

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTRE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE MOHAMED BOUDIAF - M'SILA

FACULTE DES MATHÉMATIQUES ET
D'INFORMATIQUE

DEPARTEMENT D'INFORMATIQUE

N° :



DOMAINE : MATHÉMATIQUES ET
D'INFORMATIQUE

FILIERE : INFORMATIQUE

OPTION : SYSTEM D'INFORMATION

ET GENIE LOGICIEL

**Mémoire présenté pour l'obtention
Du diplôme de Master Académique**

Par : LADJENEF Mohamed

Intitulé

**ystème à base d'ontologie pour l'aide à la
maintenance des machines**

Soutenu devant le jury composé de :

Lakehal Meftah

Université de M'sila

Président

Mr BOUGHERARA Seddik

Université de M'sila

Rapporteur

Boucetta Mohammed

Université de M'sila

Examineur

Année universitaire : 2017 /2018

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTRE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE MOHAMED BOUDIAF - M'SILA

FACULTE DES MATHÉMATIQUES ET
D'INFORMATIQUE

DEPARTEMENT D'INFORMATIQUE

N° :



DOMAINE : MATHÉMATIQUES ET
D'INFORMATIQUE

FILIERE : INFORMATIQUE

OPTION : SYSTEM D'INFORMATION

ET GENIE LOGICIEL

**Mémoire présenté pour l'obtention
Du diplôme de Master Académique**

Par : LADJENEF Mohamed

Intitulé

**ystème à base d'ontologie pour l'aide à la
maintenance des machines**

Soutenu devant le jury composé de :

Lakehal Meftah

Université de M'sila

Président

Mr BOUGHERARA Seddik

Université de M'sila

Rapporteur

Boucetta Mohammed

Université de M'sila

Examineur

Année universitaire : 2017 /2018

DÉDICACES

Nous voila à la fin d'un parcours si long mais plein de bons événements, de meilleurs souvenirs et d'inoubliables amitiés tissées tout au long de nos années d'études ;

Ce mémoire n'est pas seulement la finalité d'une année d'étude mais le résultat de tant d'années de travail et de recherche du savoir ;

Cet humble travail que nous allons présenter ci-après, n'a pu avoir lieu si tout ceux qui me sont chers ne se pas consacrés et venus à mon aide.

Je dédier mon modeste travail à :

Ma famille : Mon père Ben Benazouz et ma mère, qui mon jamais laissé sentir de manque et qui me ont pris soin de moi depuis ma naissance.

Mes Amis : Messaoud, Iakheddar, Fatah, Rahim, Amine, Taher, Rabeh, Mohamed ali, Khalil, Yassine, Saleh, Hakim, Younes.....la liste est grande.

C'est pour cela que je leur dédie mon travail en leur disant :

« **Je vous aime et merci beaucoup pour tout ce que tout ce que vous m'avez offert** »

MOHAMED

REMERCIEMENTS

Tout d'abord nous remercions notre bon Dieu le tout puissant, pour son aide et pour nous avoir donné le courage et la patience afin d'accomplir notre travail dans les meilleures conditions.

Nous tenons aussi à remercier notre promoteur

Mr BOUGHERARA Seddik pour son suivi et ses précieux conseils durant l'évolution de ce travail.

Nous remercions qui vont également :

A tous nos enseignants qui ont contribué à notre formation, ainsi que tous les personnels de Collège.

A toute notre promotion pour tous les bons moments qu'on a passés ensemble.

➤ Ladjenef Mohamed

TABLE DES MATIERS

Liste des figures.....	IX
Liste des tableaux.....	XI
Liste des abréviations.....	XII
INTRODUCTION GÉNÉRALE	1
CHAPITRE 1 : LA MAINTENANCE DES MACHINES	
Introduction.....	3
1 Définitions.....	3
1.1 la machine.....	3
1.2 La fonction.....	3
1.3 La défaillance.....	3
1.4 La Panne.....	4
1.5 Composant.....	4
1.6 l'analyse des modes de défaillance de leurs effets et de leur criticité (AMDEC) ..	4
1.7 Définition de la maintenance.....	4
2 Risques liés aux machines.....	4
3 Les objectifs de la maintenance.....	5
4 Les différents types de maintenance.....	5
4.1 La maintenance corrective	6
4.2 La maintenance préventive	7
5 Les avantages et les inconvénients de maintenance.....	7
5.1 Les avantage de la maintenance	7
5.2 Les inconvénient de la maintenance.....	7
6 Le rôle de la maintenance.....	7
7 But de la maintenance.....	8
8 Qu'est ce qu'un système expert?	8
8.1 Les composants de système expert.....	8

8.2 Les avantages de système expert.....	8
8.3 Domaine d'application.....	8
10 Conclusion	9
CHAPITRE 2 : LE WEB SEMANTIQUE ET L'ONTOLOGIE	
Introduction.....	10
1 le web sémantique.....	10
2 Architecture du web sémantique.....	10
3 La notion d'ontologie.....	11
3.1 définition.....	11
3.1.1 Définition En informatique	12
3.1.2 Définition en philosophie.....	13
3.2 Apparition d'ontologie.....	13
3.3 Rôle de l'ontologie	13
3.4 Les composants d'ontologie	13
3.4.1 Les concepts	14
3.4.2 Les relations.....	14
3.4.3 Les fonctions.....	15
3.4.4 Les axiomes (ou Règles).....	15
3.4.5 Les instances (ou individus).....	15
3.5 Classification d'ontologies.....	15
3.5.1 Classification Selon l'objet de conceptualisation.....	15
3.5.1.1 Ontologie de Représentation des connaissances.....	16
3.5.1.2 Ontologie Supérieure ou de Haut niveau.....	16
3.5.1.3 Les ontologies génériques	16
3.5.1.4 Les ontologies des tâches (Mizoguchi, 2003)	16
3.5.1.5 Les ontologies de domaine.....	16
3.5.1.6 Les ontologies de tâches-domaine.....	17

3.5.1.7 Les ontologies d'application.....	17
3.5.2 Classification selon le formalisme utilisé.....	17
3.5.3 Classification Selon le niveau de détail de l'ontologie	17
4 Conclusion.....	18
CHAPITRE 3: L'INGENIERIE ONTOLOGIQUE	
Introduction.....	19
1 Construction des ontologies.....	19
1.1 Critères d'évaluation d'une ontologie.....	19
1.2 Le processus de construction d'une ontologie.....	20
1.2.1 La conceptualisation.....	20
1.2.2 L'ontologisation.....	20
1.2.3 l'opérationnalisation.....	20
2 Cycle de vie d'une ontologie.....	20
2.1 Conception	21
2.1.1 Conceptualisation	21
2.1.2 Ontologisation.....	22
2.1.3 Opérationnalisation.....	22
2.2 Spécification des besoins.....	22
2.3 Diffusion et cas d'utilisation de l'ontologie.....	22
2.4 Utilisation.....	23
2.5 Evaluation d'une ontologie.....	23
3 Quelques méthodologies de construction d'ontologie.....	23
3.1 La méthode METHONTOLOGY.....	23
3.1.1 Spécification.....	24
3.1.2 Conceptualisation.....	24
3.1.3 Formalisation.....	27
3.1.4 Implémentation.....	27

3.1.5 Maintenance.....	27
3.2 La méthode Tove.....	27
3.3 La méthode ENTREPRISE.....	27
4 Outils de développement d'ontologie.....	28
4.1 Les langages de spécification d'ontologies.....	28
4.2 Langages d'interrogation d'ontologie	31
4.2.1 utilisation du protocole de requête SPARQL.....	31
4.2.2 Le langage SWRL	31
4.3 Les moteurs d'inférence.....	32
4.3.1 Racer.....	33
4.3.2 Pellet.....	34
4.4 Les éditeurs d'ontologie.....	34
4.4.1 ONTOLINGUA.....	34
4.4.2 Protégé.....	35
4.4.3 WEBONTO.....	35
4.4.4 SWOOP.....	35
4.4.5 ONTOEDIT.....	35
5 Conclusion.....	36
CHAPITRE 4: CONCEPTION ET RÉALISATION	
Introduction.....	37
1 Choix de la méthode de construction de notre ontologie.....	37
1.1 Spécification.....	37
1.1.1 Diagramme de cas d'utilisation.....	38
1.1.2 Les diagrammes de séquence.....	38
1.1.2.1 diagramme de séquence ajouter une machine.....	39
1.1.2.2 diagramme de séquence ajouter un maintenance.....	39
1.1.2.3 diagramme de séquence ajouter une panne.....	40

1.1.2.4 diagramme de séquence ajouter une défaillance.....	40
1.1.2.5 diagramme de séquence ajouter une analyse.....	41
1.1.2.6 diagramme de séquence modifier une machine.....	41
1.1.2.7 diagramme de séquence modifier une maintenance.....	42
1.1.2.8 diagramme de séquence modifier Une panne.....	42
1.1.2.9 diagramme de séquence modifier une défaillance.....	42
1.1.2.10 diagramme de séquence modifier une d'analyse.....	43
1.1.2.11 diagramme de séquence recherche une machine.....	44
1.1.2.12 diagramme de séquence recherche une maintenance.....	44
1.1.2.13 diagramme de séquence recherche une panne.....	45
1.1.2.14 diagramme de séquence recherche une défaillance.....	45
1.1.2.15 diagramme de séquence recherche Une d'analyse.....	46
1.2 Conceptualisation.....	46
1.2.1 Construction du glossaire de termes.....	46
1.2.2 Détermination de la taxonomie de concepts.....	47
1.2.3 Construction d'un diagramme des relations binaires.....	48
1.2.4 Construction d'un dictionnaire des concepts.....	49
1.2.5 Construction du tableau des relations binaires.....	49
1.2.6 Construction du tableau des attributs d'instances.....	49
1.2.7 Construction du tableau des attributs de classe.....	50
1.3 Table d'analyse des panne méthode (AMDEC).....	50
1.4 La formalisation.....	50
1.5 Implémentation.....	52
1.5.1 Langages de programmation.....	52
1.5.1.1 Jena.....	52
1.5.1.2 Java.....	53
1.5.2 Outils de programmation.....	54

1.5.2.1 Protégé.....	54
1.5.2.2 NetBeans.....	57
1.5.3 Langages web sémantique.....	57
1.5.3.1 OWL.....	57
1.5.3.2 SPARQL.....	59
1.5.3.3 SWRL-Tab.....	59
1.6 Maintenance.....	59
2 L'interface de l'application.....	59
3 Conclusion.....	60
CONCLUSION GÉNÉRALE.....	61
Bibliographies.....	62

LISTE DES FIGURES

Figure 1.1 Organigramme types de maintenance.....	6
Figure 2.1 XML, RDF et OWL : les 3 couches du WS	10
Figure 2.2 les couches du web sémantique.....	11
Figure 2.3 Hiérarchie de concepts d'une ontologie pour les recettes de cuisine.....	12
Figure 2.4 Classification des ontologies Selon l'objet de conceptualisation.....	17
Figure 3.1 Processus de construction d'ontologie.....	20
Figure 3.2 Le cycle de vie d'une ontologie.....	21
Figure 3.3 Processus de construction de l'ontologie selon METHONTOLOGY.....	24
Figure 3.4 Processus de développement d'une ontologie.....	25
Figure 3.5 La pyramide des langages d'ontologies basés Web.[26].....	28
Figure 4.1 Diagramme de cas d'utilisation.....	38
Figure 4.2 ajouter une machine.....	39
Figure 4.3 ajouter une maintenance.....	39
Figure 4.4 ajouter une panne.....	40
Figure 4.5 ajouter une défaillance.....	40
Figure 4.6 ajouter une analyse.....	41
Figure 4.7 modifier une machine.....	41
Figure 4.8 modifier une maintenance.....	42
Figure 4.9 modifier une panne.....	42
Figure 4.10 modifier une défaillance.....	43
Figure 4.11 modifier une d'analyse.....	43
Figure 4.12 recherche une machine.....	44
Figure 4.13 recherche une maintenance.....	44
Figure 4.14 recherche Une panne.....	45
Figure 4.15 recherche une défaillance.....	45

Figure 4.16 recherche Une d'analyse.....	46
Figure 4.17 Diagramme de classe.....	47
Figure 4.18 Diagramme de relation binaire.....	48
Figure 4.19 l'interface de JENA.....	53
Figure 4.20 l'interface de NetBeans.....	54
Figure 4.21 Interface graphique de Protégé 5.2.0.....	55
Figure 4.22 Création des classes.....	55
Figure 4.23 Création d'une relation.....	56
Figure 4.24 Création des instances.....	56
Figure 4.25 exemple OWL.....	58
Figure 4.25 une partie d'application	60

LISTE DES TABLEAUX

Tableau 3.1 Tableau comparatif des principaux moteurs d'inférence pour LD.[10].....	33
Table 4.1 Glossaire de termes.....	52
Table 4.2 dictionnaire des concepts.....	54
Table 4.3 tableau des relations binaires.....	54
Table 4.4 tableau des attributs d'instances.....	55
Table 4.5 tableau des attributs de classe.....	55
Table 4.6 analyse des pannes méthode (AMDEC).....	55

LISTE DES ABBREVIATIONS

1. DAML : Darapa Agent Markup Language
2. DAML+OIL : Darapa Agent Markup Language OIL
3. DTD : Document Type Définition
4. GL : Génie Logiciel
5. AMDEC : Analyse des modes de défaillance de leurs effets et de leur criticité
6. IA : Intelligence Artificielle
7. AFNOR : Association Française de Normalisation
8. IDE : Integrated Development Environment
9. JVM : Java Virtual Machine
10. LDs : Logique des Descriptions
11. nRQL : new Racer pro Query Language
12. OIL : Ontology Inference layer
13. OML : Ontology Markup Language
14. OWL : Ontology Web Language
15. OWL – DL : Ontology Web Language Description Logique
16. RDF : Resource Description Framework
17. RDF(S) : Resource Description Framework Schema
18. RDQL : RDF Data Query Language
19. SAD : Système d'Aide à la Décision
20. SADM : Système d'Aide à la Décision Médicale
21. SBC : Système à Base de Connaissance
22. SE : Système Expert
23. SGML : Standard General Markup language
24. SHOE : Simple HTML Ontology Extensions
25. SPARQL : SPARQL Protocol And RDF Query Language
26. SWRL : Semantic Web Rule Language

27. TOVE : Toronto Virtual Enterprise
28. UML : Unified Modeling Language
29. URI : Uniform Resource Identifier
30. W3C : World Wide Web Construction
31. XML : eXtensible Markup Language
32. XOL : XML – based Ontology exchange Language

INTRODUCTION GÉNÉRALE

La machine est considérée l'une des bases dont nous avons besoin dans notre vie quotidienne, c'est pour cela, nous devons accorder une grande importance dans divers domaines de la maintenance et d'autres mesures à prendre pour que nous rassurons son bon fonctionnement dans la production, que nous espérons obtenir. La maintenance de la machine est la démarche par laquelle le mécanicien va déterminer la panne de la machine, et qui va également permettre de proposer un traitement.

L'étude ontologie des études les plus importantes qui accompagnent la vision humaine de l'importance de lui-même imposé pour fournir une pénurie de besoins humains et caractérisé par une base de données RDF/XML en termes de maintien de la base de connaissances et de faciliter l'accès aux données et relier les nouvelles données et les choses attendues des règles que nous avons établies. Technologies web sémantiques plus connectées, ouvertes et intelligentes, bases de données distribuées, traitement du langage naturel, apprentissage automatique, raisonnement machine, agents autonomes, microformats, recherche en langage naturel, exploration de données, agents de recommandation, technologies d'intelligence artificielle, informations ... Bienvenue sur le Web intelligent.[31]

Les ontologies sont utilisées pour construire des bases de connaissances. Une base de connaissances peut être formée par une ontologie et un ensemble d'instances de ces classes.[20]

Ces dernières années avec l'avènement du web sémantique, l'utilisation des ontologies comme formalisme de représentation des connaissances a eu une progression significative dans de nombreux domaines, dont la maintenance de la machine.

Le système de modélisation ontologie, nous avons besoin du langage UML commun à utiliser la méthode METHONTOLOGY.

Notre contribution dans ce contexte consiste en la construction et l'interrogation d'une ontologie au sujet des publications scientifiques . Le développement de cette ontologie doit suivre un processus de construction d'ontologies comptant un ensemble de phases spécifiées de façon très détaillée afin de cerner l'étendue et d'aboutir à une ontologie qui répond aux besoins. L'interrogation de cette ontologie est faites par des requêtes formelles SPARQL formulées par les utilisateurs qui sont assistés par un processus d'auto-complétion de requêtes SPARQL.[30]

On peut faire face à beaucoup de problèmes dans notre travail, c'est pour cela nous allons expliquer dans les chapitres suivants comment résoudre les problèmes qui nous confrontent.

Chapitre I : Machine et ses problèmes, la maintenance et ses types, le système expert et ses domaines.

Chapitre II : le web sémantique et ses composantes et les ontologies : des démarches en détail : La notion d'ontologie, son apparition , son rôle , ses composantes et sa classification.

Chapitre III : Explication la méthode METHONTOLOGY : expliquer en détaillant les différentes étapes de chaque méthode puis les outils de développement d'ontologie: les langages de spécification d'ontologie, les moteurs d'inférence, les langages d'interrogation d'ontologie et enfin les différents éditeurs d'ontologies.

Chapitre IV : Représentation de UML et l'application de la méthode METHONTOLOGY sur notre problème et sa représentation dans le protégé et son application au langage de Java NetBeans.

CHAPITRE 1

LA MAINTENANCE DES MACHINE

Introduction

La maintenance est l'un des éléments essentiels qui aide une personne à effectuer de nombreuses tâches différentes dont une personne a besoin pour protéger sa machine contre les dommages et également pour protéger la quantité de production que cette machine fournit. La maintenance est le raisonnement menant à l'identification de la cause d'une défaillance, d'un problème ou d'une panne, à partir des caractères ou défaillance relevés par des observations ou des analyses.

Et aussi contribuer à la maintenance d'éviter les défaillances inattendues de la machine que nous traitons dans ce chapitre de la machine et la maintenance, d'autre part aide le système expert en maintenance.

1 Définitions

1.1 la Machine

Une machine est un ensemble équipé ou destiné à être équipé d'un système d'entraînement autre que la force humaine ou animale appliquée directement, composé de pièces ou d'organes liés entre eux, dont au moins un est en cas de mouvement, et qui sont réunis de façon solidaire en vue d'une application définie.

Est aussi considéré comme une machine un ensemble de machines ou de quasi machines qui, afin de concourir à un même résultat, sont disposées et commandées de manière à être solidaires dans leur fonctionnement.[5]

1.2 La fonction des machine

D'un système est ce à quoi le système est destiné, comme elle est décrite par la spécification fonctionnelle, qui inclut les performances attendues du système. [28]

1.3 La défaillance :

Altération ou cessation de l'aptitude d'une entité à accomplir une fonction requise. Après défaillance d'une entité, celle-ci est en état de panne.

La défaillance est un passage d'une entité d'un état de fonctionnement normal à un état de fonctionnement anormal ou de panne.[28]

1.4 La Panne

Inaptitude d'une entité à accomplir une fonction requise ou à assurer le service approprié à la suite d'une défaillance.

Note : Une panne est généralement la conséquence d'une défaillance ; néanmoins, elle peut exister sans défaillance préalable. C'est l'ensemble des défaillances des composants. La cause supposée d'une panne est un défaut physique ou une erreur humaine.[2]

1.5 Composant

Partie d'un élément fonctionnel constituant une entité élémentaire considérée comme indivisible.[2]

1.6 l'analyse des modes de défaillance de leurs effets et de leur criticité (AMDEC)

l'amdec est un outil qui consiste à prévoir pour ne pas être obligé de revoir. Il existe plusieurs sortes d'AMDEC: l'AMDEC du concept, produit, procédé, moyen, de la machine,

La méthode est très utilisée dans le secteur de l'automobile, de l'aéronautique et du ferroviaire. Une méthode dérivée de l'AMDEC est utilisée dans l'agroalimentaire, la chimie et le secteur pharmaceutique.[6]

1.7 Définition de la maintenance

L'opération de maintenance peut se définir comme étant une suite d'actions organisées, intervenant sur un système et ayant un double objectif.

Premier objectif

Rétablir un bien, en état de dysfonctionnement et le replacer en état de fonctionnement, donc de produire.

Deuxième objectif

Maintenir ce bien, par une suite d'actions préventives et planifiées, en état parfait de fonctionnement, donc de produire. En règle générale, le service maintenance doit garder l'outil de production en état opérationnel, afin d'assurer une production efficace et maximale.

(Bien : machines, système automatisés de production, mécanisme, appareils divers). [10]

2 Risques liés aux machines

Les risques liés aux machines peuvent être de différentes natures. Il peut s'agir de :

- ❖ Risques mécaniques
- ❖ Risques dus aux énergies électrique, hydraulique, pneumatique...
- ❖ Risques dus à l'électricité statique

- ❖ Risques dus aux erreurs de montage
- ❖ Risques thermiques (températures extrêmes)
- ❖ Risques d'incendie et d'explosion
- ❖ Risques dus au bruit
- ❖ Risques dus aux vibrations
- ❖ Risques dus aux rayonnements (ionisants, non-ionisants, laser...)
- ❖ Risques dus aux émissions de matières et substances dangereuses
- ❖ Risques liés à la conception des circuits de commande
- ❖ etc.

