



**Université Batna 2 – Mostefa Ben Boulaïd**  
**Faculté de Technologie**  
**Département de Génie industriel**



**Thèse**

Préparée au sein du (Laboratoire d'Automatique et Productique LAP)

Présentée pour l'obtention du diplôme de :  
**Doctorat en Sciences en Génie industriel**  
**Option : Génie industriel**

Sous le Thème :

**Amélioration du processus de capitalisation et de partage  
des connaissances pour la maximisation de la valeur d'un  
système de production**

Présentée par :

**TITAH Mawloud**

**Devant le jury composé de :**

Mouss Leila Hayet	Prof.	Université de Batna 2	Président
Mouss Mohamed Djamel	Prof.	Université de Batna 2	Rapporteur
Okba Kazar	Prof.	Université de Biskra	Examineur
Bourekache Samir	MCA	Université de Biskra	Examineur
Rezeg Khaled	MCA	Université de Biskra	Examineur
Aouag Sofiane	MCA	Université de Batna 2	Examineur
Aitouche Samia	MCA	Université de Batna 2	Invité

**2020-2021**

Dans cette thèse, nous nous sommes intéressés à un modèle de gestion des connaissances des entreprises industrielles. Certaines tâches manufacturières impliquent un niveau élevé de connaissance tacite des opérateurs qualifiés. L'industrie a besoin des méthodes fiables pour la capture et l'analyse de ces connaissances tacites afin qu'elles puissent être partagées et sans aucune perte. Nous proposons, un modèle de gestion contenant deux processus de gestion, le premier processus est la capitalisation des connaissances basée sur une tâche industrielle. Nous avons utilisé une combinaison de deux méthodologies : une méthodologie d'ingénierie de connaissances CommonKADS et une méthodologie d'élicitation des connaissances MACTAK. Dans la phase de modélisation, nous avons utilisé deux différentes techniques de modélisation, une modélisation basée sur les connaissances d'expert et la deuxième une représentation ontologique. Ce modèle facilite la capture des connaissances d'experts et transforme les connaissances tacites en explicites avec une maximisation des règles de production. Le deuxième processus concerne le partage des connaissances à base d'une ontologie des Tâches Manufacturières MATO en identifiant un ensemble des concepts de fabrication et leurs relations, cette ontologie proposée facilite le partage des connaissances entre les tâches de fabrication et aide à partager et à réutiliser les connaissances durant l'exécution des tâches. Ensuite, une application proposée pour le diagnostic de système d'alarme dans une centrale thermique a été présentée pour démontrer l'importance et l'apport de l'ontologie.

**Mots-clés :** *Connaissances tacites, capitalisation des connaissances, partage des connaissances, CommonKADS, Ontologie, MATO, OWL, SWRL, requêtes SQWRL, requêtes SPARQL*

### Improvement of capitalization and sharing of knowledge to maximize the value of production system

### Abstract

In this thesis, we are interested in a knowledge management model for industrial companies. Some manufacturing tasks imply a high level of tacit knowledge which exists only in internal cognitions of qualified operators. Industry needs reliable methods for capturing and analyzing this tacit knowledge so that it can be shared without any lost. Therefore, we propose two management model processes. The first process is knowledge capitalization based on an industrial task, where we used a combination of two methodologies: a CommonKADS knowledge engineering methodology and an elicitation methodology. MACTAK knowledge is used in the modeling phase. Two different modeling techniques are used, an expert-based modeling and the second an ontological representation. This model facilitates the capture of expert knowledge and transforms tacit knowledge into explicit knowledge with the maximization of production rules. The second process is sharing knowledge based on ontology of MATO Manufacturing Tasks by identifying set of manufacturing concepts and their relationships. The proposed ontology facilitates the sharing of knowledge between manufacturing tasks and helps to share and reuse knowledge during task execution. Finally, an application of our contribution to the alarm system diagnosis in a thermal power plant was presented to demonstrate the importance of our contribution.

