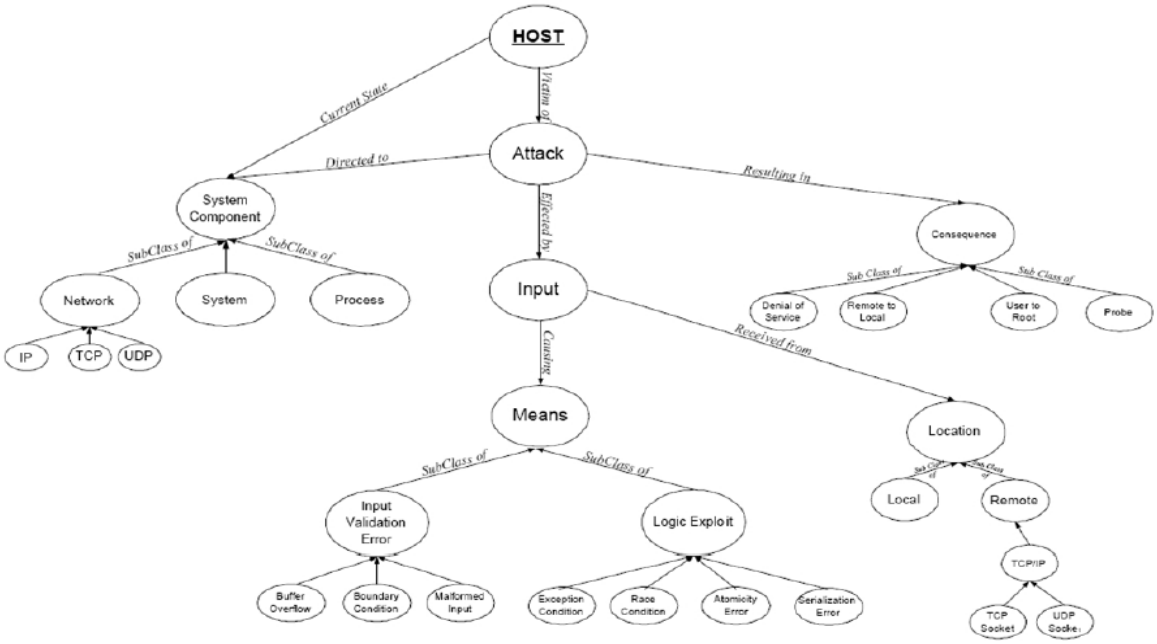


Figure : *Ontologie développée dans* **[51]**

D.4. Diagramme de classification des concepts présenté dans [31]