Les risques dus aux machines sont présents lors de l'utilisation normale, mais aussi lors de situations particulières telles que le montage/démontage, la maintenance, le réglage, le nettoyage.[5]

3 Les objectifs de la maintenance

L'objectif de la maintenance est de limiter les effets de ces perturbations afin d'atteindre les performances exigées et des actions sont élaborées de manière a :

- ✓ Limiter les indisponibilités,
- ✓ Garantir la qualité des produits et des services,
- ✓ Maîtriser les couts,
- ✓ Protéger les personnes, l'environnement et les biens.[28]

4 Les différents types de maintenance

- ✓ La maintenance corrective
- ✓ La maintenance préventive

La Figure suivant présenter tout les types de la maintenance

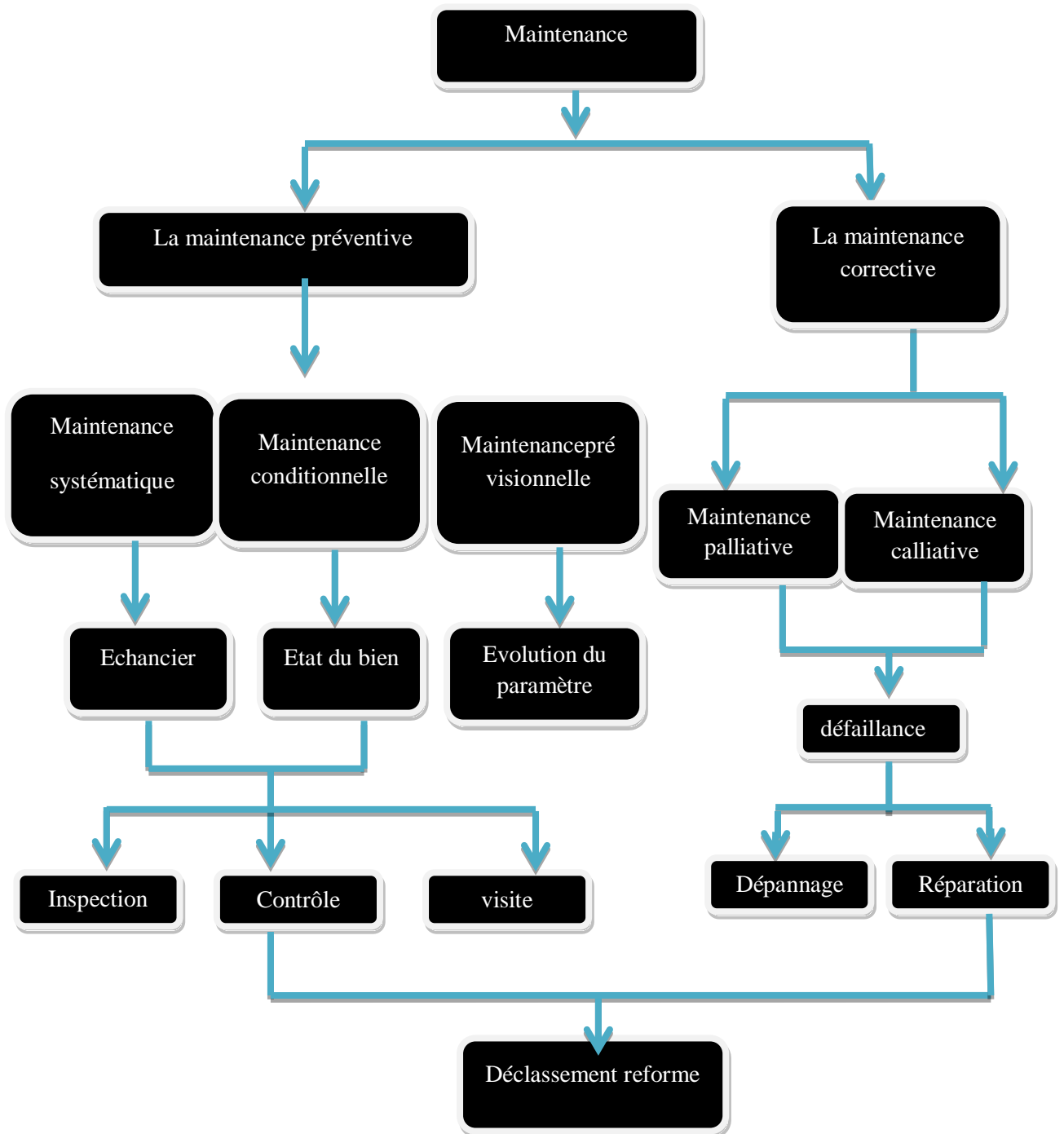


Figure 1.1 Organigramme types de maintenance[9]

4.1 La maintenance corrective

Definition

La norme AFNOR NF X 60 010 définit la maintenance corrective comme une maintenance effectuée après défaillance, où l'on distingue :

Le dépannage : action sur un bien en panne en vue de le remettre en état de fonctionnement. Aussi, dans ce cas, il est indispensable de travailler méthodiquement. Une bonne préparation du travail, la saisie et la gestion de toutes les informations concernant les interventions de maintenance, permettront de terminer le travail en limitant les temps morts et en évitant les erreurs de démontage et de montage.

On envisage deux types d'intervention :

- ✓ Palliative (dépannage), c'est-à-dire une remise en état de fonctionnement «caractère provisoire».
- ✓ Curative (réparation), c'est la réparation complète, parfois après dépannage «caractère définitif».

Cette maintenance est utilisée lorsque l'indisponibilité du matériel n'as pas de conséquences majeures sur le processus de production ou quand les contraintes de sécurité sont faibles.[9]

4.2 La maintenance préventive

Definition

Définition AFNOR (X-60-010): Maintenance effectuée dans l'intention de réduire la probabilité de défaillance d'un bien ou la dégradation d'un service rendu.

Les types de maintenance préventive

- ✓ La maintenance préventive systématique.
- ✓ La maintenance préventive conditionnelle.[9]

5 Les avantages et les inconvénients de maintenance

5.1 Les avantage de la maintenance

C'est une maintenance facile à gérer car les périodes d'interventions sont fixes, elle permet

- ✓ d'éviter les détériorations importantes.
- ✓ de diminuer les risques d'avaries imprévues.[28]

5.2 Les inconvénient de la maintenance

Reposer sur la notion de et ne prends pas en compte les phénomènes d'usure.[28]

6 Le rôle de la maintenance

Le service maintenance doit mettre en œuvre la politique de maintenance définie par la direction de l'entreprise ; cette politique devant permettre d'atteindre le rendement maximal des systèmes de production. Cependant, tous les équipements n'ont pas le même degré d'importance d'un point de vue maintenance. Le service devra donc, dans le cadre de la politique globale, définir les stratégies les mieux adaptées aux diverses situations. La fonction maintenance sera alors amenée à établir des prévisions ciblées :

- ✓ Prévisions à long terme : elles concernent les investissements lourds ou les travaux durables. Ce sont des prévisions qui sont le plus souvent dictées par la politique globale de l'entreprise.
- ✓ Prévisions à moyen terme : la maintenance doit se faire la plus discrète possible dans le planning de charge de la production. Il lui est donc nécessaire d'anticiper, autant que faire se peut, ses interventions en fonction des programmes de production. La production doit elle aussi prendre en compte les impératifs de suivi des matériels.
- ✓ Prévisions à courts termes : elles peuvent être de l'ordre de la semaine, de la journée, voire de quelques heures. Même dans ce cas, avec le souci de perturber le moins possible la production, les interventions devront elles aussi avoir subi un minimum de préparation.[28]

7 But de la maintenance

La maintenance à pour but:

- ✓ Le maintien du capital machine.
- ✓ Minimiser les arrêts et les chutes de production.
- ✓ Améliorer la sécurité de personnel et la protection de l'environnement.[12]

8 Qu'est ce qu'un système expert?

les systèmes experts sont des logiciels peut-être bientôt aussi des matériels destinés à remplacer ou assister l'homme dans des domaines où est reconnue une expertise humaine :

- ✓ insuffisamment structurée pour constituer une méthode de travail précise, sûre, complète, directement transposable sur ordinateur.
- ✓ sujette à révisions ou compléments (selon l'expérience accumulée).

Une telle définition fait apparaître au moins deux conditions pour implanter un système expert (S.E.) :

- ✓ l'absence d'algorithme facilement programmable pour résoudre le problème
- ✓ la nécessité de prévoir une maintenance facile. [1]

8.1 Les composants de système expert

Un SE est fondé sur le principe de séparation des connaissances (la base des connaissances) et les programmes qui les traitent (le moteur d'inférence).[3]

8.2 Les avantages de système expert

- ✓ Modification de la BC sans besoin de toucher aux programmes .
- ✓ Possibilité de réutilisation du même moteur d'inférence avec d'autres .[3]

8.3 Domaine d'application

1. Mathématiques :

MacSyma (1967) : résolution des problèmes mathématiques (intégrales, différentielles, systèmes d'équations,...).

2. Géologie :

Prospector : pour l'exploration des gisements minéraux

3. Chimie :

Dendral (1969), Métadendral (1978) : identification de la structure moléculaire d'un corps inconnu.

4. Médecine :

Mycine (1977): détection des maladies infectieuses.

5. Linguistique :

Hearsay II : compréhension de la parole.

6. Informatique :

Xcon : pour la configuration des systèmes informatique Cyc : Outil pour la recherche documentaire intelligente.

7. Autres domaines et utilisations :

- Aide à la FAO.
- Commande de robots.
- Maintenance (diagnostic de pannes et anomalies).
- Choix de produits dans une ligne de production.
- Planification et ordonnancement.
- Contrôle de qualité.
- Contrôle d'un atelier de production en temps réel.
- Finances.
- Marketing.
- Ressources humaines.[3]

9 Conclusion

Dans ce chapitre, nous avons présenté l'importance de la maintenance, ses types et les avantages et les inconvénients, et le rôle, et sa relation avec la machine.

CHAPITRE 2

LE WEB SEMANTIQUE ET L'ONTOLOGIE

Introduction

Le Web sémantique nous permet d'atteindre des choses que nous n'avons pas pu atteindre avant, notamment avec l'avènement de l'ontologie, car l'ontologie utilise les technologies web et représente la base de connaissances et l'accès à une description informationnelle par l'utilisation du formalisme qui organise les connaissances sous forme d'ontologie.

Dans ce chapitre, nous présentons le Web sémantique et ses composantes, ainsi que l'ontologie science de l'existence, ses composantes, son importance, son but et sa classification.

1 le web sémantique

Le Web sémantique intègre une communauté d'agents capables d'échanger des données et des services des sources différentes afin d'atteindre un but spécifique. Plus précisément, les services Web sont des composants logiciels réutilisables qui implémentent une fonctionnalité accessible sur le Web.

De plus, la description et le contenu des documents sont définis précisément grâce à une série de méta données.[13]

2 Architecture du web sémantique

La vision courante du Web sémantique proposée par peut être représentée dans une architecture en plusieurs couches différentes (Figure 2.2), dont les trois premières sont considérées comme couche de base (Figure 2.1).

Chaque niveau repose sur les résultats définis au niveau inférieur, c'est-à-dire que chaque niveau est progressivement plus spécialisé et plus complexe que le niveau précédent.



Figure 2.1 XML, RDF et OWL: les 3 couches du WS [14]

Les couches inférieures sont stabilisées à l'heure actuelle. Par contre, la faisabilité des couches supérieures ne semble toujours pas claire.

Les travaux visant la réalisation du web sémantique se situent à des niveaux de complexité très différents. Les plus simples utilisent des jeux plus ou moins réduits de métadonnées dans un contexte de recherche d'information ou pour adapter la présentation des informations aux

utilisateurs. Dans ce cas, des langages de représentations simples sont suffisants. Dans les travaux plus complexes mettant en œuvre des architectures sophistiquées, pour permettre par exemple l'exploitation de ressources hétérogènes, des langages plus expressifs et plus formels issus des travaux de représentations et en ingénierie des connaissances, sont nécessaires. Dernièrement, le Langage SWRL pour le niveau logique a été soumis au W3C pour devenir une recommandation au même titre que les autres langages.

Grâce aux efforts établis pour réaliser la vision du Web sémantique et au grand consensus dans la communauté d'inclure les règles avec l'ontologie dans l'architecture du Web sémantique, un diagramme de couche plus raffiné a résulté, ce diagramme est montré dans la partie (b) de la Figure 2.2. [14]

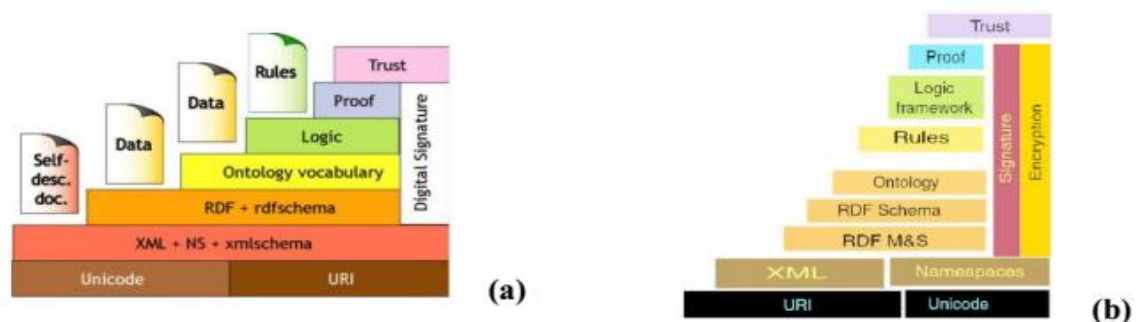


Figure 2.2 les couches du web sémantique [14]

3 La notion d'ontologie

Le terme « ontologie » provient du domaine de la philosophie et est défini comme la théorie de l'être ou des concepts généraux de l'être. Bien que cette notion trouve ses origines dans le domaine de la philosophie, elle a aussi été adoptée dans les domaines de l'intelligence artificielle, du web sémantique, du génie logiciel, des mathématiques, etc. En informatique, ontologie est un terme technique qui dénote un artefact conçu dans le but de pouvoir modéliser des connaissances d'un domaine réel ou imaginaire. L'un des premiers objectifs du développement d'une ontologie est de partager une même compréhension de la structure d'une information entre des personnes. On distingue aussi, entre autres, les raisons suivantes pour lesquelles il est nécessaire de développer une ontologie:

- ✓ Permettre la réutilisation des connaissances d'un domaine.
- ✓ Rendre explicites les hypothèses d'un domaine.
- ✓ Séparer les connaissances du domaine des connaissances opérationnelles.
- ✓ Analyser les connaissances d'un domaine.
- ✓ Faciliter l'interopérabilité entre deux systèmes.[22]

3.1 définition

Une ontologie, selon la définition la plus répandue en informatique, est « la spécification explicite d'une conceptualisation ». Plus concrètement, il est habituel de considérer une ontologie comme la définition formelle des concepts relatifs à un domaine et des relations

entre ces concepts. Elle doit de plus être conçue comme un modèle réutilisable et partagé des connaissances afin de servir de base commune à l'ensemble du domaine pour l'interprétation et l'échange d'informations. La forme la plus courante selon laquelle les ontologies sont présentées consiste en une hiérarchie de concepts, organisés des plus généraux aux plus spécifiques. Une ontologie relative aux recettes de cuisine contiendra par exemple les concepts d'aliments et de plats. La figure 2.3 présente un extrait de la hiérarchie de concepts d'une telle ontologie. Chacun de ces concepts doit de plus être associé à une définition formelle, c'est-à-dire à une expression permettant de reconnaître un objet comme relatif au concept et d'en inférer de nouvelles informations. Sur l'exemple, le concept de recette pourra être défini comme un procédé permettant de réaliser un plat. Un plat froid pourra quant à lui être défini comme un plat dont la recette a un temps de cuisson nul. L'expression de ces définitions, c'est-à-dire la syntaxe utilisée et la sémantique associée, dépend du langage de représentations dans lequel l'ontologie est formalisée.[15]

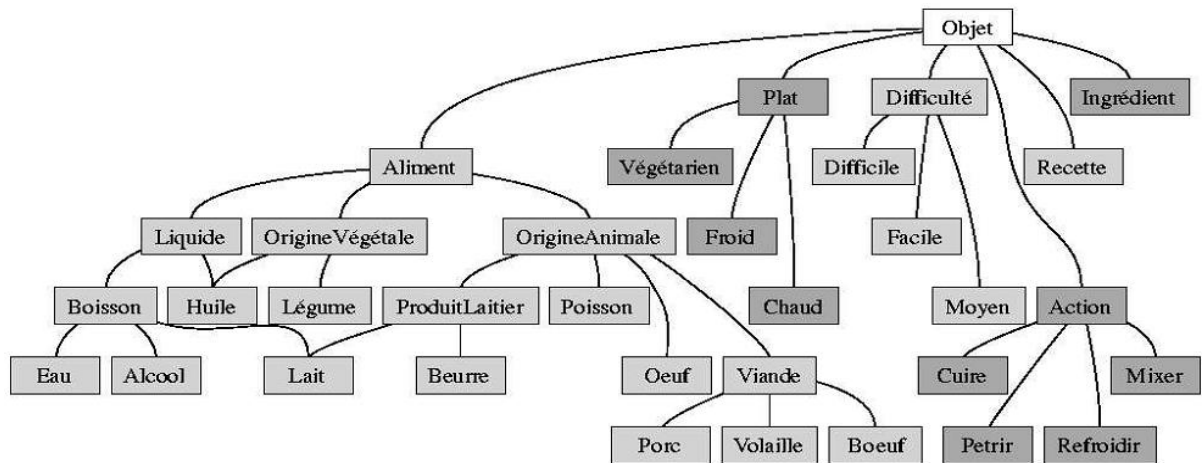


Figure 2.3 Hiérarchie de concepts d'une ontologie pour les recettes de cuisine.[15]

3.1.1 Définition En informatique

Les ontologies informatiques sont des outils qui permettent précisément de représenter un corpus de connaissances sous une forme utilisable par une machine.

Une des définitions de l'ontologie qui fait autorité est celle de Gruber:

Une ontologie est la spécification d'une conceptualisation d'un domaine de connaissance. Cette définition s'appuie sur deux dimensions :

Une ontologie est la conceptualisation d'un domaine, c'est-à-dire un choix quant à la manière de décrire un domaine. C'est par ailleurs la spécification de cette conceptualisation, c'est-à-dire sa description formelle.

C'est une base de formalisation des connaissances. Elle se situe à un certain niveau d'abstraction et dans un contexte particulier. C'est aussi une représentation d'une conceptualisation partagée et consensuelle, dans un domaine particulier et vers un objectif commun. Elle classe en catégories les relations entre les concepts. [16]

3.1.2 Définition en philosophie

Historiquement, le terme Ontologie a tout d'abord été défini en Philosophie comme une branche de la Métaphysique qui s'intéresse à l'existence, à l'être en tant qu'être et aux catégories fondamentales de l'existant, c'est-à-dire l'étude des propriétés générales de ce qui existe. Plus tard, le concept d'ontologie est apparu en pleine lumière dans le domaine de l'intelligence artificielle, afin de résoudre les problèmes de modélisation des connaissances et plus précisément, en ingénierie des connaissances.[21]

3.2 Apparition d'ontologie

L'ingénierie des connaissances (IC) a pour but la résolution automatique des problèmes, alors que les systèmes à Base de Connaissances (SBC) devraient permettre le stockage et la consultation et la modification des connaissances, ainsi que le raisonnement automatique dessus .

Le partage de connaissances entre systèmes informatiques permettra, de plus, une interaction et une coopération entre ces derniers et l'utilisateur humain. Cela se manifeste, par exemple dans : les systèmes d'aide à la décision, les systèmes d'enseignement assisté par ordinateur, la recherche d'information sur le web, etc. Pour permettre un traitement automatique efficace, les représentations à utiliser par les machines (les modèles) doivent être chargées de sens et cela en reliant les informations ainsi collectées et représentées à d'autres types d'informations vouées essentiellement à la sémantique sous-jacente. Cela a donné naissance à l'ingénierie ontologique.[17]

3.3 Rôle de l'ontologie

Les ontologies sont utilisées pour construire des bases de connaissances. Une base de connaissances peut être formée par une ontologie et un ensemble d'instances de ces classes.

Les ontologies définissent des vocabulaires structurés, regroupant des concepts utiles d'un domaine et de leurs relations et qui servent à organiser et échanger des informations de façon non ambiguë.

Les ontologies peuvent jouer le rôle d'entrepôt de données pour organiser l'information pour une communauté spécifique.

Le partage et la réutilisation d'information ou les ressources d'information peuvent être transmises entre les humains et les agents logiciels.

Interopérabilité : les relations sémantiques dans les ontologies sont lisibles par les machines de telle façon qu'elles augmentent l'interopérabilité entre les systèmes hétérogènes.[20]

3.4 Les composants d'ontologie

Comme suite:

Les concepts, Les relations, Les fonctions, Les axiomes, Les instances

3.4.1 Les concepts

Les concepts sont utilisés pour représenter les objets sur lesquels portent les connaissances à spécifier. Un concept décrit une notion et peut être décomposé en quatre parties : une intension, une extension, une terminologie et un identifiant.

- ✓ L'intension correspond à la définition formelle du concept. Elle présente un ensemble de caractéristiques (propriétés et attributs) qui expriment la sémantique du concept.
- ✓ L'extension correspond à l'ensemble des êtres que le concept englobe. Ces êtres possèdent en commun les caractéristiques définies par l'intension. Chacun de ces êtres correspond à une production du concept. Nous appellerons production lexicale du concept une représentation en corpus d'un élément de son extension.
- ✓ La terminologie d'un concept correspond aux termes qui lexicalisent la notion décrite par le concept. Usuellement elle est constituée d'un terme vedette (celui qui est le plus employé ou qui doit être employé) et de ses synonymes et variantes. Notons qu'un terme peut être ambigu, c'est-à-dire qu'il lexicalise plusieurs notions, chaque notion correspondant à un sens du terme. Dans ce cas, il convient au concepteur de l'ontologie de statuer sur le sens à retenir pour le terme dans le contexte de l'ontologie (c'est-à-dire en fonction des concepts présents dans celle-ci).
- ✓ L'identifiant (ou libellé formel) permet d'identifier le concept dans le modèle. Il correspond à une unité lexicale qui nomme le concept. L'identifiant est généralement choisi dans la terminologie du concept (il s'agit en général du terme vedette). Il est néanmoins préférable de distinguer l'identifiant du concept et le terme afin d'éviter les confusions entre les représentations conceptuelles et linguistiques d'une notion. Une solution consiste à utiliser une unité lexicale dérivée du terme vedette comme par exemple le terme vedette préfixé par « concept » ou « C_ ».[18]

3.4.2 Les relations

Une relation correspond à un lien s'établissant entre des concepts et décrit un type d'interaction entre ces concepts. Dans les ontologies, les relations sont généralement binaires. Une relation peut se décomposer en plusieurs éléments : une intension, une extension, une terminologie, un identifiant et une signature.

- ✓ L'intension d'une relation exprime la nature de la relation (équivalente au type d'interaction qu'elle décrit). Elle s'exprime par l'ensemble des attributs et des propriétés communes à toutes les réalisations de cette relation.
- ✓ L'extension d'une relation correspond à l'ensemble des réalisations de cette relation dans le domaine modélisé.
- ✓ La terminologie d'une relation correspond aux termes qui lexicalisent la notion décrite par la relation. Comme pour les concepts un terme peut être ambigu c'est-à-dire lexicaliser plusieurs relations. Dans ce cas, c'est au concepteur de déterminer, en fonction de l'ontologie, à quelle relation le terme doit faire référence.

- ✓ L'identifiant identifie une relation par un terme traduisant la nature de la relation. Ce terme est généralement issu ou inspiré d'un ou plusieurs termes de la terminologie.
- ✓ La signature d'une relation correspond aux concepts qui peuvent être liés par une réalisation de la relation dans le domaine modélisé. Elle s'exprime par un n-uplet de concepts.[18]

3.4.3 Les fonctions

Elles constituent un cas spécial de relations dans lesquelles le nième élément de la relation est unique est défini en fonction des n-1 éléments précédents.[18]

3.4.4 Les axiomes (ou Règles)

Les axiomes servent à modéliser des assertions toujours vraies dans le domaine (acceptées sans démonstration) qui se traduisent sous la forme de propriétés sur les concepts et/ou les relations. Les axiomes sont utilisés pour représenter des connaissances élémentaires du domaine qui ne peuvent être formellement définies par les autres constituants d'une ontologie. C'est le cas des fonctions qui imposent des contraintes sur le modèle. Par exemple, « apanne » est une fonction contraignant que la mère biologique d'une personne soit toujours une femme. Les axiomes permettent ainsi de restreindre l'interprétation des concepts et des relations dans le modèle.

Ils peuvent être inclus dans une ontologie pour plusieurs buts, tels que définir la signification de composants d'ontologie, définir les contraintes complexes sur les valeurs des attributs, les arguments de relations, etc., vérifier l'exactitude d'informations indiquées dans l'ontologie ou déduire une nouvelle information.[18]

3.4.5 Les instances (ou individus)

Elles regroupent à la fois les instances des concepts et celles des relations. Les instances d'un concept sont utilisées pour représenter les éléments spécifiques, singuliers. Une instance de relation est un n-uplet d'instances de concept (n le nombre de concepts de la signature). Par exemple « les individus mach1 et mach2 sont des instances du concept machine ».[18]

3.5 Classification d'ontologies

Les ontologies peuvent être classifiées selon plusieurs dimensions :

dans cette partie nous décrivons les plus connues dans la littérature et pour davantage de détails.