**Keywords:** *Tacit knowledge, knowledge capitalization, knowledge sharing, CommonKADS, Ontology, MATO, OWL, SWRL, SQWRL query, SPARQL query.*

### تحسين عملية رسملة و تبادل المعرفة لزيادة قيمة نظام الإنتاج

### ملخص

في هذه الأطروحة نهتم بنموذج إدارة المعرفة للشركات الصناعية. تتطلب بعض مهام التصنيع مستوى عالٍ من المعرفة الضمنية التي لا توجد إلا في الإدراك الداخلي للمشغلين المؤهلين. تحتاج الصناعة إلى طرق موثوقة لالتقاط هذه المعرفة الضمنية وتحليلها بحيث يمكن مشاركتها وعدم فقدها. لذلك نقترح أن يحتوي نموذج الإدارة على عمليتين إداريتين، الأولى هي رسملة المعرفة على أساس مهمة صناعية. استخدمنا مزيجاً من منهجيتين: منهجية هندسة المعرفة CommonKADS ومنهجية الاستنباط MACTAK. في مرحلة النمذجة، استخدمنا طريقتين مختلفتين، النمذجة القائمة على الخبراء والثانية التمثيل الأنطولوجي. يسهل هذا النموذج الحصول على معرفة الخبراء ويحول المعرفة الضمنية إلى معرفة واضحة مع تعظيم قواعد الإنتاج. من خلال تحديد مجموعة من مفاهيم التصنيع وعلاقاتها. تسهل هذه الأنطولوجيا المقترحة مشاركة المعرفة بين مهام التصنيع وتساعد على مشاركة المعرفة وإعادة استخدامها أثناء تنفيذ المهمة. تم تطبيق هذا النموذج في تشخيص نظام الإنذار في محطة الطاقة الحرارية لإثبات أهمية بحثنا.

**المفاتيح:** *المعرفة الضمنية، رسملة المعرفة، تبادل المعرفة، علم الوجود Ontologie، علم الوجود لمهمة MATO، لغة قواعد الويب الدلالية SWRL، طلبات SQWRL، طلبات SPARQL، لغة الويب الوجود OWL.*

## Remerciements

Je tiens à remercier vivement Dr. **Mohamed Djamel Mouss** Prof à l'université de Batna 2, pour la confiance qu'il m'a témoignée en acceptant la direction scientifique de ma thèse, Je lui suis reconnaissant de m'avoir fait bénéficier tout au long de ce travail de sa grande compétence, de son dynamisme, de sa rigueur intellectuelle, et de son efficacité certaine que je n'oublierai jamais. Soyez assuré de mon attachement et de ma profonde gratitude.

Mes vifs remerciements s'adressent à Dr. **Mouss Leila Hayet** Prof à l'université de Batna 2, pour l'honneur qu'elle me fait en acceptant de présider mon jury, et pour l'intérêt qu'elle accorde à mon travail.

Mes remerciements s'adressent également avec force aux membres du jury, Dr. **Okba Kazar** Prof à l'université de Biskra, Dr. **Bourekache Samir**, Dr. **Rezeg Khaled** tous de maitre de conférence class A à l'université de Biskra, Dr. **Aouag Sofiane** et Dr. **Aitouche Samia** tous de maitre de conférence class A à l'université de Batna 2. Leur relecture détaillée de ce manuscrit, les remarques et suggestions qu'ils ont émises ont évidemment beaucoup contribué à améliorer la qualité de ce document et constituent des recommandations pour mes travaux à venir.

Je remercie particulièrement Dr. **Kinza-Nadia Mouss** Prof à l'université de Batna 2, pour son encouragement et pour son soutien moral tout au long de ces années.

Je remercie également mes collègues, **Chahmana Safia**, **Serati Fatima Zohra**, **Guetarni Islam H. M.** de m'avoir encouragé.

Je tiens à remercier mes collègues doctorants et docteurs et surtout ceux qui m'ont toujours encouragé ainsi que les enseignants et le staff administratif du Département génie industriel de l'université Batna 2.

A mes Parents

A tous ceux qui me sont chers ...





## Abréviations

AI:	Artificial Intelligence
API:	Application Programming Interface
CommonKADS :	Common Knowledge Acquisition and Design Structuring
CML:	Conceptual Modeling Language
CRA:	Cause Root Analysis
JEE :	Java Enterprise Edition
JESS:	Java Expert System Shell
JS :	Java Script
JSP :	Java Server Pages
KA:	Knowledge Acquisition
KADS:	Knowledge Acquisition and Design Structuring
KB:	Knowledge Base
KBS:	Knowledge Based System
KCMMT:	Knowledge Capitalization Model for Manufacturing Task
KE:	Knowledge Engineering
MACTAK:	Methodology for Acquisition of Collective tacit Knowledge
MASSON:	MAnufacturing 's Semantics ONtology
MATO:	MAnufacturing Task Ontology
MRO:	Manufacturing Reference Ontology
MVC :	Modèle Vue-Contrôleur
OOPS! :	OntOlogy Pitfall Scanner!
OWL :	Ontology Web Language
PPASD :	Power Plant Alarm System Diagnosis
RDF :	Resource Description Framework
RDFS:	Resource Description Framework Schema
SWRL :	Semantic Web Rule Language
SQWRL:	Semantic Query-Enhanced Web Rule Language
SPARQL:	Simple Protocol And RDF Query Language
SRH:	Système de Refroidissement par Hydrogène
URL :	Uniform Resource Locator
UML:	Unified Modeling Language
W3C:	World Wide Web Consortium