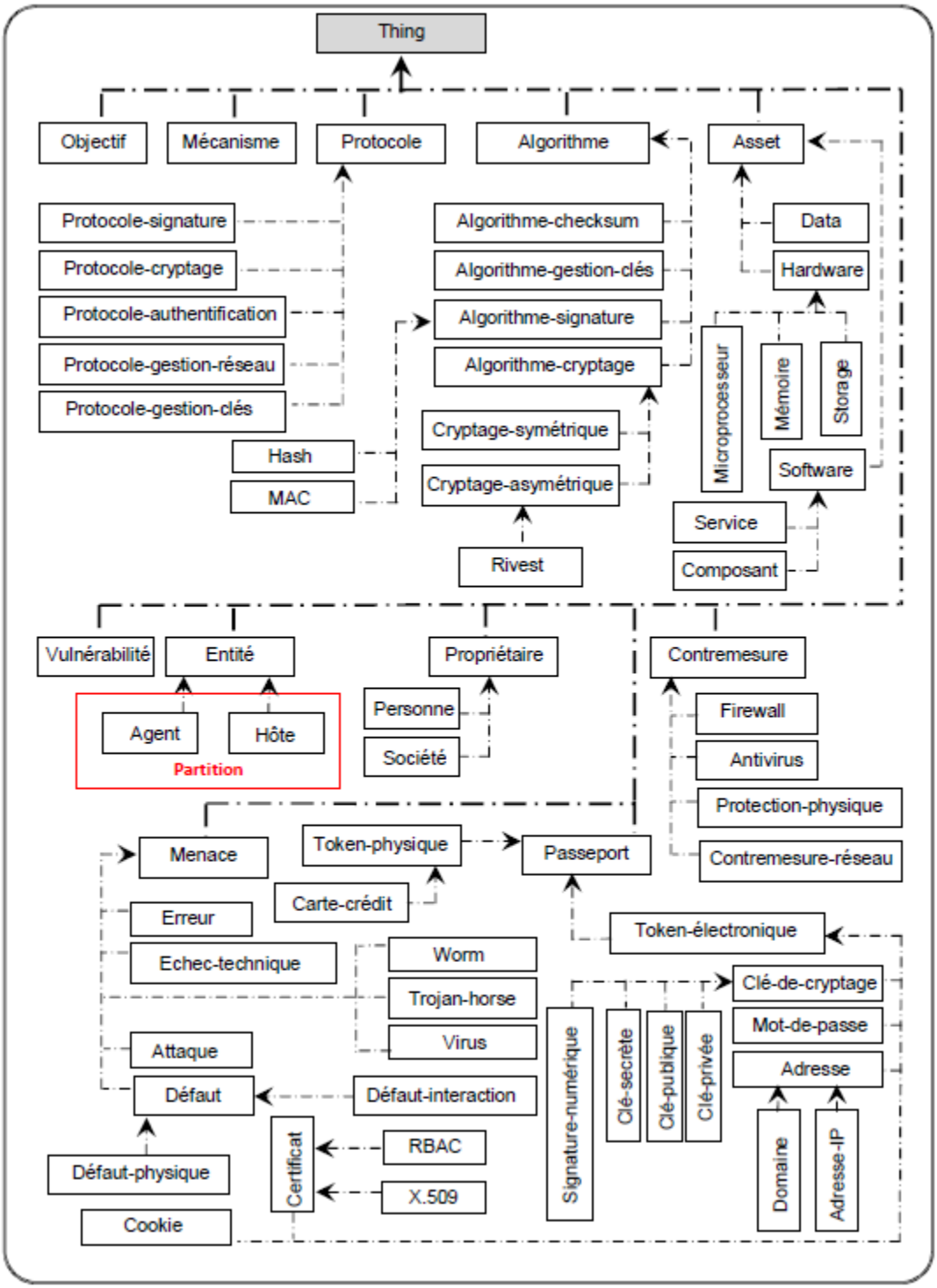


Figure : *Diagramme de classification des concepts* **[31]**

D.5. Diagramme de classification des relations binaires présenté dans [31]

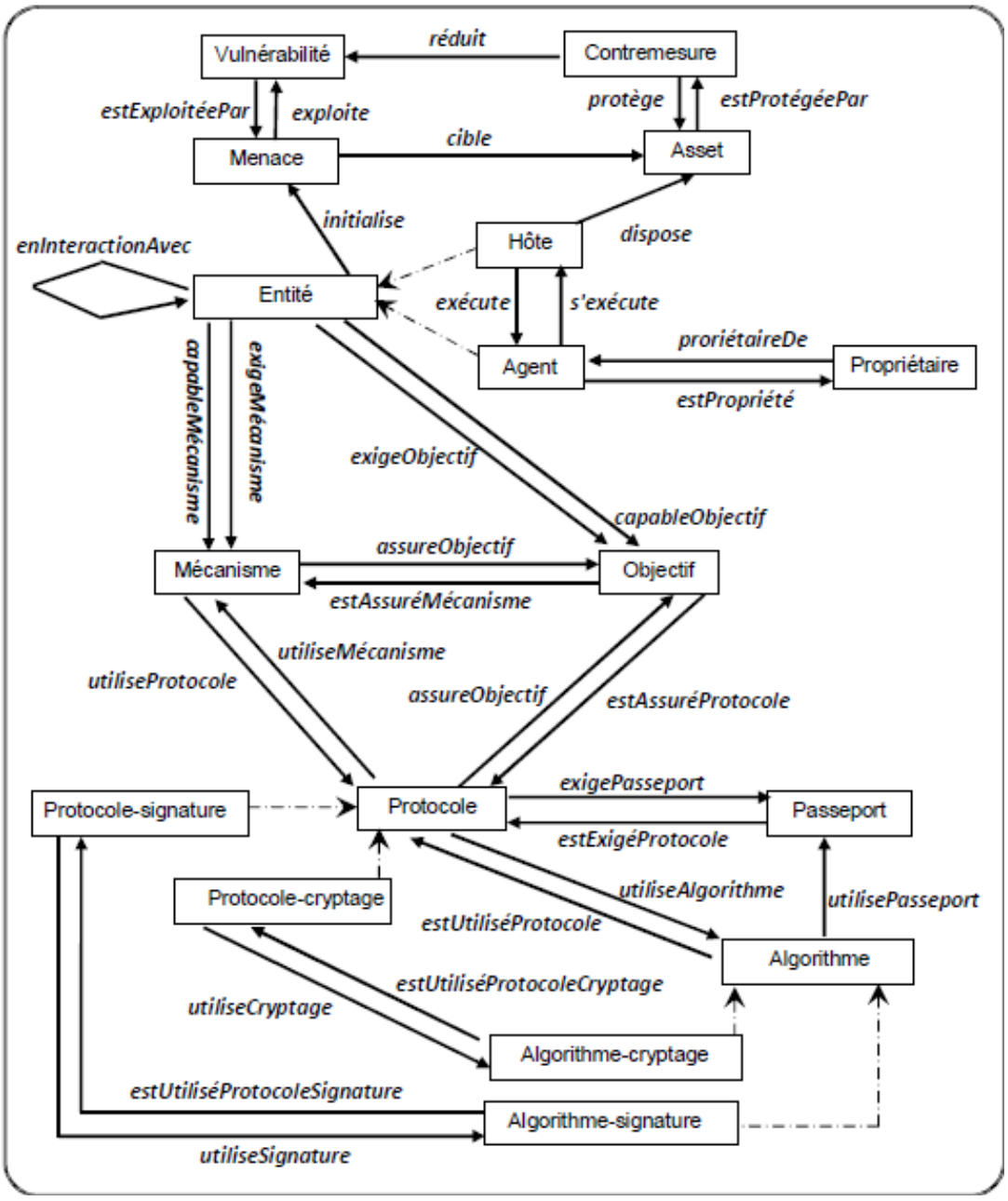


Figure : *Diagramme de classification des concepts* **[31]**