3.5.1 Classification Selon l'objet de conceptualisation

(le but de leur utilisation) :

on peut distinguer selon Gomez Pérez, les types suivants :

- 1- Ontologie de Représentation des connaissances.
- 2- Ontologie Supérieure/ Haut niveau.
- 3- Ontologie Générique.
- 4- Ontologie de Domaine.
- 5- Ontologie de Tâches.
- 6- Ontologie d'Application (Figure 2.4).[13]

3.5.1.1 Ontologie de Représentation des connaissances :

Elle décrit les concepts utilisés par les langages de représentation des ontologies. Elle qui définit des primitives de représentation utilisées pour formaliser la connaissance avec un paradigme donné. Les exemples les plus représentatifs sont la "Frame Ontology" et la "OKBC 12 Ontology", toutes les deux sont accessibles sur le serveur Ontolingua.[13]

3.5.1.2 Ontologie Supérieure ou de Haut niveau

("Top-level ou Upper-level ontologies") : Cette ontologie est une ontologie générale. Son sujet est l'étude des catégories des choses qui existent dans le monde, soit les concepts de haute abstraction tels que: les entités, les événements, les états, les processus, les actions, le temps, l'espace, les relations, les propriétés. L'ontologie de haut de niveau est fondée sur : la théorie de l'identité, la méréologie [13] et la théorie de la dépendance.[13]

3.5.1.3 Les ontologies génériques :

elles contiennent elles-aussi des concepts généraux, mais moins abstraits que ceux contenus dans les ontologies de haut niveau. On pourra réutiliser dans plusieurs domaines les connaissances que l'on y trouve (Psyché et al., 2003). Un exemple d'une telle ontologie est Ontolingua.

3.5.1.4 Les ontologies des tâches (Mizoguchi, 2003) :

ce type d'ontologie sert à modéliser les tâches d'un problème ou d'une activité donnée. Ce type d'ontologie est utile pour décrire la structure d'une tâche de résolution de problème de manière indépendante du domaine concerné.

3.5.1.5 Les ontologies de domaine

sont réutilisables à l'intérieur d'un domaine donné (Gomez Perez, 1999) et modélisent le vocabulaire à l'intérieur de ce domaine. La plupart des ontologies existantes sont des ontologies de domaine (Psyché et al., 2003).

3.5.1.6 Les ontologies de tâches-domaine

ce sont des ontologies de tâches spécifiques à un certain domaine. Un exemple d'une telle ontologie est celui d'une ontologie des termes liés à la planification chirurgicale (Gomez-Perez,1999).

3.5.1.7 Les ontologies d'application

Il s'agit du type d'ontologie le plus spécifique (Psyché et al. 03). Les concepts que l'on trouve dans ce genre d'ontologies modélisent les concepts d'un domaine particulier dans le cadre d'une application donnée.[19]

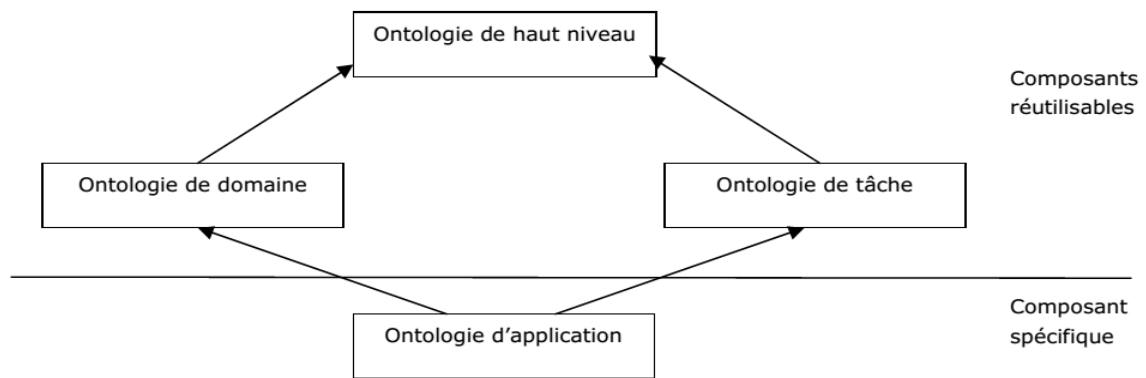


Figure 2.4 Classification des ontologies Selon l'objet de conceptualisation [13]

3.5.2 Classification selon le formalisme utilisé

les ontologies sont aussi classées en plusieurs niveaux d'expression selon leur utilisation. Quatre catégories principales qui vont des ontologies hautement informelles jusqu'aux ontologies rigoureusement formelles :

- ✓ Très informelle : l'ontologie est exprimée en langage naturel.
- ✓ Semi-informelle : l'ontologie est exprimé sous une forme restreinte et structurée de langage naturel pour augmenter la clarté et pour réduire l'ambiguïté.
- ✓ Semi-formelle : l'ontologie est exprimée en langage formel.
- ✓ Rigoureuse formelle : l'ontologie est défini avec une sémantique formelle, permettant le théorème et la preuve.[14]

3.5.3 Classification Selon le niveau de détail de l'ontologie

par rapport au niveau de détail utilisé lors de la conceptualisation de l'ontologie en fonction de l'objectif opérationnel envisagé pour l'ontologie, deux catégories au moins peuvent être identifiées :

- ✓ Granularité fine : correspondant à des ontologies très détaillées, possédant ainsi un vocabulaire plus riche capable d'assurer une description détaillée des concepts

pertinents d'un domaine ou d'une tâche. Ce niveau de granularité peut s'avérer utile lorsqu'il s'agit d'établir un consensus entre les agents qui l'utiliseront.

- ✓ Granularité large : correspondant à un vocabulaire moins détaillé comme par exemple dans les scénarios d'utilisation spécifiques où les utilisateurs sont déjà préalablement d'accord à propos d'une conceptualisation sous-jacente. Les ontologies de haut niveau possèdent une granularité large, compte tenu que les concepts qu'elles traduisent sont normalement raffinés subséquemment dans d'autres ontologies de domaine ou d'application.[13]

4 Conclusion

Nous connaissons maintenant le web sémantique et toutes ses composantes, en outre la classification et la définition de l'ontologie dans certains domaines et la description de certaines de ses composantes.

Pour construire cette ontologie, le prochain chapitre est celui qui accomplira cette tâche.

CHAPITRE 3
L'INGENIERIE ONTOLOGIQUE

Introduction

Dans ce chapitre, nous aborderons, donc les processus qui nous aident à construire de l'ontologie. Dans ce contexte, nous allons discuter de la façon dont construction des ontologies. C'est pour cela nous allons expliquer dans ce chapitre :

Cycle de vie d'une ontologie, quelques méthodologies de construction d'ontologie, outils de développement d'ontologie.

1 Construction des ontologies

La construction d'une ontologie est un travail réalisé conjointement par un ou plusieurs ingénieurs (ontologues) et des experts du domaine, ainsi qu'éventuellement de futurs utilisateurs de l'ontologie. De nombreuses méthodes de construction d'ontologie existent mais, de manière générale, ce processus peut se décomposer en deux étape suivant.[18]

1.1 Critères d'évaluation d'une ontologie

D'après Gruber, cinq critères permettent de mettre en évidence des aspects importants d'une ontologie :

- La clarté : la définition d'un concept doit faire passer le sens voulu du terme, de manière aussi objective que possible (indépendante du contexte). Une définition doit de plus être complète (c'est à dire définie par des conditions à la fois nécessaires et suffisantes) et documentée en langage naturel.
- La cohérence : rien qui ne puisse être inféré de l'ontologie ne doit entrer en contradiction avec les définitions des concepts (y compris celles en langage naturel).
- L'extensibilité : les extensions qui pourront être ajoutées à l'ontologie doivent être anticipées, il doit être possible d'ajouter de nouveaux concepts sans avoir à toucher aux fondations de l'ontologie.
- Une déformation d'encodage minimale : une déformation d'encodage a lieu lorsque la spécification influe la conceptualisation (un concept donné peut être plus simple à définir d'une certaine façon pour un langage d'ontologie donné, bien que cette définition ne corresponde pas exactement au sens initial). Ces déformations doivent être évitées autant que possible.
- Un engagement ontologique minimal : le but d'une ontologie est de définir un vocabulaire pour décrire un domaine, si possible de manière complète ni plus ni moins.

Contrairement aux bases de connaissances par exemple, on n'attend pas d'une ontologie d'être capable de fournir systématiquement une réponse à une question arbitraire sur le domaine. Une ontologie est la théorie la plus faible couvrant un domaine, elle ne définit que les termes nécessaires pour partager la connaissance liée à ce domaine.[24]

1.2 Le processus de construction d'une ontologie

Le processus de construction d'une ontologie est une collaboration qui réunit des experts du domaine de connaissance, 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. La Figure 3.1 représente le processus de construction d'ontologie.[23]

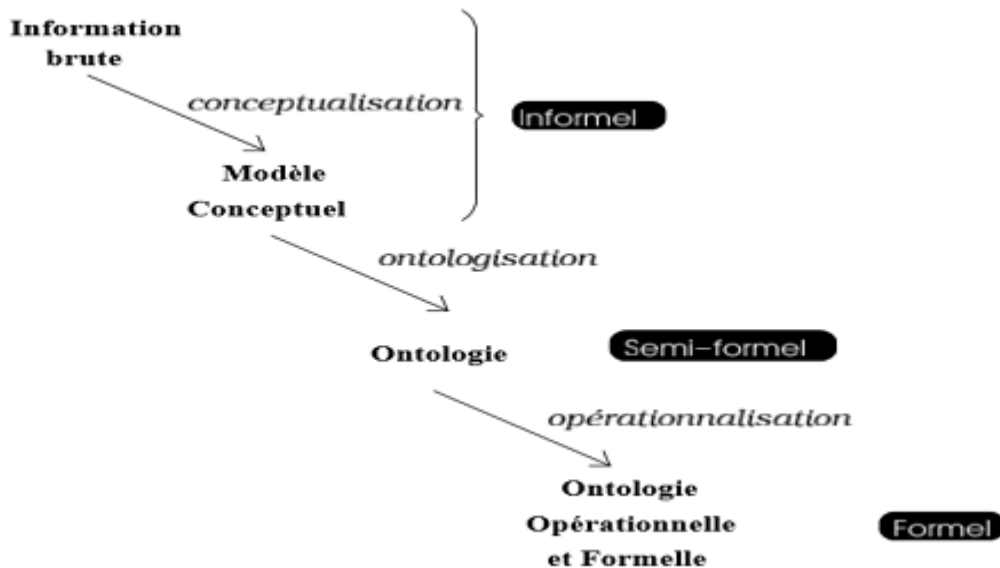


Figure 3.1 Processus de construction d'ontologie[23]

1.2.1 La conceptualisation

Identification des connaissances contenues dans un corpus représentatif du domaine. Ce travail doit être mené par un expert du domaine, assisté par un ingénieur de la connaissance.

1.2.2 L'ontologisation

Tormalisation, autant que possible, du modèle conceptuel obtenu à l'étape précédente. Ce travail doit être mené par l'ingénieur de la connaissance, assisté de l'expert du domaine.

1.2.3 l'opérationnalisation

Transcription de l'ontologie dans un langage formel et opérationnel de représentation de connaissances, pour permettre à une machine, via cette ontologie, de manipuler des connaissances du domaine. Ce travail doit être mené par l'ingénieur de la connaissance.[14]

2 Cycle de vie d'une ontologie

Étant donné que les ontologies sont destinées à être utilisées comme des composants logiciels dans des systèmes informatiques répondant à des objectifs opérationnels différents, leur développement doit s'appuyer sur les mêmes principes que ceux appliqués en génie logiciel. En particulier, elles doivent être considérées comme des objets techniques évolutifs

et posséder un cycle de vie spécifique. Les activités liées à une ontologie peuvent être regroupées en trois catégories .

- ✓ Des activités de gestion de projet : planification, contrôle, assurance qualité.
- ✓ Des activités de développement : spécification, conceptualisation, formalisation.
- ✓ Des activités de support: évaluation, documentation, gestion de la configuration.

La Figure 3.2 représente les différentes activités qui expliquent que le cycle de vie préconisé est un cycle par prototypes.[24]

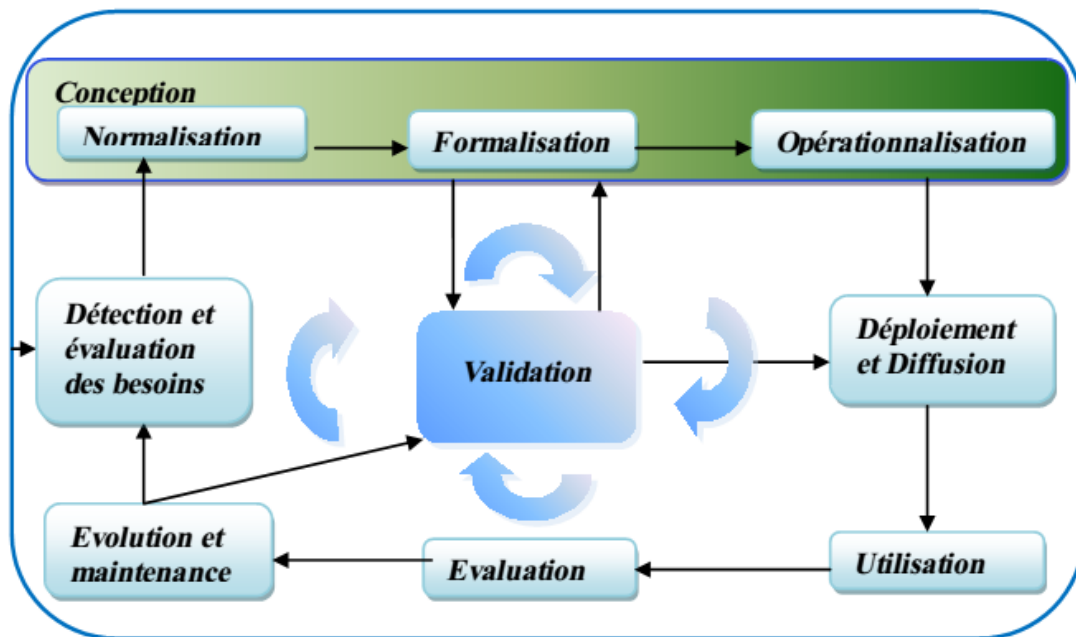


Figure 3.2 Le cycle de vie d'une ontologie [24].

Fernandez et ses collègues insistent sur le fait que les activités de documentation et d'évaluation sont nécessaires à l'étape du processus de construction d'ontologie, l'évaluation précoce permettant de limiter la propagation d'erreurs.[24]

2.1 Conception

2.1.1 Conceptualisation

Cette étape permet d'aboutir à un modèle informel, donc sémantiquement ambiguë et généralement exprimé en langage naturel. Elle consiste, à partir des données brutes à dégager les concepts et les relations entre ces concepts permettant de décrire de manière informelle les entités cognitives du domaine.

L'objectif est d'aboutir à un modèle conceptuel: ce modèle consiste en un ensemble de termes désignant les entités du domaine de connaissances (concept, relation, propriétés des concepts et des relations,. . .), assortis d'informations exprimant leur sémantique. La découverte des connaissances d'un domaine peut s'appuyer a la fois sur l'analyse de

documents et sur l'interview d'experts du domaine. Ces activités doivent être raffinées au fur et à mesure que la conceptualisation émerge.[30]

2.1.2 Ontologisation

L'Ontologisation consiste en une formalisation partielle, sans perte d'information, du modèle conceptuel obtenu dans l'étape précédente. Ce qui permet de faciliter sa représentation ultérieure dans un langage complètement formel et opérationnel. Elle effectue une transcription des connaissances dans un certain formalisme de connaissances, ce formalisme devant être aussi générique que possible mais sémantiquement clair. Le modèle obtenu est souvent qualifié de semi-formel (car certaines connaissances ne peuvent pas être totalement formalisées). Le caractère semi-formel d'une ontologie lui interdit d'être utilisée telle quelle dans un SBC. En revanche, une ontologie, contenant toutes les connaissances d'un domaine constitue le support idéal de communication et de partage des connaissances de ce domaine.[30]

2.1.3 Opérationnalisation

Cette étape consiste à formaliser complètement l'ontologie obtenue dans un langage de représentation de connaissance formel (i.e. possédant une syntaxe et une sémantique) et opérationnel (i.e. doté de services différentiels permettant de mettre en œuvre des raisonnements), 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.[30]

2.2 Spécification des besoins

Cette étape consiste à établir un document formel de spécification des besoins représenté dans le langage RDF. Ce dernier permet de décrire l'ontologie à construire à travers les cinq aspects suivants :

- ✓ Le domaine de connaissance: 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 au maximum les futurs utilisateurs de l'ontologie à créer..
- ✓ Les sources d'informations : déterminer les sources d'informations d'où les connaissances seront obtenues, par exemple, les experts du domaine, les documents techniques, etc., sont des sources de connaissance.
- ✓ La portée de l'ontologie : déterminer à priori la liste des termes (les plus importants) pour le domaine à représenter.[30]

2.3 Diffusion et cas d'utilisation de l'ontologie

Pour rendre une ontologie publique, en plus de lui donner une URL, on peut :

- ❖ L'ajouter à un ou plusieurs répertoires d'ontologies publiques disponibles sur le web.
- ❖ La publier dans une version facile à interpréter par la machine (RDF/OWL, etc.) ou dans le format html plus facile à lire et à interpréter pour l'humain.
- ❖ La rendre explorable dans un éditeur d'ontologies en ligne.

Les répertoires d'ontologies aussi appelés librairies d'ontologies sont des systèmes qui collectent les ontologies de différentes sources afin de faciliter leur recherche, leur exploration et leur utilisation. Mathieu d'Aquin et al.[22]

2.4 Utilisation

L'utilisation d'ontologie pour la simulation est un domaine de recherche à explorer, il est également nécessaire pour les utilisateurs de simulation pour être en mesure de recueillir des informations auprès des différents niveaux, donc l'objectif est de développer une représentation fidèle de connaissances en utilisant un éditeur d'ontologie.[24]

2.5 Evaluation d'une ontologie

L'évaluation d'une ontologie se fait soit en l'utilisant dans des applications soit en la discutant avec les experts de domaine.[24]

3 Quelques méthodologies de construction d'ontologie

Les méthodologies peuvent porter sur l'ensemble du processus et guider l'ontologiste dans toutes les étapes de la construction. Bien qu'aucune méthodologie générale n'ait pour l'instant réussi à s'imposer, de nombreux critères de construction d'ontologies ont été proposés pour des méthodologies. METHONTOLOGIE TOVE ENTERPRISE, et sont les méthodologies les plus représentatives pour construire des ontologies. [20]

3.1 La méthode METHONTOLOGY

Cette méthode est développée au laboratoire d'intelligence artificielle. Elle vise la construction d'ontologie au niveau de connaissance. Ce projet a été motivé par le constat suivant :

L'absence de méthodes ou de guides structurés est un obstacle à la construction d'ontologies partagées et consensuelles. Il est également un obstacle à l'extension d'une ontologie existante ou à sa réutilisation dans d'autres ontologies. L'approche METHONTOLOGY distingue plusieurs étapes dans son processus de construction.[26]

La Figure 3.3 : présente le processus de la méthode METHONTOLOGIE.

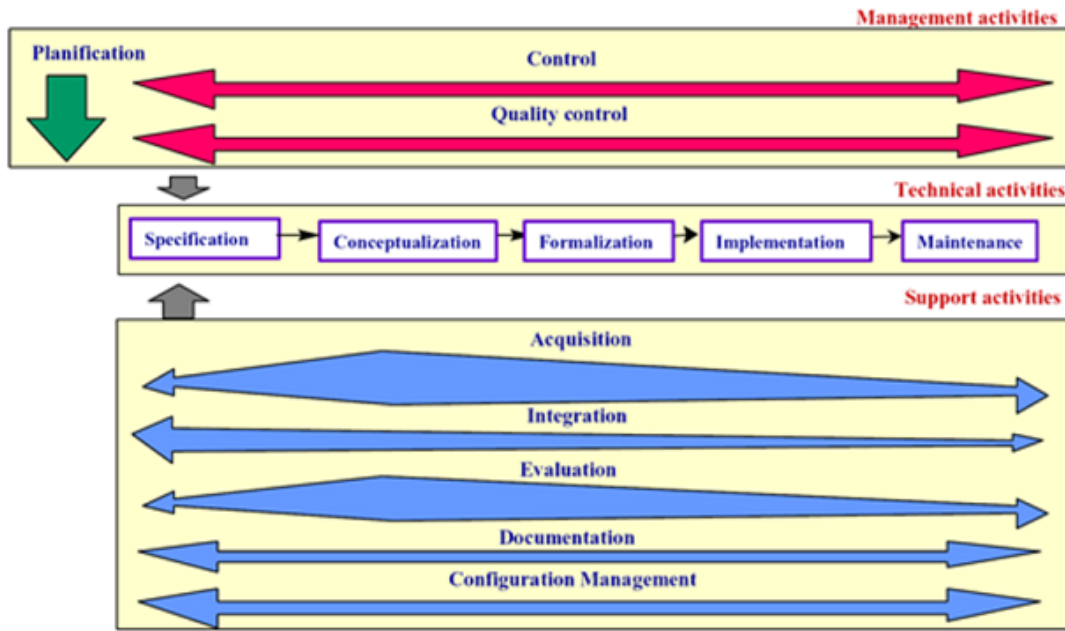


Figure 3.3 Processus de construction de l'ontologie selon METHONTOLOGY.[26]

3.1.1 Spécification

Cette étape a pour but de fournir une description claire du problème étudié ainsi que la façon de le résoudre. Elle permet de préciser l'objectif, la portée et le degré de granularité de l'ontologie qui sera construite.[26]

3.1.2 Conceptualisation

Cette étape est illustrée dans la figure (3.4). 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. La Figure 3.4 est contenue les tâches qui consistent dans l'étape de conceptualisation.[20]

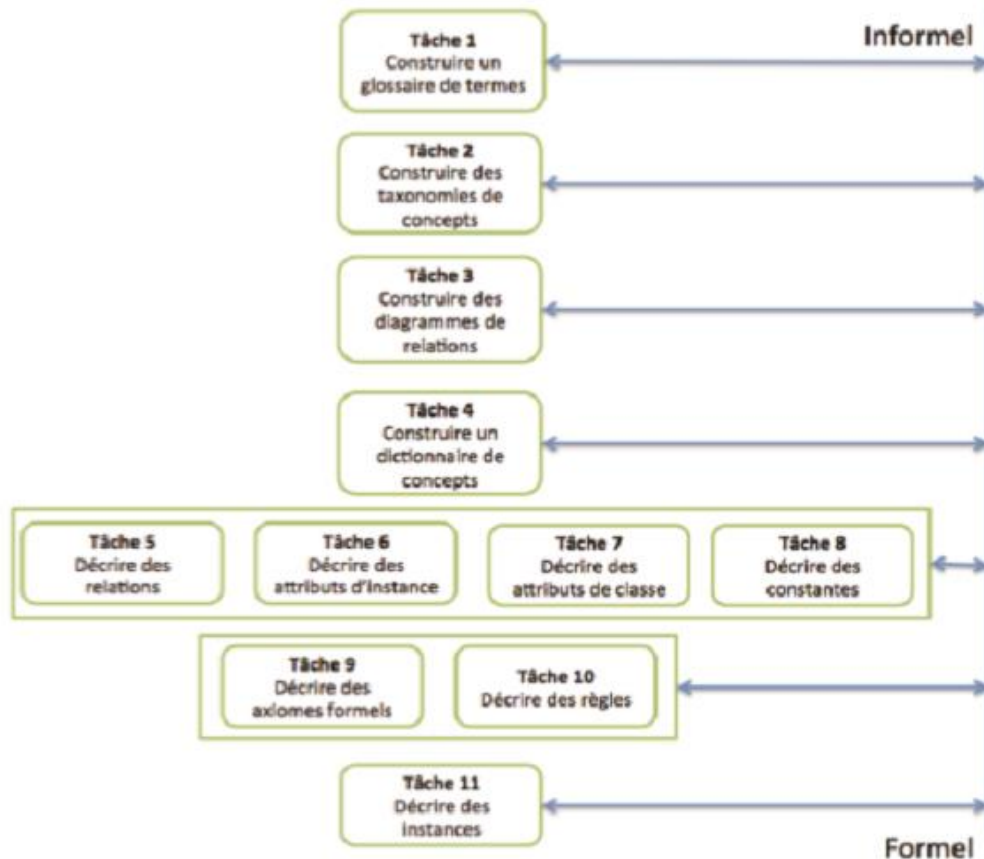


Figure 3.4 Processus de développement d'une ontologie[20]

Tache 1 : Construire le glossaire de termes

Cette tache consiste a rassembler un maximum de termes du domaine dans un glossaire, en utilisant les sources d'information déjà citées dans la phase de spécification. Le glossaire inclut les éléments suivantes :une description des concepts, leurs synonymes, leurs acronymes (abréviation) et leurs types (concept(classes) , instance...).