## Liste Des Figures

Figure 1.1	Modèle hiérarchique de la connaissance et outils associés.....	6
Figure 1.2	Cycle de management des connaissances.....	8
Figure 1.3	Le modèle de création de Connaissance.....	10
Figure 1.4	La méthodologie MACTAK.....	17
Figure 1.5	Cycle de capitalisation des connaissances.....	18
Figure 1.6	Méthodologie de KADS.....	22
Figure 1.7	Modèles de CommonKADS.....	22
Figure 1.8	Classification des modèles de connaissance et techniques manipulation.....	24
		35
Figure 2.1	La nature multidimensionnelle d'une ontologie.....	
Figure 2.2	Classification d'ontologie.....	40
Figure 2.3	Les concepts principaux de MASON.....	52
Figure 2.4	Les concepts principaux de modèle MTM.....	56
Figure 2.5	Les concepts principaux de modèle MOP.....	61
		66
Figure.3.1	Le modèle de la connaissance proposée sur une tâche industrielle.....	
Figure.3.2	Le modèle d'externalisation des connaissances CommonKADS-MACTAK.....	67
Figure.3.3	L'identification de la tâche.....	67
Figure.3.4	Combinaison CommonKADS-MACTAK.....	68
Figure.3.5	La phase d'externalisation des connaissances du modèle KCMMT.....	69
Figure 3.6	Le modèle de l'organisation.....	70
Figure.3.7	La phase de modélisation des connaissances du modèle KCMMT.....	75
Figure 3.8	Les catégories de connaissance dans un modèle de connaissances.....	75
Figure 3.9	La connaissance de la tâche.....	78
Figure 3.10	La connaissance d'inférence.....	78
Figure 3.11	La méthodologie de développement de l'ontologie.....	81
Figure 3.12	Diagram de class en UML2 de domaine manufacturier.....	84
Figure 3.13	Les concepts principaux de l'ontologie MATO.....	86
Figure 3.14	Les concepts principales de l'ontologie MATO par Ontograph.....	88
Figure 3.15	Structure des classes principales et l'hierarchie des classes.....	89
Figure 3.16	Description de concept « ManufacturingTask ».....	92
Figure 3.17	Description de concept « ManufacturingEquipment ».....	93
Figure 3.18	Description de concept « ManufacturingAgent ».....	94
Figure 3.19	Description de concept « ManufacturingProblem/Solution ».....	95
Figure 3.20	Description de concept « ManufacturingFeature».....	96
Figure 3.21	Les classes principale de l'ontologie MATO et leurs relations.....	98
Figure 3.22	Un flux de travail du MATO.....	105
		109
Figure.4.1	Schéma simplifié de processus de production de la centrale de Jijel.....	
Figure 4.2	Diagramme d'activité du service quart de production.....	110
Figure 4.3	Modèle de tâche.....	112
Figure 4.4	Diagramme d'activité du service quart de production.....	114
Figure.4.5	Diagramme de Pareto nombre d'alarme des équipements.....	114
Figure 4.6	Réseau Sémantique pour les causes possibles de l'alarme.....	117
Figure.4.7	Règles de production pour le système de refroidissement turbo-alternateur.....	119

Figure.4.8 Les concepts principaux des classes.....	120
Figure.4.9 Les instances de la class « MainEquipment ».....	121
Figure.4.10 Les relations entre les classes. ....	121
Figure.4.11 Description de la connaissance et contrôle de la tâche .....	122
Figure.4.12 La procédure d'inférence .....	123
Figure 4.13 L'application de l'ontologie MATO avec la méthode de résolution des problèmes CRA.....	124
Figure 4.14 Les deux parties constitutives de l'alternateur , le rotor et le stator, ainsi que les organes qu'elles englobent.....	126
Figure 4.15 Exemple des équipements principales et secondaires de la centrale (MainEquipment) en OWL..	129
Figure 4.16 Les relations entre “manufacturingtask”, “manufacturingProblem” et “manufacturingAgent” en OWL.....	130
Figure 4.17 Exemple des propriétés des données pour le système de refroidissement en OWL.....	130
Figure 4.18 Exemple de l'individu « AI_SRH_1 » de type Alarme en OWL.....	131
Figure 4.19 Structure de l'ontologie MATO avec la méthode PSM .....	137
Figure 4.20 Implémentation de la connaissance de domaine dans Protégé.....	138
Figure 4.21 Implémentation de la connaissance d'inférence dans Protégé.....	139
Figure 4.21 Implémentation de la connaissance d'inférence dans Protégé.....	144
Figure 4.22 Sélection les causes possibles de problème lié à l'alarme AI_SRH_1 par la requête SQWRL...	142
Figure 4.23 Sélection la procédure de la solution par la requête SQWRL.....	143
Figure 4.24 Les règles générique SWRL de MATO en langage RDF.....	148
Figure 4.25 L'architecture globale du système et les composants de base.....	150
Figure 4.26 Le paradigme MVC pour notre système basé sur JEE.....	151
Figure 4.27 L'interface graphique du système PPASD en HTML +CSS+JS.....	152
Figure 4.28 Interface des paramètres d'entrées de système PPASD en JSP.....	152
Figure 4.29 Exemple d'une étape 1.1 de système PPASD.....	153
Figure 4.30 Les résultats de vérification de MATO par Pelett.....	154
Figure 4.31 Les résultats de vérification on line de MATO par OOPS !.....	155
Figure 4.32 Les indicateurs d'évaluation de MATO par OntoMetric.....	158
Figure 4.33 La mise en ligne de l'ontologie MATO sur le site on line WEBPROTÉG.....	160

## Liste Des Tableaux

Tableau 1.1	Le processus d'interview.....	14
Tableau 1.2	Les modèles de connaissances récents dans l'industrie manufacturière .....	28
		37
Tableau 2.1	Les types d'ontologies .....	
Tableau 2.2	Classification des ontologies .....	39
Tableau 2.3	Les différents raisonneurs d'une ontologie .....	46
Tableau 2.4	Les différents langages des développements d'une ontologie.....	47
Tableau 2.5	Les différents outils de développement d'une ontologie .....	49
		77
Tableau 3.1	Représentation ontologiques des connaissances de domaine .....	
Tableau 3.2	Les relation entre les concepts de l'ontologie MATO.....	87
Tableau 3.3	Exemple de formalisation des connaissances MATO en OWL .....	97
Tableau 3.4	Les règles générique SWRL de MATO.....	101
Tableau 3.5	Les règles générique SQWRL pour MATO.....	103
Tableau 3.6	Les composants inclus Processus de partage des connaissances par l'ontologie MATO.....	106
		110
Tableau 4.1	Modèle d'organisation de CommonKADS .....	
Tableau 4.2	Modèle de tâche de CommonKADS.....	111
Tableau 4.3	Modèle d'Agent de CommonKADS.....	113
Tableau 4.4	Temps pour l'acquisition des connaissances lors des entretiens avec les experts.....	115
Tableau 4.5	Construction la connaissance du domaine .....	122
Tableau 4.6	Les conditions normales et transitoires dans le système de refroidissement par Hydrogène....	127
Tableau 4.7	Les alarmes principales de système de refroidissement par Hydrogène.....	128
Tableau 4.8	Les règles SWRL spécifique pour diagnostiquer l'alarme Al_SRH_1.....	136
Tableau 4.9	Les règles SQWRL spécifique pour diagnostiquer l'alarme Al_SRH_1.....	141
Tableau 4.10	Les règles générique SWRL de MATO en langage RDF.....	144
Tableau 4.11	Conception des requêtes SPARQL à partir des requêtes SQWRL.....	145
Tableau 4.12	Les technologies et les outils utilisés dans le système.....	150
Tableau 4.13	Les pièges déclarés dans MATO.....	156
Tableau 4.14	Les résultats d'évaluation de MATO par OntoMetric.....	159

## Table des matières

Liste des abréviations	
Liste des figures	
Liste des tableaux	

### Introduction Générale

1. Problématique.....	1
2. Cadre et objectifs de travail.....	2
3. Contribution et démarche de travail.....	2
4. Organisation du mémoire.....	3