Tache 2 : Construction d'une taxonomie de concepts

Il s'agit de déterminer les sous classes d'une classe.

Tache 3 : Construction d'un diagramme de relations

Il s'agit de déterminer toutes les relations binaires et vérifier ces relations entre les classes.

Tache 4 : Construction d'un dictionnaire de concepts

Le dictionnaire de concepts dérive du glossaire de termes. Il est constitué des éléments suivantes :

- Les concepts du domaine et leurs instances
- Les relations

- Les attributs des classes et des instances.[20]

Tache 5 : Décrire les relations

La description des relations définies dans le dictionnaire des concepts en précisant un ensemble d'élément à savoir: le nom, des relations, leurs concepts source, leur concept cible, leurs cardinalités, leurs relations inverses si elles existent.

Tache 6 : Décrire les attributs d'instance

Il s'agit de construire un tableau qui va contenir le nom de l'attribut, le nom de concept père, son type et sa valeur, la plage de ses valeurs, les formules ou les règles qui peuvent inférer les valeurs de cet attribut.

Tache 7 : Décrire les attributs de classe

Faire de la même manière que la description des attributs d'instances.

Tache 8 : Décrire les constantes

La description d'une constante revient à lui donner sa valeur.

Tache 9 : Décrire les axiomes formelles

Il faut identifier les axiomes formels nécessaires à l'ontologie et les décrire avec précision dans une table. Pour chaque définition d'axiome formel, il faut spécifier le nom, la description; l'expression logique qui le décrit formellement (de préférence en utilisant la logique du premier ordre), les concepts, les attributs et les relations auxquelles l'axiome fait référence ainsi que les variables utilisées.

Tache10 :Description des règles et des axiomes formelles

Il faut identifier quelles règles sont nécessaires dans l'ontologie et les décrire dans une table de règles. Pour chaque règle, il faut spécifier : le nom; la description; l'expression qui la décrit formellement, les concepts, les attributs, les relations auxquelles elle font référence et les variables utilisées dans l'expression. Pour la spécification des règles, il est suggéré d'utiliser la syntaxe suivante :

Si<condition>alors<conséquences et actions>

Tache 11 : Décrire les instances

C'est une tache facultative qui sert à donner plus de précision sur les instances. Elle consiste à associer à chaque instance le nom de sa super classe et attribue des noms et des valeurs de ses attributs. [20]

3.1.3 Formalisation

Cette étape consiste en la transcription du modèle conceptuel de l'ontologie dans un langage formel de représentation de connaissances.

3.1.4 Implémentation

Elle consiste à la codification de l'ontologie formelle dans un langage opérationnel du Web Sémantique.

3.1.5 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 LDs, et qui sont offerts par des raisonneurs.[26]

3.2 La méthode Tove

TOVE (Toronto Virtual Enterprise) développé par l'université de Toronto, cette méthodologie repose sur les expériences de développement d'une entreprise. 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), identifier 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.[30]

3.3 La méthode ENTREPRISE

Uschold, propose le squelette d'une méthode basé sur l'expérience de construction d'ontologie dans le domaine de la gestion des entreprises. La méthode ENTERPRISE repose sur les quatre étapes suivantes :

- Identifier le rôle et la portée de l'ontologie.

- Identifier les concepts et relations fondamentaux et des définitions provisoires de ces éléments. Coder l'ontologie dans un langage adapté. Intégrer des ontologies existantes. Dans cette étape, l'ontologie est réellement construite.
- Evaluer l'ontologie.
- Rédiger une documentation et une trace des actions réalisées lors des différentes phases.[30]

4 Outils de développement d'ontologie

Les outils de développement d'ontologies qui existent sur le marché aujourd'hui sont divers et variés. Cet état de choses suscite beaucoup d'interrogations lorsque vient le moment d'en choisir un pour construire une nouvelle ontologie : L'outil offre-t-il une assistance au développement ? L'outil dispose-t-il d'un moteur d'inférence? Quels langages d'ontologies l'outil supporte-t-il? L'outil permet-il d'importer / exporter des ontologies ? L'outil offre t-il un support à la réutilisation d'ontologies existantes? L'outil permet-il de documenter les ontologies construites ? L'outil offre-t-il un support graphique à la construction des ontologies ? L'outil est-il stable, convivial, « mature» ? 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.[23]

4.1 Les langages de spécification d'ontologies

Plusieurs langages de spécification d'ontologies (ou langages d'ontologies) ont été développés pendant les dernières années, et ils deviendront sûrement des langages d'ontologie dans le contexte du Web sémantique. Certains d'entre eux sont basés sur la syntaxe de XML, tels que XOL (Ontology Exchange Language), SHOE (Simple HTML Ontology Extension - qui a été précédemment basé sur le HTML), OML (Ontology Markup Language), RDF (Resource Description Framework), RDF Schéma. Les deux derniers sont des langages créés par des groupes de travail du World Wide Web Consortium. En conclusion, trois langages additionnels sont établis sur RDF(S) pour améliorer ses caractéristiques: OIL (Ontology Inference Layer), DAML+OIL et OWL (Web Ontology Language).

La figure 3.5 présente des langages de spécification d'ontologie, qui ont été récemment développés. La figure ci - dessous représente les rapports principaux entre tous ces langages sous la forme d'une pyramide des langages du Web sémantique.[26]

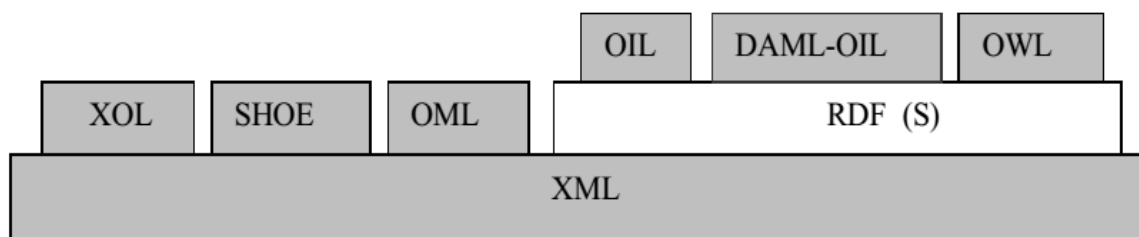


Figure 3.5 La pyramide des langages d'ontologies basés Web.[26]

✚ RDF(Ressource Description Framework)

RDF développé et recommandé par le W3C, permet de décrire les ressources du web sémantique qui sont l'élément de base de RDF. Chaque ressource est pourvue d'un identifiant URI (Uniform Resource Identifier). Tout document RDF est composé d'un ensemble de triplets (sujet, prédicat, objet) ou encore (ressource, propriété, valeur). Un ensemble de tels triplets est appelé un graphe RDF. Ceci peut être illustré par un diagramme composé de nœuds et d'arcs orientés, dans lequel chaque triplet est représenté par un lien nœud-arc-nœud (d'où le terme de "graphe"). A ce modèle est associée une syntaxe écrite en XML et basée sur les triplets :

- ✓ Ressource (Sujet) : une entité d'informations pouvant être référencée par un identificateur. Cet identificateur doit être une URI.
 - ✓ Propriété (prédicat) : l'attribut ou la relation utilisé (e) pour décrire une ressource.
 - ✓ Valeur (objet) : la valeur d'une propriété associée à une ressource spécifique. [18]
- ✚ RDF(S) :

RDF permet de déclarer des propriétés (et relations) sur des ressources ,spécifier le type des propriétés et des ressources (rdf:type), mais ne permet pas de décrire les types des propriétés (relations), des ressources. En d'autre terme RDF ne permet pas de répondre :

- ✓ Quelles sont les valeurs autorisées pour une propriété ?
- ✓ Quelles sont les propriétés autorisées sur une ressource ?
- ✓ Quels sont les liens entre les ressources ?
- ✓ Hiérarchie des types de ressources (spécialisation/généralisation) ?

Résultat : absence de vocabulaire —→ La solution :RDFS

RDFS définition:

- ✓ RDFS a Pour exprimer un vocabulaire.
- ✓ Pour des raisons historiques ce langage est dénommé RDFS (analogie XMLSchema).
- ✓ On parle maintenant plutôt de 'RDF Vocabulary Language.
- ✓ Le préfixe pour tous les éléments du vocabulaire RDFS est :

<http://www.w3.org/2000/01/rdf-schema#>

- ❖ RDFS est un système de typage pour RDF comparable à l'approche orienté objet:
 - Classes et instances.
 - Propriétés des classes.
 - Hiérarchie des classes (spécialisation/généralisation).
 - Pas les méthodes.
- ❖ RDFS permet de :
 - Définir des classes et des sous classes.
 - Définir les propriétés d'une classes.
 - Définir les relations entre classes.

- Restriction de domaine et co-domaine d'une propriété.
 - RDFS/RDF permet de décrire des ontologies légères.[7]
- ✚ DAML+OIL

DAML (DARPA Agent Markup Language) est un langage qui a comme 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, nouveau langage nommé OIL a été développé par un groupe des chercheurs pour le même but. Ce langage a une syntaxe basée sur RDF et il est explicitement construit pour sa sémantique puisse être spécifiée à travers une description logique très expressive, la logique de description.

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.[30]

✚ OWL : (Web Ontologie Language)

Le problème de RDF est qu'il n'introduit aucune normalisation sur la sémantique proprement dite, et que divers utilisateurs pourront donner des noms différents à des entités de même sens, ce qui empêche d'imaginer toute agrégation entre des données provenant de sources Web différentes.

La solution vient du développement d'ontologies, qui décrivent formellement les termes et les relations entre eux. OWL a été conçu pour satisfaire le besoin d'un langage d'ontologie du Web, il est basé sur la recherche effectuée dans le domaine de la logique de description, il repose sur la syntaxe des triplets RDF et réutilise certaines des constructions RDFS, il ajoute plus de vocabulaire pour décrire les propriétés et les classes. On peut citer entre autre : les relations entre classes (par exemple la disjonction), les cardinalités (par exemple exactement un), l'égalité, typage plus riche des propriétés, caractéristiques des propriétés (par exemple la symétrie) et les classes énumérées. OWL a été fractionné en trois sous langages ou chacun est une extension par rapport à son prédécesseur :

- ❖ OWL LITE :répond à des besoins de hiérarchie de classification et de fonctionnalité de contraintes simples de cardinalité 0 ou 1. Une cardinalité 0 ou 1 correspond à des relations fonctionnelles, par exemple, une personne a une adresse. Toute fois, cette personne peut avoir un ou plusieurs prénoms, OWL Lite ne suffit donc pas pour cette situation.
- ❖ OWL DL : concerne les utilisateurs qui souhaitent une expressivité maximum couplée à la complétude du calcul (cela signifie que toutes les inférences seront assurées d'être prises en compte) et la décidabilité du système de raisonnement (c'est-à-dire que tous

les calculs seront terminés dans un intervalle de temps fini). Ce langage inclut toutes les structures OWL avec certaines restrictions, comme la séparation des types: une classe ne peut pas aussi être un individu ou une propriété. Il est nommé DL car il correspond à la logique descriptive.

- ❖ OWL FULL: se destine aux personnes souhaitant une expressivité maximale. Il a l'avantage de la compatibilité complète avec RDF/RDFS, mais l'inconvénient d'avoir un haut niveau de capacité de description, quitte à ne pas pouvoir garantir la complétude et la décidabilité des calculs liés à l'ontologie. [26].

4.2 Langages d'interrogation d'ontologie

4.2.1. utilisation du protocole de requête SPARQL

Une fois l'ontologie construite et sauvegardée dans un fichier OWL, il faut utiliser un protocole de requête spécifique au langage RDF construit sur des graphes de données .

Le SPARQL (prononcer « sparkle », acronyme de SPARQL Protocol And RDF Query Language), recommandé depuis le 15 Janvier 2008 par le W3C, définit la syntaxe et la sémantique nécessaire à l'expression de requêtes sur une base de connaissances RDF. SPARQL est adapté à la structure spécifique des graphes de données RDF, et s'appuie sur les triplets qui les constituent. SPARQL permet d'exprimer des requêtes de type interrogatives ou constructives.[4]

SPARQL est un langage de requête et un protocole qui permettra de rechercher, d'ajouter, de modifier ou de supprimer des données RDF/RDFS disponible à travers Internet.

SPARQL s'apparente à SQL qui comme SQL accède aux base de données via une langage de requête, mais SPARQL accède lui aux ressources présentes sur le Web.

Le principe de langage de SPARQL est:

- pattern (motif): Un pattern est un triplet ou plusieurs triplet avec des variables (prfixees par ?) sont introduites afin de matérialiser les ressources/propriétés/valeurs littérales a retourner.

SPARQL Utilise du pattern matching sur la donnée graphe:

- la requête est un graphe avec variables
- on recherche les évaluations des variables qui soient des sous-graphes de la donnée.[8]

4.2.2 Le langage SWRL

SWRL « Semantic Web Rule Language » (I. Horrocks et al., 2004) est un langage de règles pour le web sémantique, combinant le langage OWL-DL et le langage RuleML « Rule Markup Language (Unary/Binary Datalog) ». Avec SWRL, les connaissances peuvent être exprimées sous la forme: antécédent → conclusion. Les axiomes du langage OWL-DL sont étendus à l'aide des clauses de Horn (Horn, 1951) réduites aux prédicats unaires et binaires.

Ce langage est indécidable, les règles SWRL sont appliquées même si les individus ne sont pas présents dans la base. Une restriction de SWRL, appelée DL-safe rules, a été conçue pour conserver la décidabilité, c'est-à-dire que les règles écrites ne portent que sur des individus explicitement présents dans la base. Autrement dit, les règles ne peuvent s'appliquer que si l'identité de toutes les instances présentes est connue. De nombreux moteurs d'inférences commencent à supporter SWRL: Bossam, Hoolet, KAON2, Pellet, RacerPro, R2ML « REVERSE Rule Markup Language » et Sesame. Ils suivent trois types d'approches :

- Traduire SWRL en logique du premier ordre (Hoolet) et appliquer un algorithme de chaînage avant.
- Traduire OWL-DL en règles et appliquer un algorithme de chaînage avant (Bossam).
- Intégrer les règles SWRL dans le moteur d'inférences OWL-DL fondé sur les algorithmes des tableaux sémantiques (Pellet) (Sirin, Parsia, Grau, Kalyanpur, & Katz, 2007).

SWRL est aussi requis pour représenter des expressions contenant des opérateurs mathématiques (addition, quotient, soustraction, produit, etc.) (Prcela, Gamberger, & Jovic, 2008). [25]

4.3 Les moteurs d'inférence

Inférence: opération mentale qui consiste à tirer une conclusion d'une série de propositions reconnues pour vraies". Appliqué à OWL, inféré une ontologie consiste à en tirer une série de nouveau objet qui découle de la définition des objets dans l'ontologie (par exemple, ajout d'attribut sur un individus ou d'individus dans une classe). On utilise pour ceci un moteur d'inférence, aussi appelé raisonneur (raisonner). Parmi les outils de raisonnement exploités nous citons quelques moteurs d'inférence.

Actuellement, plusieurs moteurs d'inférences gratuits ou commerciaux tels que Racer, Pellet, Fact, Fact++, Surnia, F-OWL et Howlet existent. La majorité de ces moteurs sont conçus pour raisonner sur les logiques de description, mais acceptent en entrée des fichiers OWL. Certains moteurs d'inférence ne peuvent raisonner qu'au niveau terminologique (c'est-à-dire au niveau des concepts et des propriétés). Nous présenterons plus en détail dans la suite Pellet et Racer qui sont à l'heure actuelle les deux seuls moteurs d'inférence, permettant le raisonnement sur la ABox et la TBox et exploitent des ontologies possédant un niveau d'expressivité en logique de description et en OWL satisfaisant.

Le tableau suivant dresse une comparaison des principaux moteurs d'inférence pour les LD43: FaCT , Racer, Pellet , FaCT++ , F-OWL , Surnia44, et Hoolet45. Le critère "Mise-à-échelle" mesure la capacité à demeurer efficace proportionnellement à la complexification des ontologies. Le tableau reprend les données de pour ce critère, celui de décidabilité et pour les caractéristiques de Hoolet, Surnia et F-OWL. [27]

Moteur	Racer	FaCT	Pellet	FaCT++	Surnia	Hoolet	F-OWL
LD	SHIQ(D)	SHIQ, SHF	SHIN(D), SHON(D)	SHIF(D)	Logique prédicats	Logique prédicats	SHIQ(D) et RDF
Implantation	C++	Common Lisp	Java	C++	Python	Java	Java
Inférence	TBox/ABox	TBox	TBox/ABox	TBox	TBox/ABox	TBox/ABox	TBox/ABox
API Java	Oui	Oui	Natif	oui	?	Oui	oui
Mise-à-échelle	Bonne	Bonne	Bonne	bonne	Médiocre	médiocre	Médiocre
OWL	OWL-DL~ +	OWL-DL~ +	OWL-DL~ +	OWL-DL~ +	OWL-FULL~ +	OWL-DL~ +	OWL-FULL~ +
Décidabilité	oui(OWL-LITE)	oui	oui (OWL-LITE)	oui	non	Non	non
DIG46	Oui	Oui	Non	?	Non	Non	Non

Tableau 3.1 Tableau comparatif des principaux moteurs d'inférence pour LD.[27]

4.3.1 Racer

Racer est le moteur d'inférence sans doute le plus connu et l'un des plus utilisés dans le domaine à cause de performances et sa stabilité. Il est commercialisé par Racer Systems GmbH & Co. KG, fondé en 2004 par Volker Haarslev, Kay Hidde, Ralf Möller 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 requête nRQL (new Racerpro query Language) pour interroger les ontologies sur la ABox et la TBox

Les avantages

- La documentation sur Racer est importante, provenant des concepteurs et des utilisateurs.
- Racer permet l'utilisation d'un mécanisme d'abonnement à un concept qui permet d'être informé de la création de nouvelles instances de ce concept.
- Il permet de créer avec ce mécanisme d'abonnement un monde fermé local. Dans le monde ouvert, il n'est pas possible de s'abonner au concept de Livre sans Auteur, car un livre peut avoir un auteur décrit dans le futur. Racer permet de le faire. En effectuant préalablement 2 souscriptions la première (A) sur le concept de Livre, la seconde (B) sur le concept de Livre avec au moins un Auteur. L'abonnement sur le concept définit par A – B est donc un abonnement sur les livres sans auteur.
- Racer permet l'ajout d'assertions et d'individus dans les ABox après le chargement de l'ontologie.
- Racer permet l'utilisation de règles SWRL.

Les inconvénients

- Racer suppose que tous les propriétés sur les datatypes sont fonctionnelles (pas de valeurs multiples pour un datatype property)

- Racer ne permet pas l'utilisation de type de données utilisateur (type défini par l'utilisateur), car il possède ces propres types de données et il effectue une conversion avec les types de base.
- Racer est un produit commercial, il n'existe pas de version libre d'utilisation. Cependant il est possible d'obtenir une licence gratuite dans le cadre de la recherche scientifique. [27]

4.3.2 Pellet

Le moteur Pellet est beaucoup plus récent. Pellet 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 OpenSource et offre des évolutions fréquentes. Pellet travaille sur des ontologies décrites en RDF ou OWL et permet les requêtes avec RDQL et SPARQL sur la ABox et la TBox. Comme pour Racer, nous allons présenter quelques avantages et points négatifs de Pellet.

Les avantages

- Pellet est open-source et développé en Java.
- Pellet est un raisonneur OWL DL complet.
- Pellet propose en cas d'incohérence dans l'ontologie des réparations possibles, ainsi qu'une heuristique permettant d'obtenir les informations à ajouter dans l'ontologie pour passer au sous-langage OWL inférieur (OWL Full > OWL DL > OWL Lite).

Les inconvénients

- Pellet possède une documentation pauvre en comparaison de celle de Racer. En effet racer est le plus utilisé et donc le plus documenté par des particuliers. De plus la documentation officielle de Pellet reste assez pauvre en comparaison de celle de Racer.
- Actuellement Pellet ne permet pas l'utilisation de règles SWRL26 (la prochaine version l'inclura).
- Pellet n'offre pas de système de souscription à un concept. [27]

4.4. Les éditeurs d'ontologie

De nombreux outils de construction d'ontologies utilisant des formalismes variés ont été développés, beaucoup d'outils offrent des supports pour le processus d'ontologisation, mais peu d'entre eux offrent une aide à la conceptualisation. Ces outils d'aide à la construction d'ontologie sont plus ou moins indépendants des formalismes de représentation, parmi ces outils, on peut citer.[24]

4.4.1 ONTOLINGUA

De l'Université Stanford; Le serveur Ontolingua est le plus connu, c'est un environnement de construction d'ontologies en langage Ontolingua. Il consiste en un ensemble d'outils et de services qui supportent la construction en coopération d'ontologies, entre des groupes séparés géographiquement.[24]

4.4.2 Protégé

Protégé est une plateforme open-source qui propose au travers d'une communauté grandissante une suite d'outils de construction de modèles de domaine. Au delà de cette communauté, l'outil Protégé permet la création d'ontologies et la possibilité de modifier l'outil à convenance afin d'ajouter diverses fonctionnalités spécifiques. Plusieurs travaux portent sur l'intégration au sein de l'outil divers plugins permettant des traitements spécifiques. On notera par exemple les travaux de l'équipe de A. Tissaoui sur une méthodologie d'évolution d'une ressources termino-ontologique : EvOnto (Tissaoui, 2010) proposant une méthode formalisée au travers d'un plugin pour Protégé. L'ontologie étant considéré comme stable mais cependant voué à modification selon l'évolution des ressources termino-ontologique. Ce plugin propose une assistance à la modification des ressources.[29]

4.4.3 WEBONTO

Du Knowledge Media Institute de l'Open University ; WebOnto et TADZEBAO sont des outils complémentaires. Tadzebao permet aux ingénieurs des connaissances de tenir des discussions sur les ontologies, en mode synchrone et asynchrone. WebOnto supporte la navigation collaborative, la création et l'édition d'ontologies sur le Web. [24]

4.4.4 SWOOP

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 (pas seulement XML), c'est une application plus légère que Protégé, moins évoluée en termes d'interface, mais qui intègre aussi des outils de raisonnement.[24]

4.4.5 ONTOEDIT

(ONTOLOGY EDITOR) développé par la compagnie Ontoprise, c'est un environnement de construction d'ontologies basé sur une méthodologie. Il permet l'édition des hiérarchies de concepts et de relations dans le cadre du paradigme des frames et l'expression d'axiomes algébriques portant sur les relations, et de propriétés telles que la généralité d'un concept. Des outils graphiques dédiés à la visualisation d'ontologies sont inclus dans l'environnement. OntoEdit intègre, dans sa version commerciale, un serveur destiné à l'édition d'une ontologie par plusieurs utilisateurs ainsi qu'un plug-in permettant le test de la cohérence d'une ontologie. Enfin, un plugin nommé ONTOKICK offre la possibilité de générer les spécifications de l'ontologie par l'intermédiaire de questions de compétence. OntoEdit gère de nombreux formats de représentation de connaissance dont DAML+OIL, RDFS et FLogic [24].