### *Première partie : Le Processus de capitalisation et de partage des connaissances : état de l'art*

#### Chapitre 1 : Capitalisation des connaissances dans les systèmes industriels

1.1	Introduction.....	05
1.2	Gestion des connaissances dans l'entreprise.....	05
	1.2.1 Donné, Information et Connaissance.....	05
	1.2.2 Connaissance tacite et explicite.....	06
	1.2.3 Management des connaissances.....	07
1.3	Modèle de création de la connaissance.....	09
	1.3.1 Externalisation.....	09
	1.3.2 Socialisation.....	09
	1.3.3 Internalisation.....	10
	1.3.4 Combinaison.....	10
1.4 T	Techniques d'acquisition des Connaissances.....	11
	1.4.1 Interviews.....	13
	1.4.2 Classification des concepts (card sorting).....	14
	1.4.3 Techniques d'observation (observation technique).....	15
	1.4.4 Information limitée (limited information).....	15
	1.4.5 Analyse du protocole (Protocol analysis).....	16
	1.4.6 Méthodologie MACTAK.....	16
1.5	Processus de capitalisation des connaissances.....	17
1.6	Système à Base de Connaissance.....	19
1.7	Ingénierie des connaissances.....	20
	1.7.1 Méthode KADS.....	21
	1.7.2 Méthodologie CommonKADS.....	21
1.8	Techniques et outils de modélisation d'une Base de connaissances.....	24
	1.8.1 Systèmes experts.....	25
	1.8.2 Systèmes linguistiques.....	26
	1.8.3 Ontologies.....	27
	1.8.4 Systèmes cognitifs.....	27
1.9	Capitalisation des connaissances dans les systèmes industriels.....	27
1.10	Conclusion.....	32

## **Chapitre 2 : Processus du partage des connaissances manufacturiers : Approche ontologique**

2.1	Introduction.....	33
2.2	Partage des connaissances dans l'entreprise.....	33
2.3	Partage des connaissances par les ontologies.....	34
2.3.1	Notion d'ontologie.....	34
2.3.2	Composants d'ontologie.....	36
2.3.3.	Classification d'ontologies.....	36
2.3.3.1	Classification selon la structure de conceptualisation.....	37
2.3.3.2	Classification selon l'objet de conceptualisation.....	39
2.3.4	Ingénierie des ontologies.....	41
2.3.4.1	Méthodologie d'Uschold et King.....	41
2.3.4.2	Méthodologie de Gruninger et Fox.....	42
2.3.4.3	METHONTOLOGY.....	43
2.3.4.4	Méthodologie Noy et McGuinness.....	43
2.3.5	Ontologies et raisonnement.....	45
2.3.6	Formalisme d'ontologies.....	45
3.3.6.1	Langage RDF.....	46
3.3.6.2	Langage OWL.....	48
3.3.6.3	Langage SWRL.....	48
2.3.7	Outils d'ontologie.....	48
2.4	Interrogation de l'ontologie.....	50
2.4.1	Requêtes SQWRL.....	50
2.4.2	Requêtes SPARQL.....	50
2.5	Ontologies aux systèmes industriels.....	51
2.6	Conclusion.....	62

### *Deuxième Partie : Proposition et mise en œuvre d'un modèle de gestion des connaissances au système industriel*

## **Chapitre 3. Processus de capitalisation et de partage des connaissances proposé**

3.1	Introduction.....	64
3.2	Processus de capitalisation des connaissances : Externalisation et Modélisation.....	65
3.2.1	Positionnement vis-à-vis de l'état de l'art.....	65
3.2.2	Conception et architecture de KCMMT.....	65
3.2.2.1	Phase d'externalisation des connaissances.....	65
3.2.2.2	Phase de modélisation des connaissances.....	74
3.2.3	Synthèse de modèle proposé.....	77
3.3	Processus de partage des connaissances basé sur l'Ontologie des Taches Manufacturières.....	79
3.3.1	Exigences d'une ontologie de tâche manufacturière.....	79
3.3.2	Méthodologie de développement de l'ontologie MATO.....	80
3.3.3	Définition des concepts principaux de l'ontologie MATO et leurs relations.....	91
3.3.4	Formalisation de l'ontologie MATO.....	97
3.3.5	Interrogation de l'ontologie de MATO par les requêtes SQWRL.....	102
3.3.6	Synthèse de modèle proposé.....	104
3.4	Conclusion.....	104

## **Chapitre 4 : Application du processus de capitalisation et de partage de connaissances au niveau de la centrale thermique**

4.1	Introduction.....	107	
4.2	Description du domaine d'application et de la tâche industrielle.....	108	
4.3	Application du KCMMT dans une tâche industrielle.....	109	
	4.3.1 Phase d'externalisation des connaissances.....	109	
	4.3.2 Phase de modélisation.....	119	
4.4	Application de l'ontologie MATO dans une tâche industrielle.....	123	
	4.4.1 Exemple d'application : Diagnostic du système de refroidissement par hydrogène du turboalternateur.....	124	
	4.4.2 Phase de la modélisation du système.....	128	
	4.4.3 Phase du raisonnement du système.....	131	
	4.4.4 Phase d'implémentation.....	137	
	4.4.5 Sélection des connaissances de l'ontologie MATO par les requêtes SQWRL pour Diagnostic du système d'alarme.....	140	
4.5	Transfert des règles SWRL génériques en OWL vers langage RDF.....	143	
4.6	Conception des requêtes SPARQL à partir des requêtes SQWRL.....	144	
4.7	Implémentation de l'ontologie MATO pour Diagnostic du système d'alarme.....	148	
4.8	Architecture de l'application développée.....	149	
4.9	Evaluation de l'ontologie MATO .....	153	
	4.9.1 Evaluation par la validation automatique.....	153	
	4.9.2 Validation basée sur les métriques.....	157	
	4.9.3 Validation basée sur les connaissances d'expert.....	159	
4.10	Conclusion.....	161	
<b>Conclusion générale et perspectives</b>			
	1. Bilan.....	163	
	2. Perspectives.....	165	
<b>Bibliographie</b> .....			169
Annexe A : CommonKADS.....			181
Annexe B : MATO.....			188
Annexe C : Requêtes SQWRL de l'ontologie MATO dans Protégé.....			196
Annexe D : Requêtes SPARQL.....			197
Annexe E : PPSAD.....			199

# Introduction Générale

1. Problématique.....	1
2. Cadre et objectifs de travail.....	2
3. Contribution et démarche de travail.....	2
4. Organisation du mémoire.....	3



## **1. Problématique**

Aujourd'hui, l'environnement économique social a besoin d'une nouvelle forme d'organisation industrielle à cause du développement rapide de la technologie de l'information et de la communication. L'expertise forme un capital important dans une organisation. D'où l'objectif des cadres de la production d'aujourd'hui vise à maximiser la performance du système industriel, et en particulier en diminuant la perte du capital intellectuel de l'entreprise par une capitalisation des savoirs et des savoirs faire des experts.

L'acquisition des connaissances est un sous domaine de l'intelligence artificielle et l'ingénierie des connaissances (Campuzano et al., 2014 ; George et Karapistolis, 2014 ; Hunter Alarcon et al ., 2010). Le processus d'acquisition de connaissances (KA) (Schreiber., 2013 ; Compton, 2013 ; Breuker, 2013 ; Musen, 2013 et Wielinga, 2013 ; Gaines, 2013) ne consiste pas à "extraire de la tête de l'expert" et à rédiger des règles pour la construction d'un système à base de connaissances (KBS), comme c'était le cas il y a 20 ans. Aujourd'hui, le processus KA a définitivement changé. L'acquisition de connaissances est considérée comme un processus cognitif qui implique à la fois des activités de modélisation dynamique et de génération de connaissances. Le KA doit être considérée comme une spirale qui croît vers le haut en transformant la connaissance tacite en connaissance explicite.

Ainsi les problématiques suivantes se posent : 1) la méthode d'extraction des connaissances tacites dans les systèmes manufacturiers est difficile (phase d'externalisation des connaissances d'experts) 2) la source la plus difficile d'acquisition est la formalisation des connaissances (phase de modélisation) 3) certaines tâches manufacturières impliquent un niveau élevé de connaissances tacites qui n'existe que dans les cognitions internes des opérateurs qualifiés. L'industrie a besoin des méthodes fiables pour la capture et l'analyse de ces connaissances tacites afin qu'elles puissent être partagées et non perdues, mais aussi pour être mieux utilisées au nouveau processus de partage des connaissances avec une nouvelle technologie.