5. Conclusion

Nous concluons que l'ontologie a besoin de construire plusieurs exigences et nous avons fourni la plupart des méthodes qui contribuent à la construction et au développement de l'ontologie et qui nous aidera dans le travail que nous ferons.

CHAPITRE 4
CONCEPTION ET RÉALISATION

Introduction

Maintenant, nous avons appris comment nous pouvons utiliser la méthode "METHONTOLOGY" dans le chapitre précédant, donc nous pouvons utiliser dans notre travail l'application de la méthode "METHONTOLOGY" pour connaître les défaillances et les pannes de la machine, qui est également basé sur le langage UML pour créer des graphiques, et a également expliqué comment faire certaines applications en capturant l'écran, et suivez les étapes de la méthode "METHONTOLOGY".

1 Choix de la méthode de construction de notre ontologie

C'est l'étape la plus importante dans le processus de construction de l'ontologie. Elle est inspirée de la méthodologie "METHONTOLOGY" qui consiste à identifier et à structurer, à partir des sources d'informations, les connaissances du domaine. Elles permettent d'aboutir à un ensemble de représentations intermédiaires semi-formelles indépendamment des langages de formalisations à utiliser pour représenter l'ontologie "METHONTOLOGY".

Ce processus est composé de tâches suivantes :

- ❖ Spécification des besoins.
- ❖ Conceptualisation.
- ❖ Formalisation.
- ❖ Implémentation.
- ❖ Maintenance.[30]

1.1 Spécification

UML est un langage ou formalisme de modélisation graphique. Il est apparu dans le monde du génie logiciel, dans le cadre de la « conception orientée objet ». UML représente un moyen de spécifier et représenter les composantes d'un système informatique. Parmi les objectifs d'UML : être indépendant des langages de programmation et être adapté à toutes les phases de développement.

Parmi l'ensemble des diagrammes que propose UML, on va utiliser seulement : le diagramme des Cas d'Utilisation et le diagramme de Classes et le diagramme de séquence.[21]

1.1.1 Diagramme de cas d'utilisation

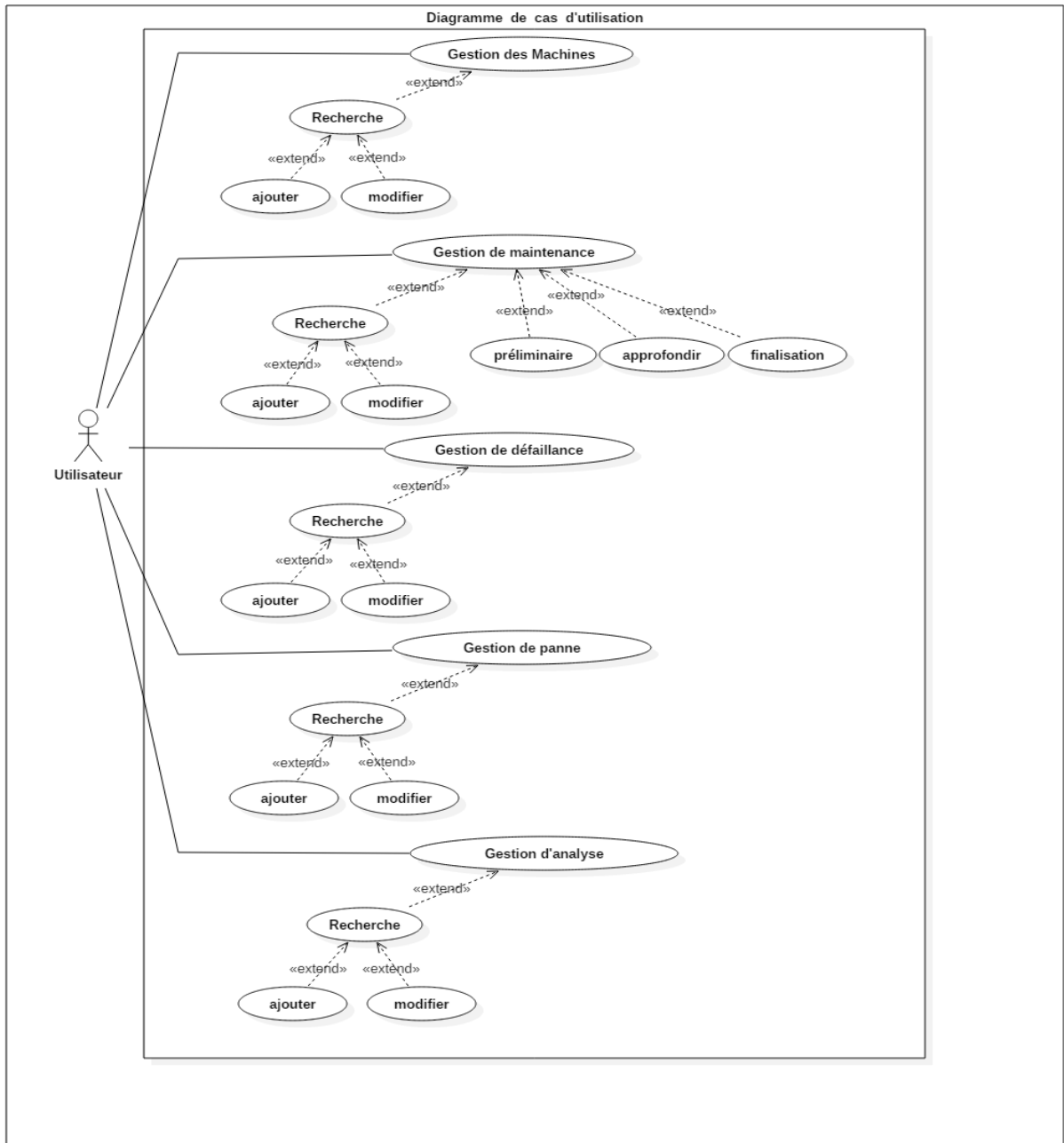


Figure 4.1 Diagramme de cas d'utilisation.

1.1.2 Les diagrammes de séquence

Tout les diagrammes de séquence existe ajouter et modifier et recherche de classe

1.1.2.1 diagramme de séquence ajouter une machine

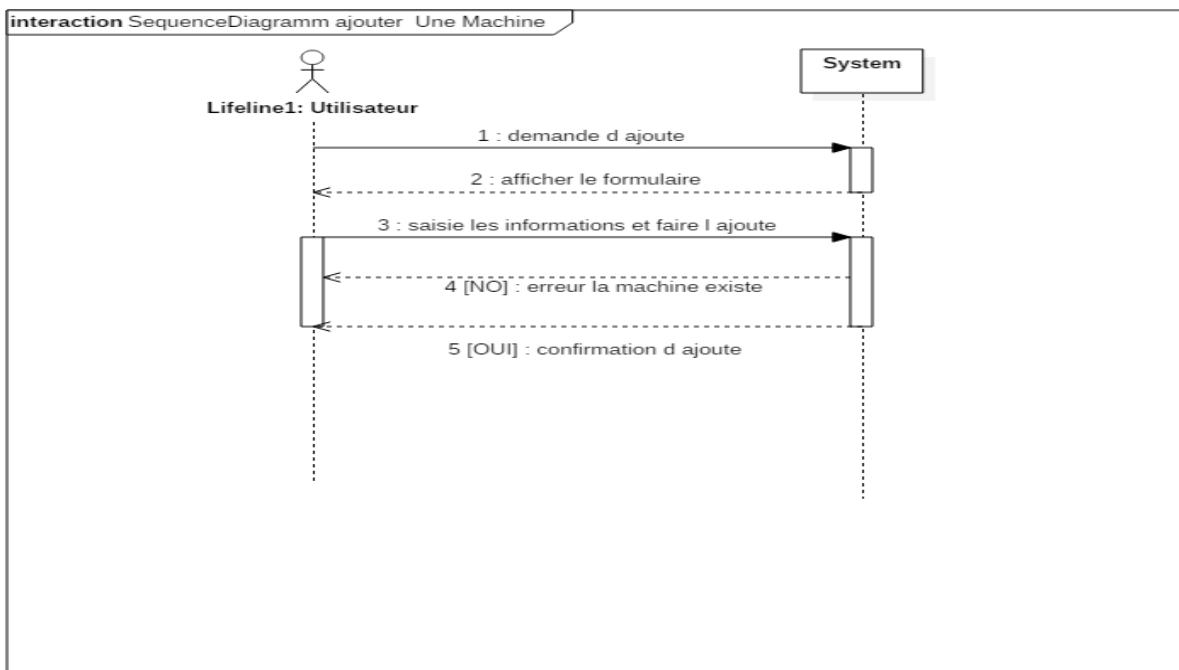


Figure 4.2 ajouter une machine

1.1.2.2 diagramme de séquence ajouter une maintenance

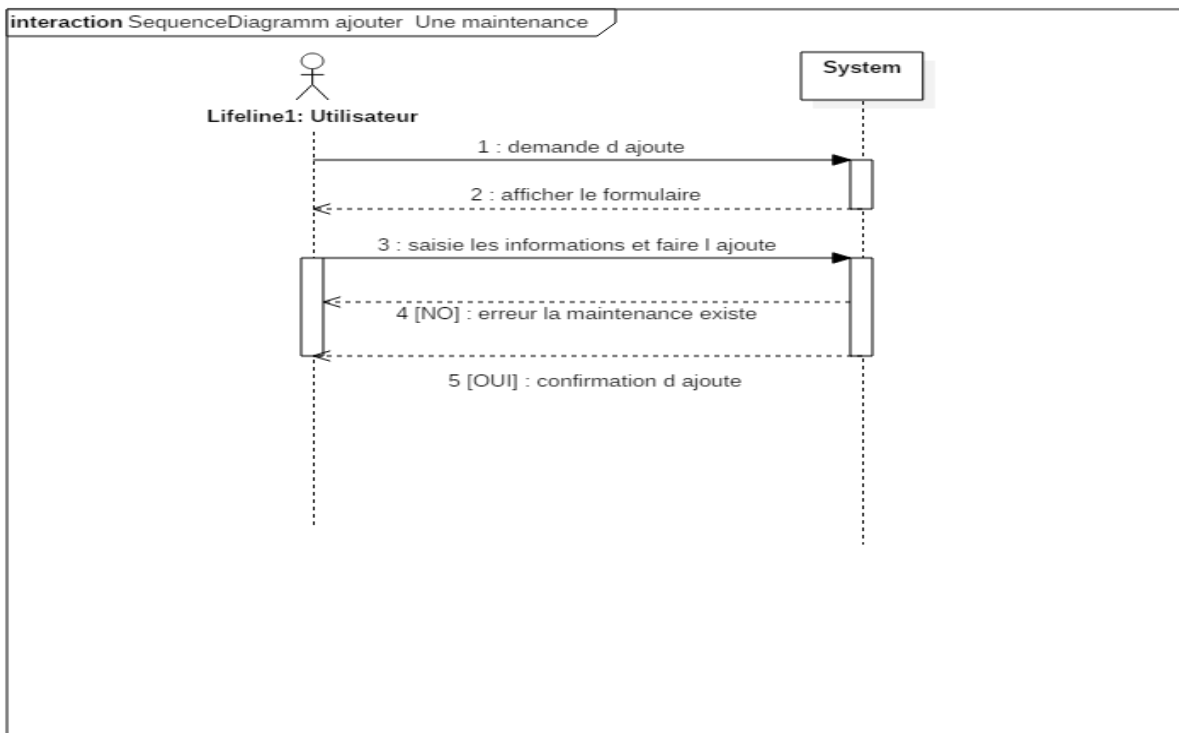


Figure 4.3 ajouter une maintenance

1.1.2.3 diagramme de séquence ajouter une panne

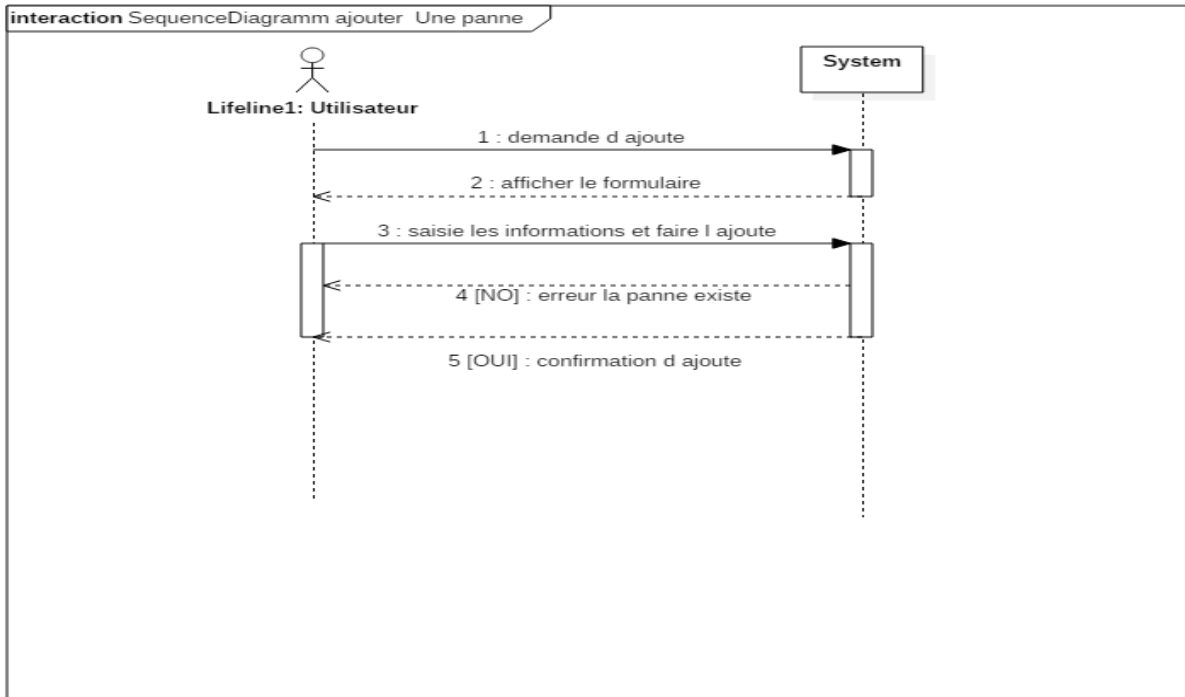


Figure 4.4 ajouter une panne

1.1.2.4 diagramme de séquence ajouter une défaillance

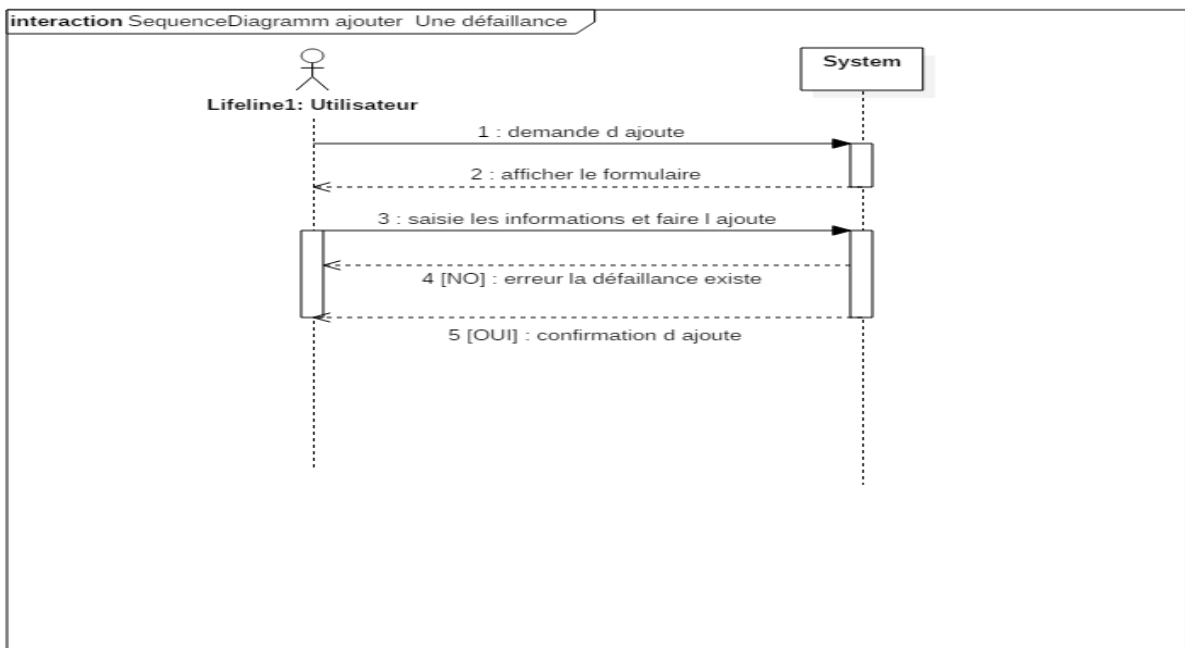


Figure 4.5 ajouter une défaillance

1.1.2.5 diagramme de séquence ajouter une analyse

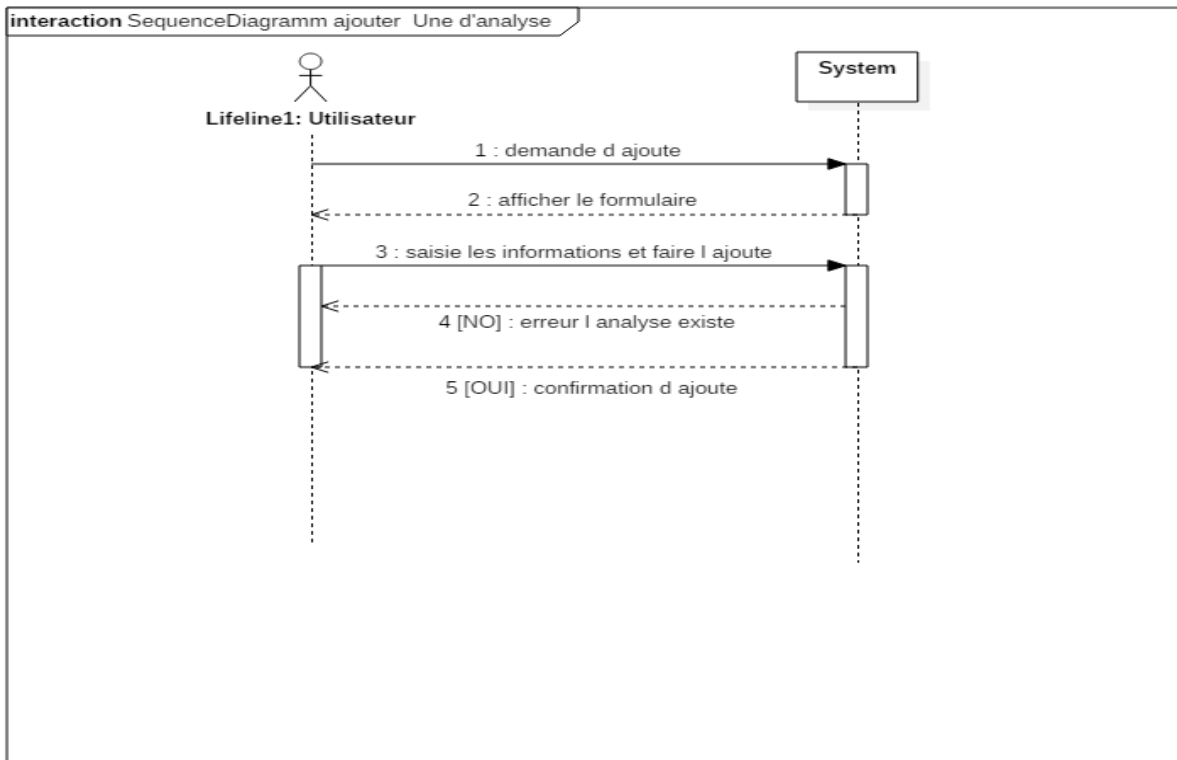


Figure 4.6 ajouter une analyse

1.1.2.6 diagramme de séquence modifier une machine

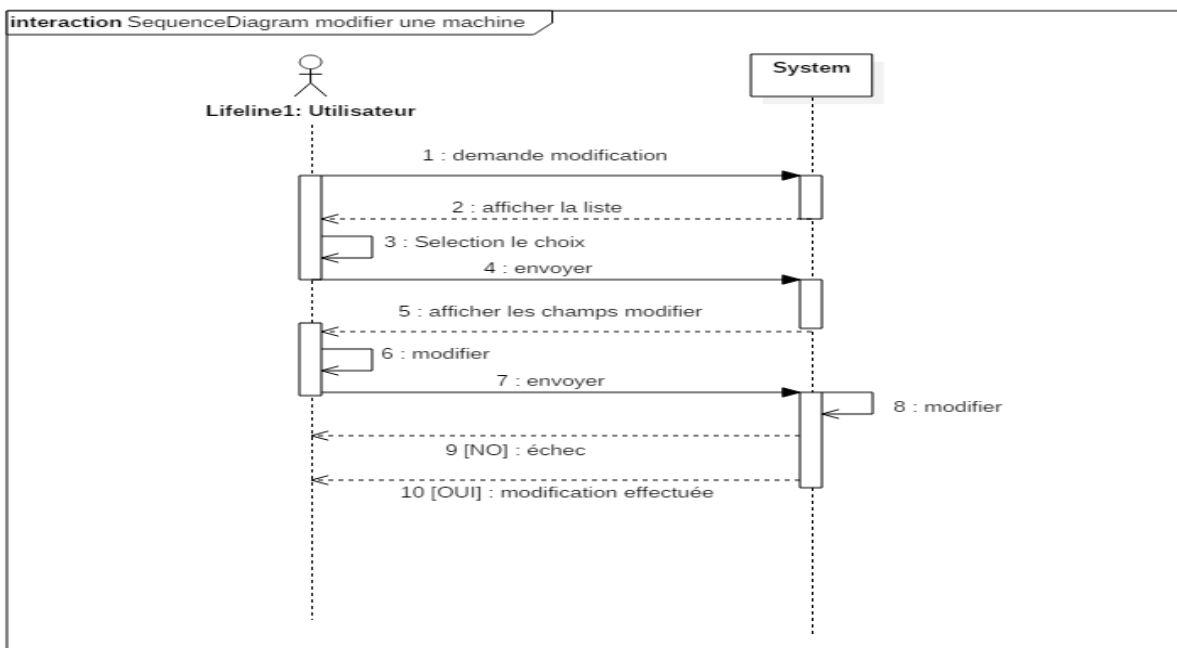


Figure 4.7 modifier une machine.

1.1.2.7 diagramme de séquence modifier une maintenance

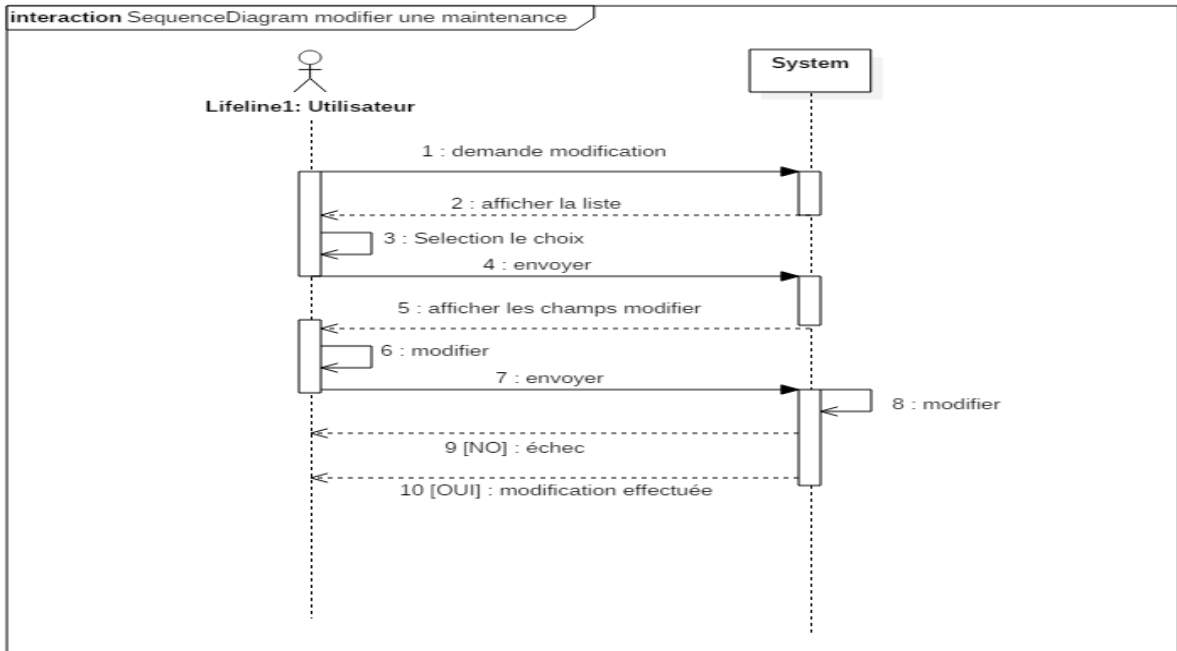


Figure 4.8 modifier une maintenance

1.1.2.8 diagramme de séquence modifier Une panne

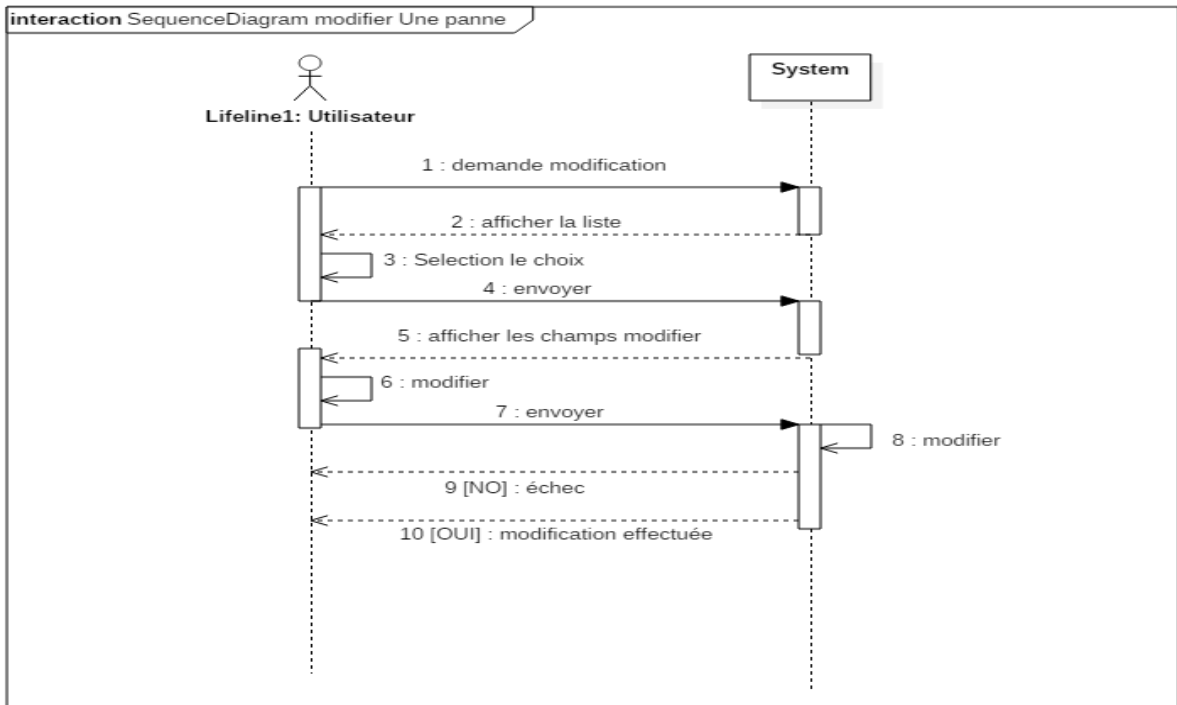


Figure 4.9 modifier une panne

1.1.2.9 diagramme de séquence modifier une défaillance

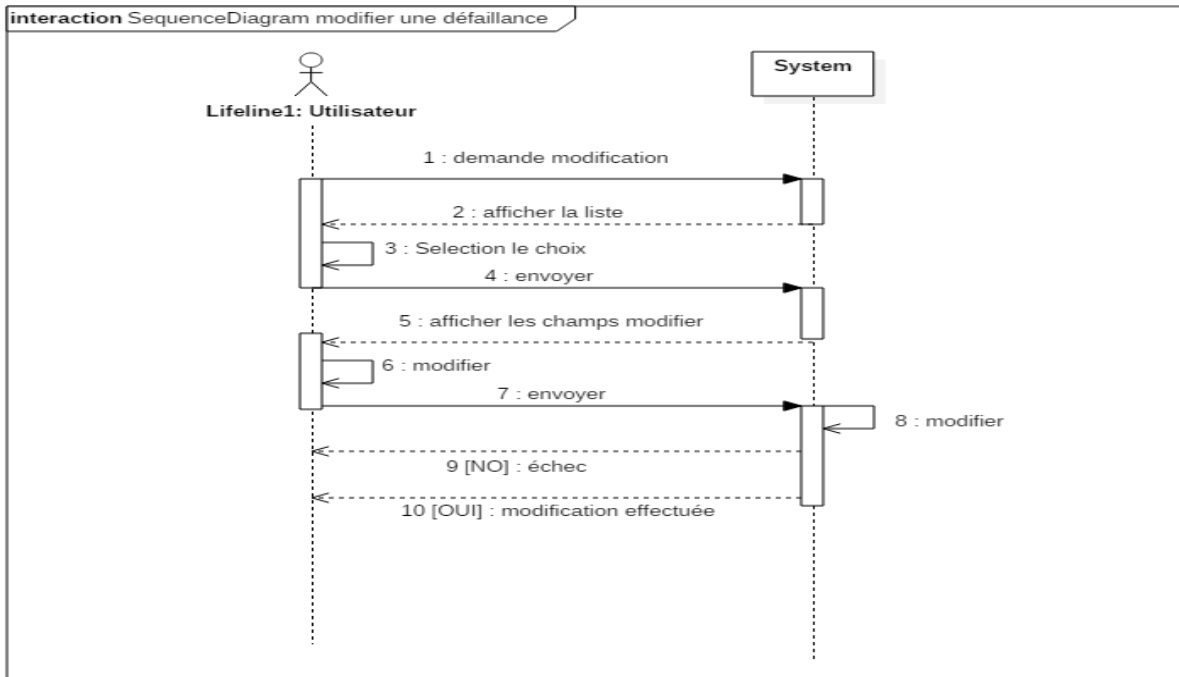


Figure 4.10 modifier une défaillance

1.1.2.10 diagramme de séquence modifier une d'analyse

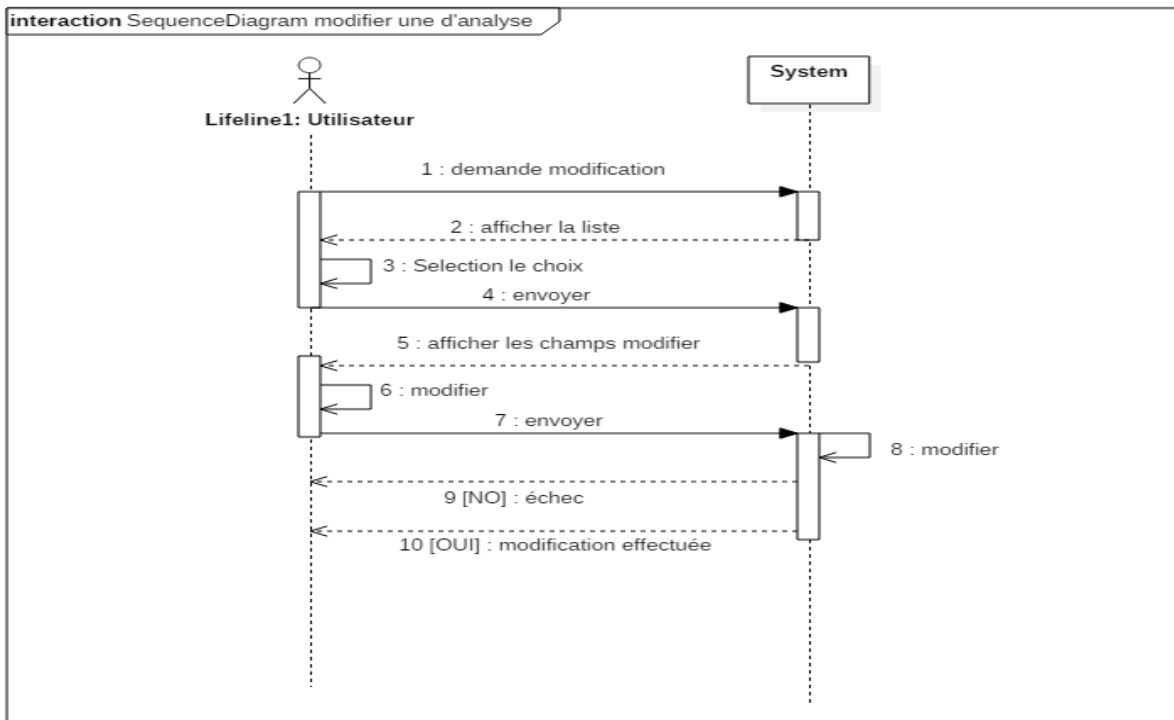


Figure 4.11 modifier une d'analyse

1.1.2.11 diagramme de séquence recherche une machine

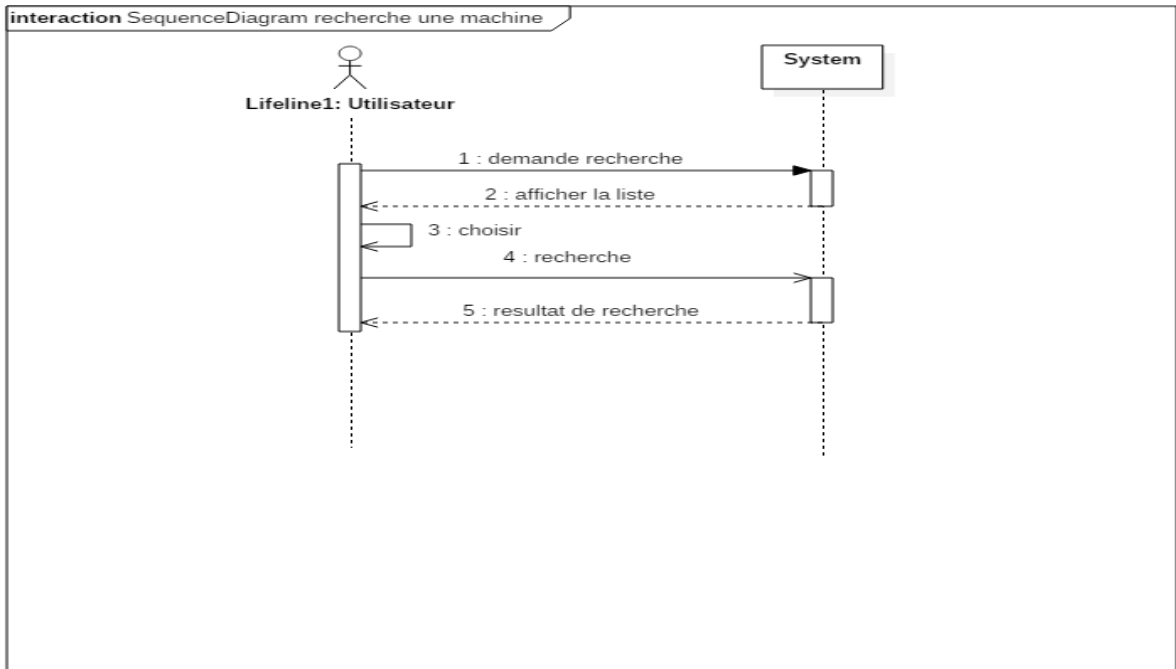


Figure 4.12 recherche une machine

1.1.2.12 diagramme de séquence recherche une maintenance

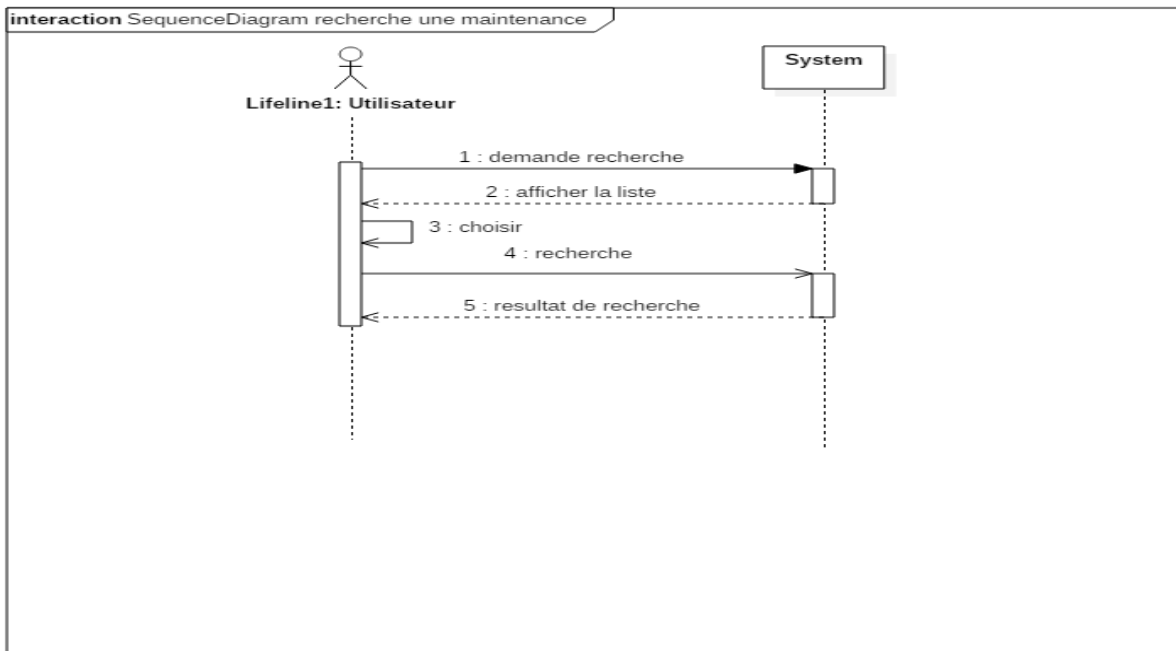


Figure 4.13 recherche une maintenance

1.1.2.13 diagramme de séquence recherche une panne

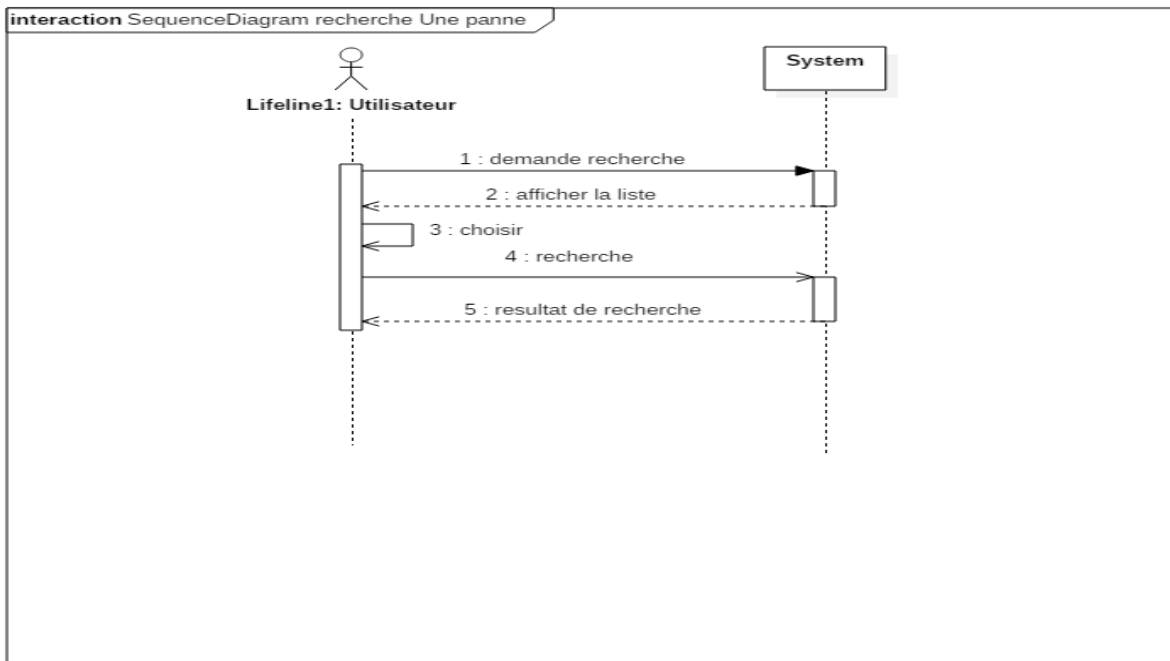


Figure 4.14 recherche Une panne

1.1.2.14 diagramme de séquence recherche une défaillance

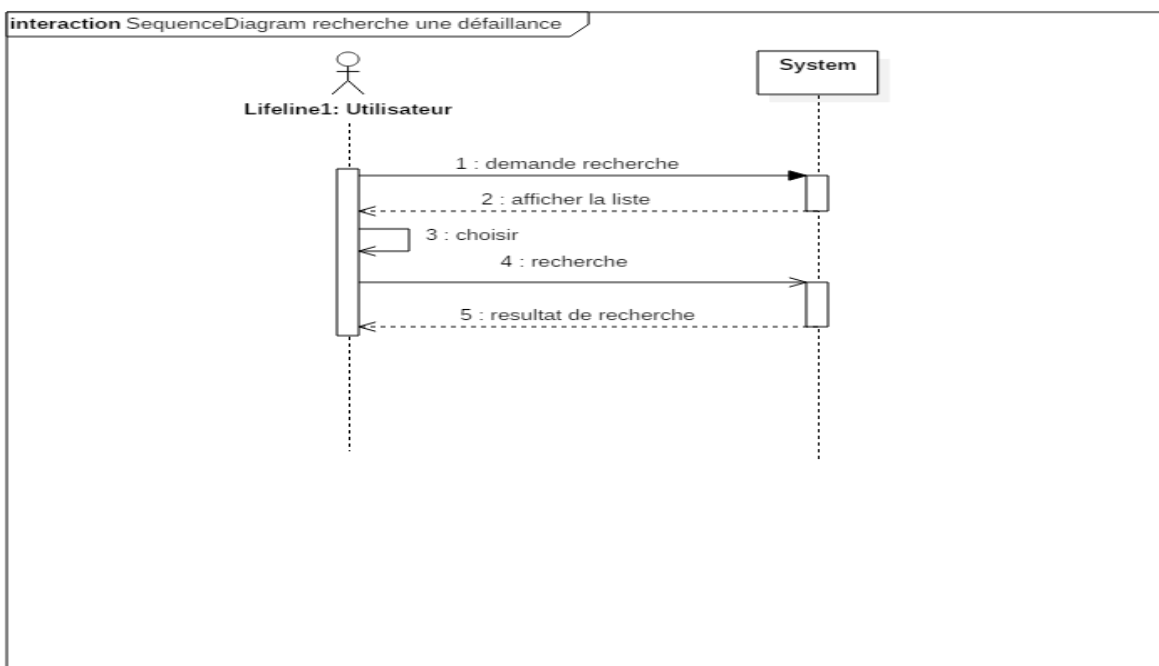


Figure 4.15 recherche une défaillance

1.1.2.15 diagramme de séquence recherche Une d'analyse

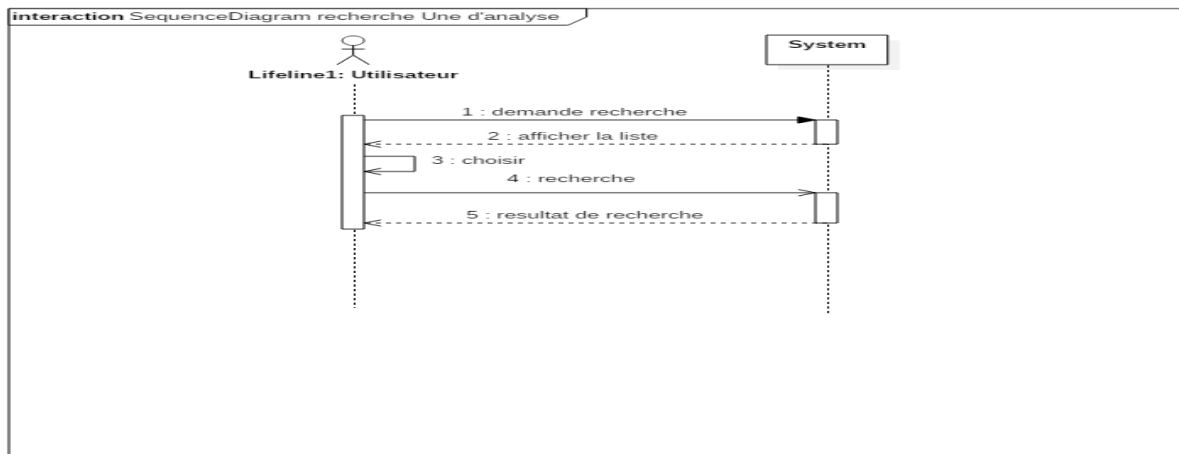


Figure 4.16 recherche Une d'analyse

1.2 Conceptualisation

Cette étape permet d’aboutir à un modèle informel, donc sémantiquement ambiguë et généralement exprimé en langage naturel. Elle consiste, à partir des données brutes à dégager les concepts et les relations entre ces concepts permettant de décrire de manière informelle les entités cognitives du domaine.[30]