## **2. Cadre et objectif de travail**

Cette thèse s'inscrit dans le cadre de la conduite des systèmes de production, en utilisant le modèle de création de connaissances, l'ingénierie des connaissances et le web sémantique. Au cours de la préparation de cette étude, nous avons constaté qu'il n'existe pas dans la littérature de méthodologie pour l'externalisation et l'acquisition des connaissances tacites dans l'industrie, mais plutôt l'utilisation de techniques d'élicitation des connaissances d'isolément sans séquence logique et objective orientée vers l'analyse et la résolution de problèmes industriels. Dans ce contexte, l'objectif de cette thèse est double :

- Premièrement, la capitalisation des connaissances tacites en connaissances explicites et leur modélisation d'une façon cognitive (une représentation qui converge aux raisonnements de l'être humain).
- Deuxièmement, le développement d'un processus de partage des connaissances pour améliorer la résolution des problèmes manufacturiers. Plus précisément, nous avons besoin d'un modèle de partage des connaissances en utilisant l'ontologie manufacturière.

## **3. Contribution et démarche de travail**

Deux contributions ont été identifiées dans cette thèse :

- Proposition d'un modèle de capitalisation des connaissances basé sur le processus d'externalisation et de modélisation des connaissances dans l'objectif d'améliorer la performance du système de production et capitaliser le savoir-faire de l'entreprise. Dans la phase d'externalisation, une transformation des connaissances tacites en connaissances explicites sous forme de règles de production. Dans la phase de modélisation, utilisant deux différentes techniques de modélisation, une modélisation basée sur le raisonnement d'expert et la deuxième une représentation ontologique.
- Développement d'un processus de partage des connaissances basé sur l'Ontologie des Tâches Manufacturières que nous appelons MATO (MAnufacturing Task Ontology), en identifiant un ensemble de concepts de fabrication et leurs relations. Cette ontologie proposée facilite le partage des connaissances entre les tâches de fabrication et aide à capturer et à réutiliser les connaissances durant l'exécution des tâches.

#### **4. Organisation de la thèse**

Cette contribution est mise en place et détaillée dans les différents chapitres de cette thèse organisée comme suit :

Le premier chapitre introduit les notions de bases sur la connaissance tacite et explicite, la gestion des connaissances comme concept général et sa relation avec le système de production. Nous expliquons aussi le modèle de création des connaissances. Ainsi, nous définissons les techniques d'acquisition des connaissances, puis l'identification du processus de capitalisation des connaissances, afin de mieux expliquer les méthodes de l'ingénierie des connaissances qui existent dans la littérature pour la conception d'un système à base de connaissances. De plus, nous présentons un état de l'art sur les différentes catégories de modélisation des connaissances et ses techniques de manipulation. Enfin, nous abordons un ensemble de modèles de connaissances appliqués dans les systèmes de production.

Le deuxième chapitre présente un état de l'art sur le partage des connaissances manufacturières, le partage des connaissances dans l'entreprise, et celui des connaissances par les ontologies. Nous discutons dans ce cadre les modèles de partage des connaissances appliqués dans les systèmes industriels.