- Construction du glossaire de termes
- Détermination de la taxonomie de concepts
- Construction d'un diagramme des relations binaires
- Construction d'un dictionnaire des concepts
- Construction du tableau des relations binaires
- Construction du tableau des attributs d'instances
- Construction du tableau des attributs de classe

1.2.1 Construction du glossaire de termes

Nom du concept	Description	Synonyme	Terme en anglais	Type
machine	est une machine qui souffre d'une panne. Il a besoin d'un traitement.		Machine	Classe
maintenance	La maintenance est. le processus de protection de la machine contre la panne.		Maintenance	Classe
panne	La panne est une altération des fonctions ou de la maintenance des machines . se traduisant par des défaillance. elle concerne un ou plusieurs machines.		breakdown	Classe

défaillance	Un critère que nous pouvons évaluer est risque qui l'entoure la machine.		failure	Classe
analyse	Est une analyse basée sur la machine.		Analysis	Classe
électrique	est une type de machine a base électrique		Electric	Sous Classe
mécanique	est une type de machine a base énergie mécanique		Mechanical	Sous Classe
électronique	est une type de machine a base énergie électronique		electronic	Sous Classe
hydraulique	est une type de machine a base énergie hydraulique		Hydraulic	Sous Classe
surcharge	Est un type de panne électrique		Overload	Sous Classe
déformation de rotor	est un type de panne mécanique		rotor deformation	Sous Classe
Bruit	est un type de Défaillance		noise	Sous Classe

Table 4.1 Glossaire de termes

1.2.2 Détermination de la taxonomie de concepts

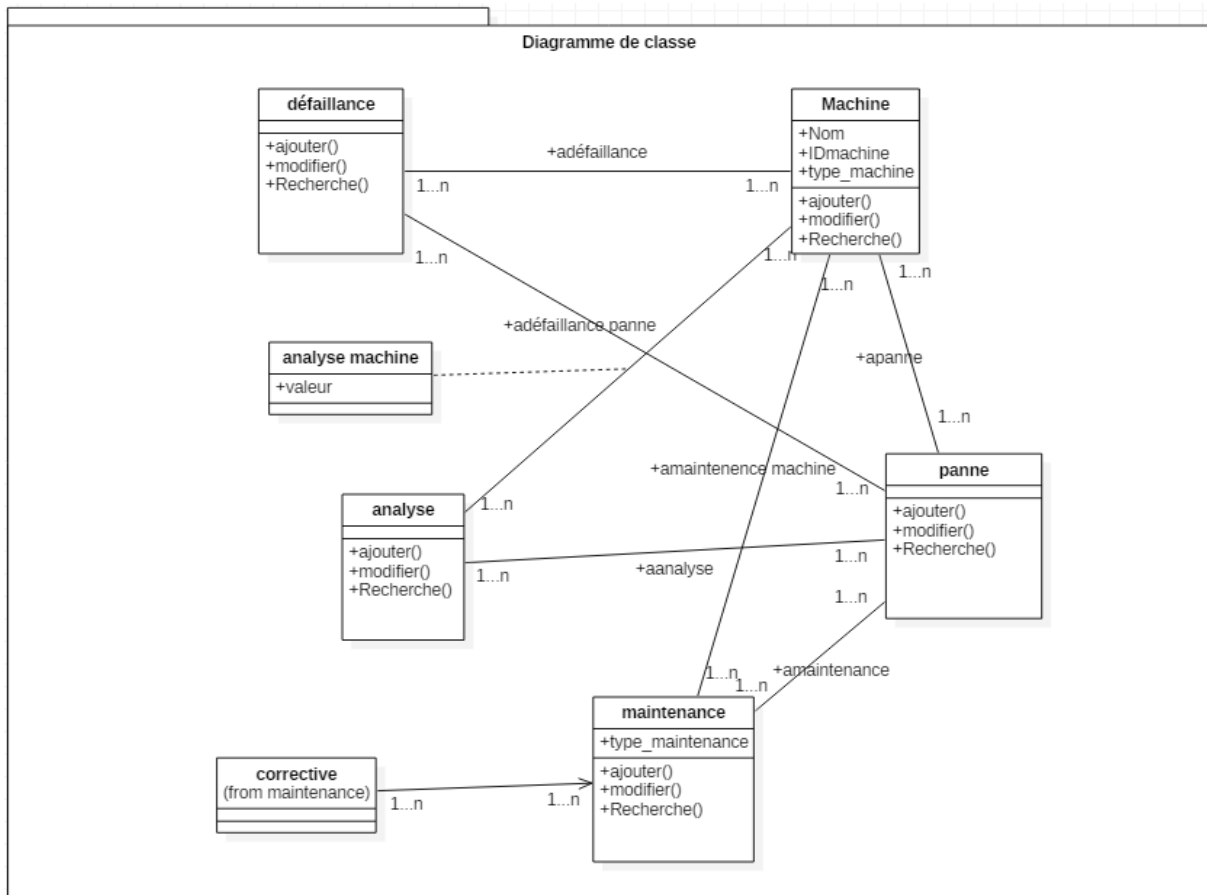


Figure 4.17 Diagramme de classe

1.2.4 Construction d'un dictionnaire des concepts

Nom de concept	Instances	Attributs de classe	Relations
machine	Mach1 Mach2 . .	nom IDmachine type_machine	adéfaillance apanne amaintenance_machine
défaillance	Dif1 Dif2 . .		adéfaillance adéfaillance_panne
panne	Pnn1 Pnn2 . .		adéfaillance_panne aanalyse apanne amaintenance
analyse	Analys1 Analys2 . .	valeur	aanalyse
maintenance	Maint1 Maint2 . .	nom type_maintenance	Amaintenance amaintenance_machine

Table 4.2 dictionnaire des concepts

1.2.5 Construction du tableau des relations binaires

Nom de relation	Concept source	Concept cible	Cardinalité	Relation inverse
apanne	machine	panne	1...n	
adéfaillance	machine	défaillance	1...n	
aanalyse	panne	analyse	1...n	
adéfaillance-panne	défaillance	panne	1...n	
amaintenance	panne	Maintenance	1...n	
analyse_machine	machine	analyse	1...n	
amaintenance_machine	machine	maintenance	1...n	

Table 4.3 tableau des relations binaires

1.2.6 Construction du tableau des attributs d'instances

Nom de l'instance	Concept concerné	L'attribut	Type de valeur
Mach1	machine	nom IDmachine type_machine	String Int

Mach2	machine	nom IDmachine type_machine	String Int
Analys1	analyse	valeur	Float
Analys2	analyse	valeur	Float
Maint1	maintenance	type_maintenance	String
Maint2	maintenance	type_maintenance	String

Table 4.4 tableau des attributs d'instances

1.2.7. Construction du tableau des attributs de classe

Nom attribut	Concept concerné	Type de valeurs	Cardinalité(min/max)
Nom	machine	String	(1.20)
IDmachine	machine	Int	(1.n)
Valeur	analyse	Float	(1.n)
type_maintenance	maintenance	String	(1.n)
type_machine	machine	String	(1.n)

Table 4.5 tableau des attributs de classe

1.3 Table d'analyse des panne méthode (AMDEC)

Tableau d'analyse des pannes											AMDEC
Machine :							Date d'analyse :				N°
Composant	Fonction	Panne	Cause	Effet	Détection	Temps	Criticité				Actions
							Fréquence	Gravité	Détection	Criticité	
Quels composants de l'équipement étudions nous ?	A quoi sert ce composant	Quels sont les différentes pannes que peut connaître le composant ?	Qu'est-ce qui peut provoquer cette panne ?	Quels effets a chaque panne ?	Comment remarque-t-on qu'il va y avoir une panne ?	Combien de temps faut il pour réparer ?	Cette panne apparaît-elle souvent ?	La panne est-elle grave ? Longue ? Dangereuse? Polluante ?	Peut-on pressentir que cette panne va arriver ?	Produit des 3 colonnes précédentes	Que peut-on faire pour baisser la criticité ?

Table 4.6 analyse des pannes méthode (AMDEC)

1.4 La formalisation

Pour la logique de description T-BOX et A-BOX alors :

Machine \subseteq T \cap (nom de machine. String) \cap (type de machine. String) \cap (IDmachine de machine.int).

Mécanique \subseteq machine

Hydraulique \subseteq machine

Electronique \subseteq machine

Pneumatique \subseteq machine

la Défaillance $\subseteq \top \cap$ (nom de défaillance. String) \cap (type de défaillance. String)

bruit \subseteq Défaillance

vibration \subseteq Défaillance

échauffement \subseteq Défaillance

L'apparition de la fumée \subseteq Défaillance

faible pression \subseteq Défaillance

Non fonctionnement \subseteq Défaillance

fonctionnement dégradé \subseteq Défaillance

plus de fonctionnement \subseteq Défaillance

fonctionnement non requise \subseteq Défaillance

la panne $\subseteq \top \cap$ (nom de la panne. String) \cap (type de panne , String)

Panne- Pneumatique \subseteq panne

Panne –Electronique \subseteq panne

Panne- Mécanique \subseteq panne

Panne- Electrique \subseteq panne

Panne- Hydraulique \subseteq panne

déformation de rotor \subseteq Panne- Mécanique

détérioration de roulement \subseteq Panne- Mécanique

pipes bloqué \subseteq Panne- Pneumatique

Filtre \subseteq Panne- Pneumatique

compresseur \subseteq Panne- Pneumatique

Circuit intégré \subseteq Panne –Electronique

les capteurs \subseteq Panne –Electronique

Distributeur bloqué \subseteq Panne- Hydraulique

Electrovanne bouché \subseteq Panne- Hydraulique

Surcharge \subseteq Panne- Electrique

Court-circuit \subseteq Panne- Electrique

la analyse \subseteq $\top \cap$ (nom de la analyse. String) \cap (Valeur de l'analyse .float)

AMDEC \subseteq analyse

AMD \subseteq analyse

Pareto \subseteq analyse

Ishikawa \subseteq analyse

Perte \subseteq analyse

Courbe abc \subseteq analyse

La Maintenance \subseteq $\top \cap$ (nom de la maintenance . String) \cap (type de Maintenance . String)

Correctiv \subseteq Maintenance.

1.5 Implémentation

Notre choix porte sur le langage OWL qui est un langage de codification utilisé pour implémenter l'ontologie en OWL (un langage de définition d'ontologie pour le Web), et cela pour toutes les fonctionnalités sémantiques que permet OWL et qui sont plus riches que celles des langages RDFS & DAML+OIL.[14]

1.5.1 Langages de programmation

Nous avons utilisé le langage JAVA et la bibliothèque Jena.

1.5.1.1 Jena

JENA est une bibliothèque java open-source et gratuite pour le Web sémantique. Elle permet les manipulations courantes des structures sémantiques écrites dans les langages spécifiés par le W3C comme standards pour ces structures :

RDF, RDFS, OWL et DAML + OIL. JENA est actuellement le cadre de travail idéal pour la plupart des équipes de recherche ou des développeurs pour le Web sémantique. La raison en est qu'elle respecte le mieux les normes énoncées par le W3C quant à l'écriture des méta-données et des structures comme les ontologies. JENA bénéficie d'une équipe de développeurs dynamiques qui n'ont cessé de l'améliorer.[11]

Le Figure suivant est présenter l'interface de JENA

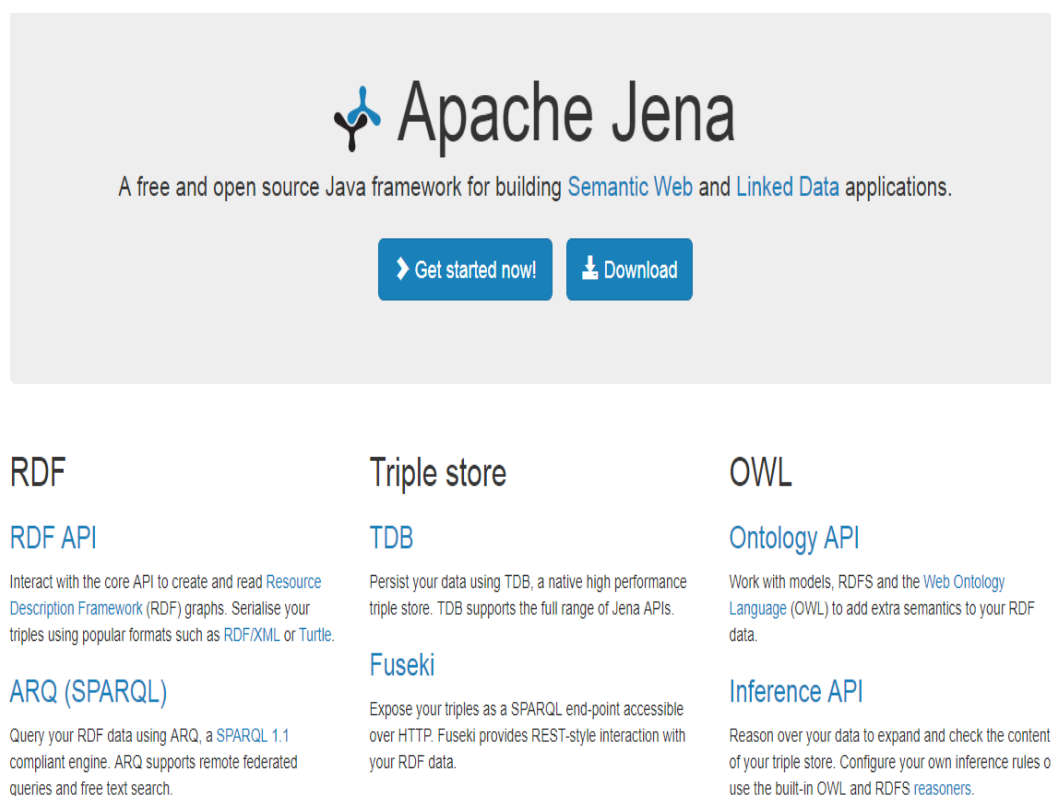


Figure 4.19 l'interface de JENA

1.5.1.2 Java

Le langage Java est un langage de programmation informatique orienté objet créé par James Gosling et Patrick Naughton, employés de Sun Microsystems, avec le soutien de Bill Joy. Il fut présenté officiellement en 1995. Selon les développeurs de Sun, Java (qui signifie café en argot américain) est un langage : simple, orienté-objet, distribué, interprété, robuste sécurisé, neutre vis à vis de l'architecture, portable, à haute performance, multi-threaded et dynamique.

Le langage Java était à la base un langage pour Internet, pour pouvoir rendre plus dynamiques les pages (tout comme le JavaScript aujourd'hui). Mais le Java a beaucoup évolué et est devenu un langage de programmation très puissant permettant de presque tout faire. Contrairement à la plupart des autres langages (sauf la plateforme .Net), Java met à la disposition du développeur une API très riche lui permettant de faire de très nombreuses choses. Il existe plusieurs IDE (Integrated Development Environment) pour le langage JAVA par exemple Eclipse, JBuilder et NetBeans que nous avons utilisé.[30]

La Figure (4.20) cet un l'interface de NetBeans.

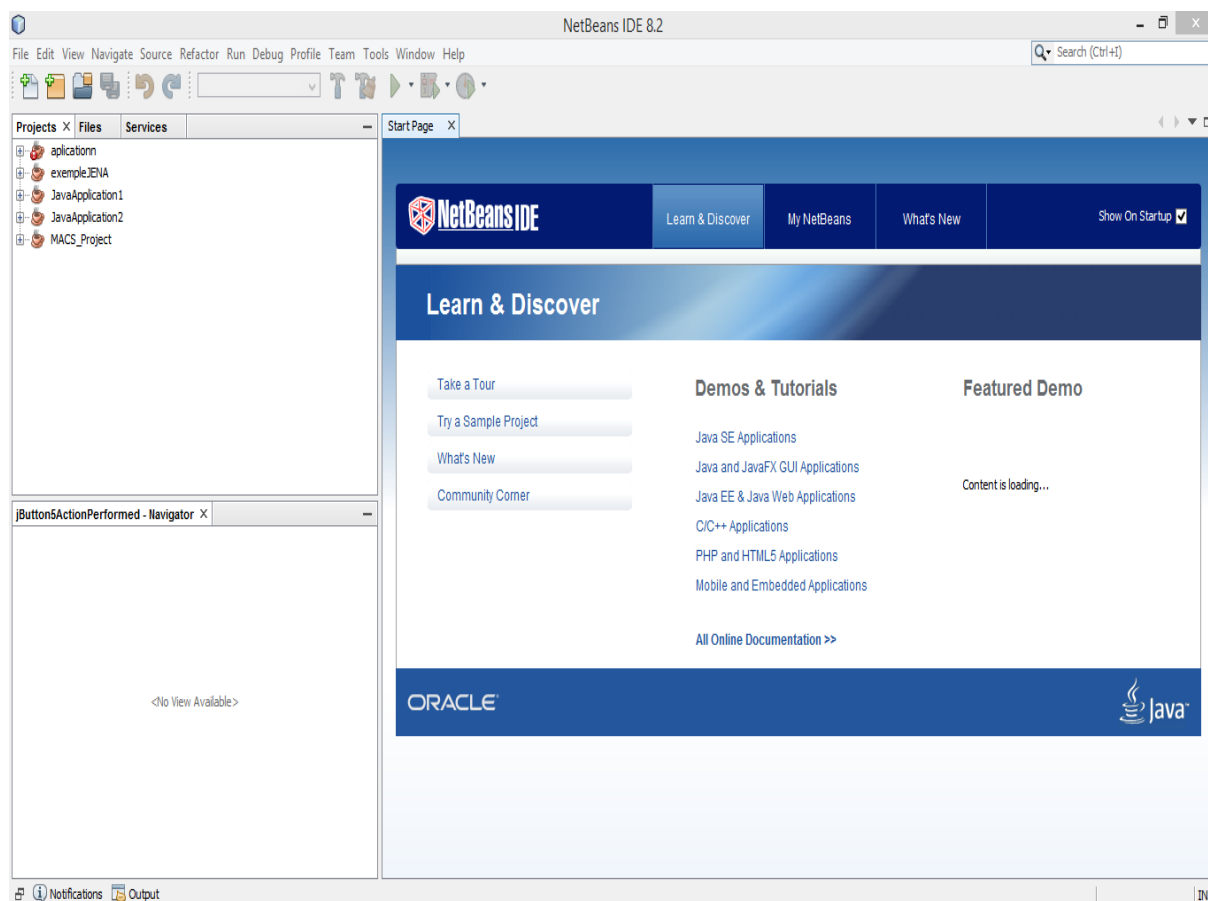


Figure 4.20 l'interface de NetBeans

1.5.2 Outils de programmation

Nous avons utilisé Protégé et NetBeans IDE.

1.5.2.1 Protégé

Protégé est une plateforme open-source qui propose au travers d'une communauté grandissante une suite d'outils de construction de modèles de domaine. Au delà de cette communauté, l'outil Protégé permet la création d'ontologies et la possibilité de modifier l'outil à convenance afin d'ajouter diverses fonctionnalités spécifiques. Plusieurs travaux portent sur l'intégration au sein de l'outil divers plugins permettant des traitements spécifiques. On notera par exemple les travaux de l'équipe de A. Tissaoui sur une méthodologie d'évolution d'une ressources termino-ontologique : EvOnto (Tissaoui, 2010) proposant une méthode formalisée au travers d'un plugin pour Protégé. L'ontologie étant considéré comme stable mais cependant voué à modification selon l'évolution des ressources termino-ontologique. Ce plugin propose une assistance à la modification des ressources.[29]

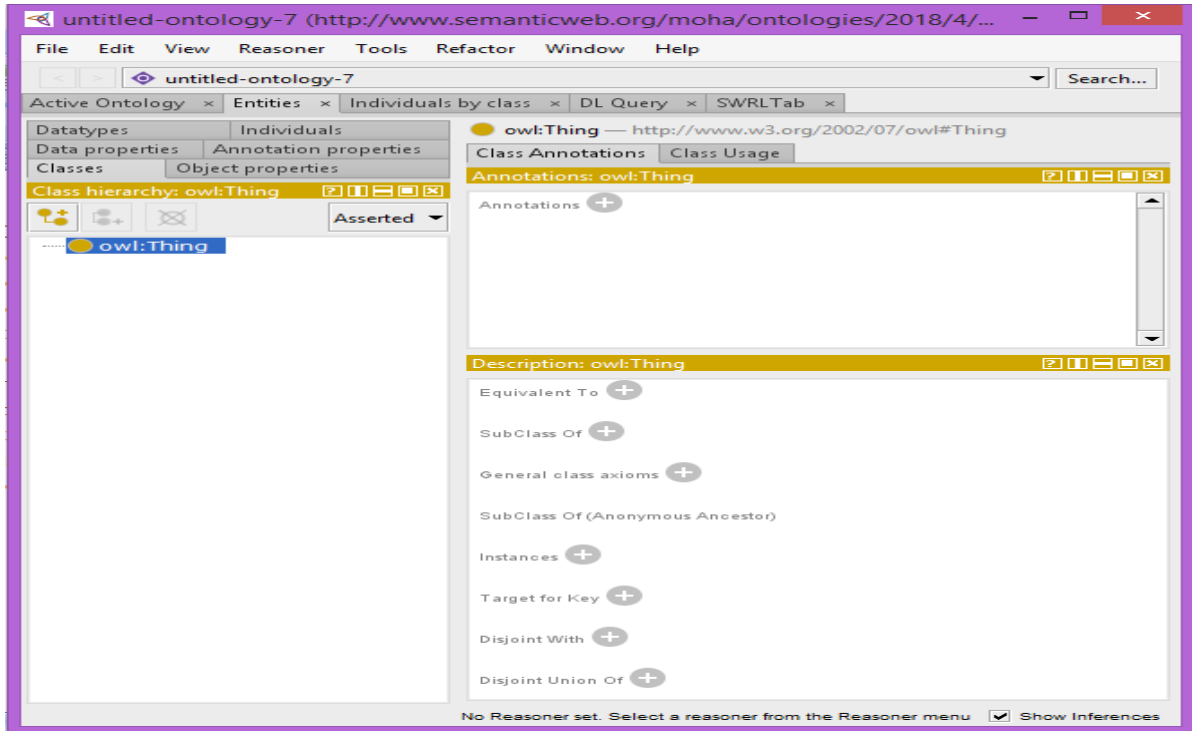


Figure 4.21 Interface graphique de Protégé 5.2.0

Création d'une classe:

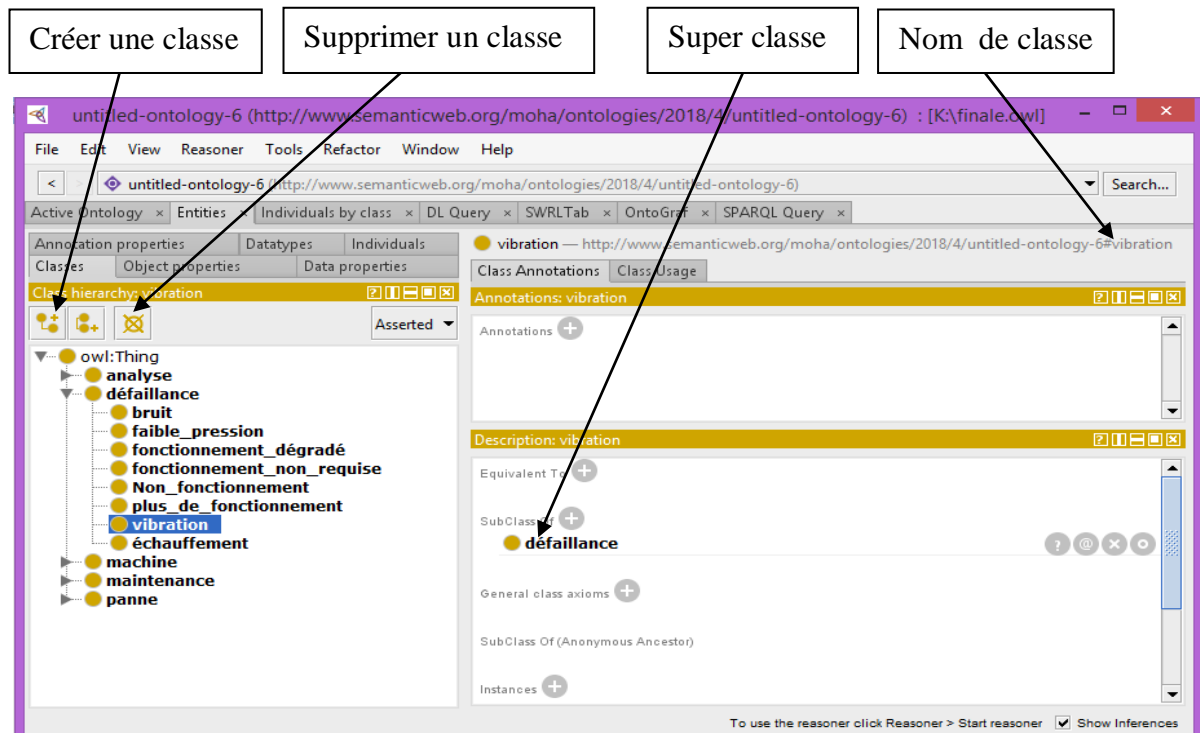


Figure 4.22 Création des classes

Création d'une relation:

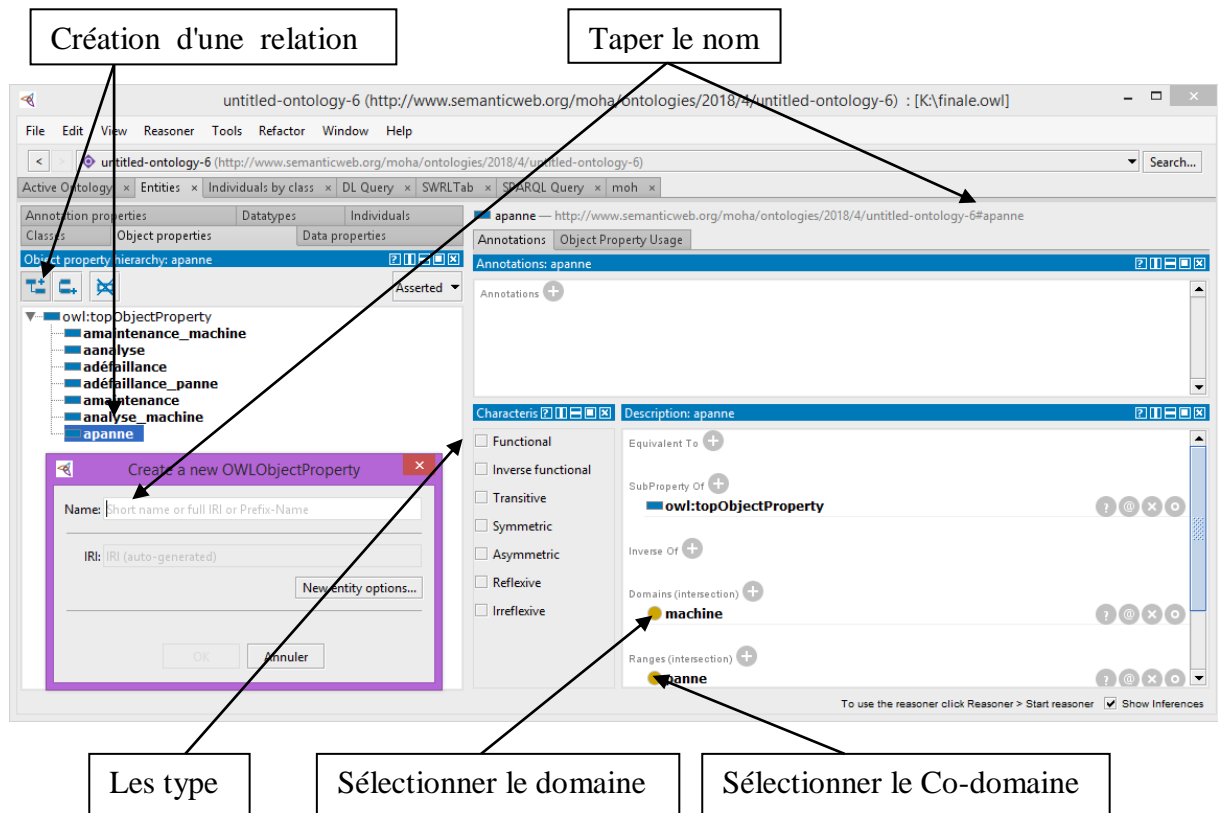


Figure 4.23 Création d'une relation.

Création des instances:

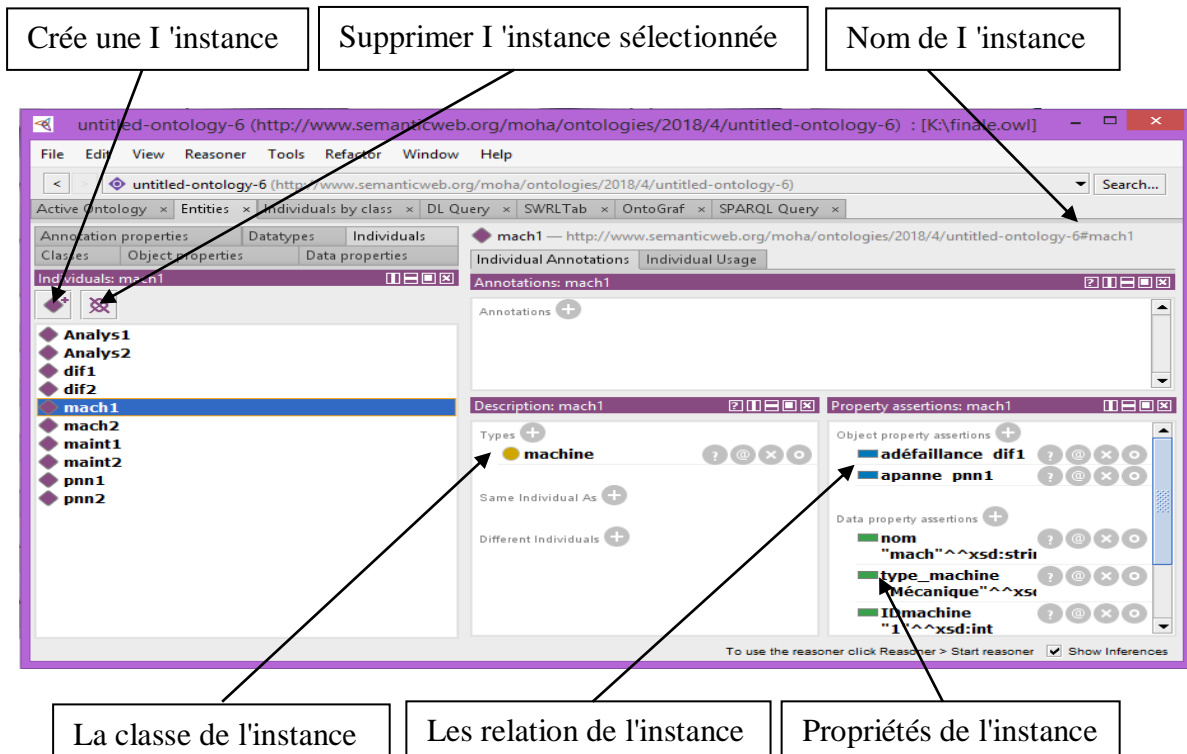


Figure 4.24 Création des instances.

1.5.2.2 NetBeans

NetBeans est un environnement de développement intégré (EDI), placé en open source en juin 2000 sous licence CDDL (Common Développement and Distribution License). En plus de Java, NetBeans permet également de supporter différents autres langages, comme C, C++, JavaScript, XML, et HTML. Il comprend toutes les caractéristiques d'un IDE moderne (éditeur en couleur, projets multi-langage, éditeur graphique d'interfaces et de pages Web).

Conçu en Java, NetBeans est disponible sous Windows, Linux, ou sous une version indépendante des systèmes d'exploitation (requérant une machine virtuelle Java). NetBeans constitue par ailleurs une plate-forme qui permet le développement d'applications spécifiques (bibliothèque Swing (Java)). L'IDE NetBeans s'appuie sur cette plate-forme, il s'enrichit à l'aide de plugins.[21]

1.5.3 Langages web sémantique

1.5.3.1 OWL

Pour l'implémentation de nos ontologies, nous avons opté pour l'éditeur d'ontologies

Protégé-OWL Plusieurs raisons ont motivé notre choix:

*Protégé-OWL est un éditeur open source et gratuit.

*Protégé-OWL permet d'importer et d'exporter des ontologies dans les différents langages d'implémentation d'ontologies (RDF- Schema, OWL, DAML, OIL,... etc.)

*Protégé-OWL possède une interface modulaire, ce qui permet son enrichissement par des modules additionnels (plugins).

*Protégé-OWL permet l'édition et la visualisation d'ontologies.

*Protégé-OWL permet le contrôle de la cohérence de l'ontologie par des vérifications de contraintes.

*Protégé-OWL est fourni avec une API écrite en JAVA, qui permet de développer des applications pouvant accéder aux ontologies de Protégé et de les manipuler.

la figure suivante permet d'afficher l'ontologie sous forme OWL.[20]

```

<?xml version="1.0"?>
<Ontology xmlns="http://www.w3.org/2002/07/owl#"
  xml:base="http://www.semanticweb.org/moha/ontologies/2018/4/untitled-ontology-6"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:xml="http://www.w3.org/XML/1998/namespace"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  ontologyIRI="http://www.semanticweb.org/moha/ontologies/2018/4/untitled-ontology-6">
  <Prefix name="owl" IRI="http://www.w3.org/2002/07/owl#" />
  <Prefix name="rdf" IRI="http://www.w3.org/1999/02/22-rdf-syntax-ns#" />
  <Prefix name="xml" IRI="http://www.w3.org/XML/1998/namespace" />
  <Prefix name="xsd" IRI="http://www.w3.org/2001/XMLSchema#" />
  <Prefix name="rdfs" IRI="http://www.w3.org/2000/01/rdf-schema#" />
  <Prefix name="swrl" IRI="http://www.w3.org/2003/11/swrl#" />
  <Prefix name="swrla" IRI="http://swrl.stanford.edu/ontologies/3.3/swrla.owl#" />
  <Prefix name="swrlb" IRI="http://www.w3.org/2003/11/swrlb#" />
  <Prefix name="untitled-ontology-6" IRI="http://www.semanticweb.org/moha/ontologies/2018/4/untitled-ontology-6#" />
  <Declaration>
    <Class IRI="#maintenance" />
  </Declaration>
  <Declaration>
    <Class IRI="#Panne_Electrique" />
  </Declaration>
  <Declaration>
    <ObjectProperty IRI="#adéfaillance" />
  </Declaration>
  <Declaration>
    <Class IRI="#Non_fonctionnement" />
  </Declaration>
  <Declaration>
    <DataProperty IRI="#nom" />
  </Declaration>
  .....

```

Figure 4.25 exemple OWL.

1.5.3.2 SPARQL

Exemple de requête pour afficher tout les classes et sous classes

```
SELECT DISTINCT ?subject ?object
WHERE { ?subject rdfs:subClassOf ?object.
        ?object rdf:type owl:Class }
```

```
SELECT ?machine ?nom
WHERE { ?machine rdf:type rdf:machine. }
```

1.5.3.3 SWRL-Tab

```
machine(?x) ^ adéfaillance(?x, ?y) ^ bruit(?y) ^ apanne(?x, ?z) ->
détérioration_de_roulement(?z).
```

```
machine(?x) ^ adéfaillance(?x, ?y) ^ faible_pression(?y) ^ apanne(?x, ?z) ->
filtre_bouchée(?z).
```

```
machine(?x) ^ adéfaillance(?x, ?y) ^ plus_de_fonctionnement(?y) ^ apanne(?x, ?z) ->
surcharge(?z)
```

1.6 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 LDS, et qui sont offerts par des raisonneurs.[30]

2 L'interface de l'application

Partie de l'interface d'application

The screenshot shows a Java application window with a light gray background and a purple title bar. The window contains a form with the following elements:

- Nom:** A text input field.
- IDmachine:** A text input field.
- type mach:** A dropdown menu currently showing 'mécanique'.
- ajouter:** A button located to the right of the IDmachine field.
- Rechercher:** A button followed by a text input field.
- Supprimer:** A button located below the Rechercher field.
- modifier:** A button followed by a text input field.
- infirer:** A button followed by a text input field, a dropdown menu (options: 'bruit'), and a text area.
- infirer:** A button followed by a text input field, a dropdown menu (options: 'AMDEC'), and a text area.
- infirer:** A button followed by a text input field, a dropdown menu (options: 'calliative'), and a text area.
- Actualiser:** A button at the bottom center.
- annuler:** A button at the bottom right.

Figure 4.25 une partie d'application.

3 Conclusion

Dans ce chapitre nous avons pu appliquer diverses techniques et méthodes afin de faciliter la création de notre travail, que nous avons étudié et utilisé le langage de Java à travers l'environnement NetBeans et utiliser le langage OWL pour fournir ontologie une base de connaissance où nous sauvegardons notre travail.

CONCLUSION GÉNÉRALE

Dans notre travail, nous devons assurer l'entretien de la machine d'une manière différente, nous utilisons le langage web sémantique pour fournir une bonne étude approfondie de notre problème, et nous avons utilisé dans notre travail méthode METHONTOLOGY, elle étudie en général (ontologie) et toutes ses formes.

Nous devons appliquer tous les éléments du problème auxquels nous sommes confrontés et en appuyant sur le système expert dans le but de Représenter et découvrir le problème.

Nous avons aussi utilisé plusieurs techniques afin de faciliter la solution de ce problème, nous avons utilisé le processus de techniques, y compris le langage protégé et aussi avec le langage Java NetBeans.

La maintenance de la machine a de nombreux types contribuent au bon fonctionnement de la machine et aussi à maintenir la production, qui est le principal facteur de la machine.

Pour l'édition de notre ontologie, on a utilisé Protégé qui est un éditeur très utilisé. Pour permettre le raisonnement sur l'ontologie, on a utilisé pellet qui est un moteur d'inférence très puissant.

il existe plusieurs perspectives possibles à notre travail :

- Enrichir notre ontologie avec d'autres concepts des machines, des panne et la relations entre eux .
- enrichir autant que possible notre physiologie avec l'aide d'experts de maintenance.
- SWRL est utilisé pour faciliter l'expression des règles.
- Contribuer autant que possible pour aider l'utilisateur et le mécanicien.

BIBLIOGRAPHIES

- [1] François-Noël CRES, de l'école Nationale Supérieure des MINES de PARIS et de l'école Nationale Supérieure des :MINES de SAINT -ETIENNE, pour obtenir le titre de Docteur, Ecole Nationale Supérieure des MINES de SAINT ETIENNE, 27 novembre 1989.
- [2] Fatime Ly Zineb Simeu-Abazi Jean-Baptiste Leger, Terminologie Maintenance : bilan, Groupe de Recherche S.P.S.F. : Groupe de travail Maintenance.
- [3] Dr. SAID KADRI, Introduction à l'Intelligence Artificielle, Chapitre III: Systèmes Experts et leurs Applications, Département of Computer Science, Faculté of Mathématiques and Informatiques, Université Mohamed Boudiaf of M'sila, 2017 - - - 2018
- [4] Thibaud Raby, Frédéric Ravaut, Utilisation d'ontologies dans une application médicale décisionnelle, ECE Paris École d'Ingénieur 37 Quai de Grenelle – 75015 Paris,2011.
- [5] Gael Kerbaol / INRS, Conception Et Utilisation Des Equipements De Travail, Santé et sécurité au travail, INRS 2018.
- [6] Lux innovation GIE , AMDEC = analyse des modes de défaillance de leurs effets et de leur criticité, National Agency For Innovation And Research, 7, rue alcide de gasperi, 1-1615 luxembourg
- [7] Mr S. Boughrara, Web Sémantique Master1 TIC, Chapitre V: ingénierie Ontologique.
- [8] Notes par Mr S. Boughrara, Web sémantique-3L /SI, Chapitre III : Langages WS, 2016/2017.
- [9] Boubakri Mohamed Lamine, Djaidja Omar Anas, Une approche d'amélioration du service maintenance basée sur les réseaux des files d'attente, Mémoire présenté pour l'obtention Du diplôme de Master Académique, Université Mohamed Boudiaf -M'sila, 2016 /2017.
- [10] Hachem Mohammed Chérif , Laimeche hadj abdellah, Présentation d'une gamme d'entretien préventive d'une fraiseuse de type 6 P 13, MEMOIRE DE FIN D'ETUDE En vue de l'obtention du diplôme Master maintenance industrielle, Université Kasdi Merbah–Ouargla Faculté des Sciences et Technologie et Science de la Matière, 2011.
- [11] Patrick Tchouanto Poosia, Modélisation Des Ontologies, Mémoire présenté Comme Exigence Partielle De La Maitrise En Informatique, Université Du Québec à Montréal, Juillet 2014.
- [12] Cherifi Mahfoudhe, étude et maintenance de treuil de forage OIL WELL 840E, Mémoire de fin d'étude pour l'obtention de diplôme Master en Génie mécanique option Maintenance Industrielle, Université Hassiba Ben Bouali De Chlef, 2012.
- [13] Belhadef Hacene, Système d'information pour l'aide à la décision spatiale basé sur une ontologie, THÈSE Présentée pour obtenir le diplôme de Doctorat en Sciences, Université Mentouri Constantine, 03 / 06 / 2010.

- [14] Afri Faiza, Raisonement sur une ontologie hybride pour la recherche d'informations médicales, Soutenance de Magister En Informatique, Université Abderrahmane Mira de Bejaia, 2008/2009.
- [15] Mathieu D'Aquin, un portail sémantique pou la gestion de connaissances en cancérologie, mémoire Présenté en vue de l'obtention du diplôme de doctorat, Université de Henri Poincaré Nancy 1 ,14 décembre 2005
- [16] Mme El Aini Shil Donia, Approche d'indexation sémantique guidée par une ontologie floue dans un système de recherche d'information, mémoire de Master en informatique, Université de Monastir, 2009 / 2010.
- [17] Hacine Gherbi Ahcine, Construction d'une ontologie pour le web sémantique, mémoire présenté pour l'obtention du diplôme de Magister, Université Ferhat Abbas-Sétif1, 2014.
- [18] Messalti Ahmed Zoubir, Hemila Abdelkader, Conception et réalisation d'un système de suivi et d'aide à la décision basé sur une ontologie dans le domaine de la mammographie, Master Académique, Université kasdi merbah ouargla , 2012/2013.
- [19] M. Dominique Lenne, Modélisation des connaissances et de l'interaction, Mémoire présenté pour obtenir une Habilitation à Diriger des Recherches, Université De Technologie De Compiègne, 3 décembre 2009.
- [20] Gherzouli Imane, Système d'aide au diagnostic médical à base d'ontologie, Mémoire présenté pour l'obtention Du diplôme de Master Académique, Université Mohamed Boudiaf -M'sila , 2016/2017
- [21] Belabbes Fatima, Conception et implémentation d'une ontologie médicale Cas : insuffisance cardiaque, Mémoire de fin d'études pour l'obtention du diplôme de Master en Informatique, Université Abou Bakr Belkaid– Tlemcen,2014/2015
- [22] Sandrine Mougang Kouamo, Intégration Du Web Sémantique Dans Un Systeme D'aide à La Décision Pour Le Génie Logiciel, Mémoire présenté Comme Exigence Partielle De La Maitrise En Informatique, Université Du Québec à Montréal, Octobre 2014.
- [23] Riad Lekhchine, Construction d'une ontologie pour le domaine de la sécurité : Application aux agents mobiles, mémoire pour l'obtention du diplôme de Magister en informatique, Université Mentouri – Constantine,2008-2009.
- [24] Ben Hebreche Halima, Proposition D'une Ontologie Formelle Pour La Modélisation Et La Simulation Intelligente, Mémoire Présenté en vue d'obtenir le diplôme de Magister en informatique, Université Kasdi Merbah Ouargla, 28/06/2012.
- [25] Pierre Sidoine V. Donfack Guefack, Modélisation des signes dans les ontologies biomédicales pour l'aide au diagnostic, Université Rennes 1, 2013.

[26] Boucetta Zouhel, Appariement Sémantique Des CVs/OFFRES D'emploi Dans Le Cadre Du E-Recrutement, Mémoire Présenté en vue de l'obtention du diplôme de Magister en informatique, Université Mentouri de Constantine, 2008.

[27] Saida Gharbi, Appariement d'ontologies hétérogènes, memoire Présenté en vue de l'obtention du diplôme de Magister, Université Badji Mokhtar-Annaba, 2011.

[28] Mr. BELOUADAH Abdenaceur, Amélioration de la fiabilité d'un système électromécanique par l'utilisation des opérations de la maintenance préventive, Mémoire De Fin D'étude En Vue De L'obtention Du Diplôme De Master En Génie Electrique, Université Mohamed Boudiaf - M'sila, 2015/2016.

[29] Jonathan Vigneron, Contribution des ontologies à la création de bases de connaissances pour la maîtrise des conformités réglementaires en santé, sécurité au travail et environnement, Doctorat ParisTech, l'École nationale supérieure des mines de Paris, 18 décembre 2013.

WEB GRAPHIES

[30] https://pmb.univ-saida.dz/butecopac/doc_num.php?explnum_id=590, consulté le:

23-04.2018 10:59:57.

[31] <https://www.commercemonde.com/2016/07/avantages-web-semantique>, consulté le: 14-06-2018 02:29.

المخلص

في هذه الرسالة قدمنا تطوير صيانة للآلة باستخدام الانطولوجيا.

للقيام بذلك قمنا بتطوير برنامج لإدارة هذه الانطولوجيا ...

يوفر تطبيقنا الميزات الأساسية: صيانة تحليل الفشل للآلات والميزات الثانوية الأخرى مثل إضافة المفاهيم والعلاقات بالإضافة إلى عرض علم الوجود.

لتطوير هذا التطبيق استخدمنا لغة جافا والمكتبة جينا.

كلمات البحث: علم الوجود ، المنطق ، الاستدلال ، الصيانة.

Abstract

In this thesis we presented the development of a maintenance system for machine using an ontology. To be able to achieve this work, we have developed a software to manage this ontology ...

Our application provides the basic features: failure, analysis, maintenance, for machines and other secondary features such as adding concepts and relationships as well as displaying the ontology.

For the development of this application we used the Java language and the Jena library.

Keywords: ontology, logic, inference, maintenance.

Résumé

Dans cette mémoire nous avons présenté le développement d'un system de maintenance pour les machine en utilisant une ontologie. pour ce faire nous avons développé un logiciel pour gérer cette ontologie...

Notre application fournit les fonctionnalités de base: défaillance, analyse, maintenance, pour des machines et autres fonctionnalités secondaires telles que l'ajout des concepts et des relations ainsi que l'affichage de l'ontologie.

Pour le développement de cette application nous avons utilisé le langage Java et la bibliothèque Jena.

Mots-clés : ontologie, logique, inférence, maintenance.