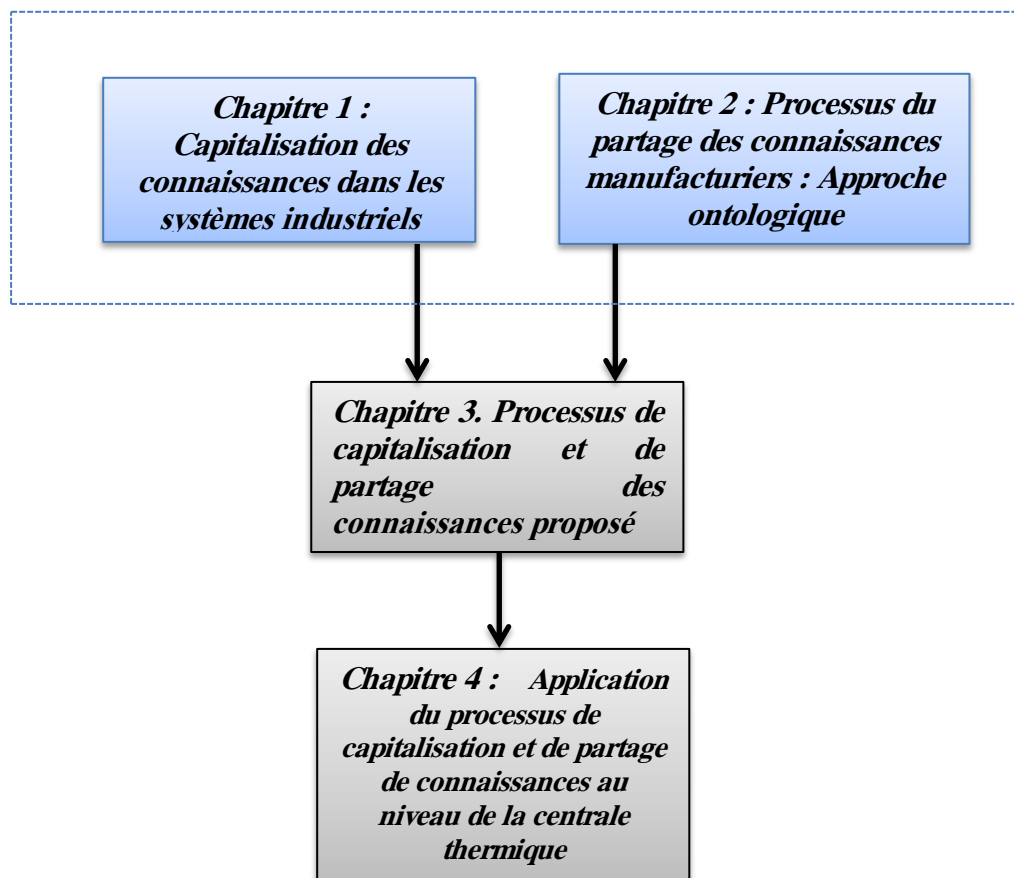
Le troisième chapitre est dédié à proposition de modèle de gestion des connaissances. Ce modèle contient deux processus de gestion. Nous discutons d'abord notre positionnement vis-à-vis de l'état de l'art. Ensuite, nous présentons la conception et l'architecture du modèle proposé. Nous présentons les propositions d'un processus à savoir de capitalisation des connaissances, à savoir et les deux phases importantes, l'externalisation et la modélisation des connaissances. Ensuite, nous proposons un processus de partage des connaissances basé sur MATO. Nous présentons tous les concepts principaux de MATO, les relations entre les concepts et la formalisation et enfin l'interrogation de l'ontologie de MATO par les requêtes SQWRL.

Le quatrième se focalise sur une application de notre contribution proposée dans le chapitre 3 dans une centrale thermique. Cette application est composée de deux modèles, le premier est la capitalisation des connaissances KCMMT et le deuxième est le partage des connaissances par l'ontologie MATO.

L'application est sur la tâche de diagnostic de système d'alarme, ce système affiche des messages d'alarmes par des panneaux de signalisation (système de surveillance en ligne). Le premier modèle de cette application est de sauvegarder les connaissances appliquées dans l'industrie manufacturière, en utilisant une nouvelle technique et méthodologie d'externalisation des connaissances. Dans la phase d'externalisation des connaissances : nous étudions les cinq étapes : l'identification des problèmes par la méthodologie CommonKADS, l'analyse des problèmes par diagramme de Pareto, diagnostiquer les problèmes analysés, nous appliquons les techniques de MACTAK, nous présentons un modèle explicite sous forme de règles de production. La deuxième phase est la phase de modélisation, nous avons utilisé deux types de modélisation, une modélisation à base d'une représentation ontologique pour la connaissance de la tâche et la connaissance du domaine et l'autre à base des connaissances d'experts pour la connaissance d'inférence. Dans le deuxième modèle, nous appliquons l'ontologie des tâches de fabrication MATO pour le diagnostic des systèmes d'alarmes de système de production, le système utilise tous les concepts importants du domaine de fabrication. Nous présentons un exemple d'application détaillé pour le diagnostic d'alarme de système de refroidissement par hydrogène de turbo alternateur. Nous étudions la phase de modélisation utilisant le langage OWL, la phase de raisonnement par les règles SWRL et la phase d'implémentation des connaissances de domaine et d'inférence par l'éditeur Protégé. Pour mettre en place l'interrogation de l'ontologie, nous présentons les requêtes SQWRL pour la sélection des connaissances de l'ontologie MATO pour le diagnostic du système d'alarme. Une application web basée sur Java JEE. Cette application web que nous appelons PPASD (Power Plant Alarm System Diagnosis). La requête est présentée par le langage SPARQL qui joue le rôle d'un pont entre les technologies du Web sémantique (dont RDF), et les plateformes Web utilisant le moteur Jena. D'autre part, une évaluation de la qualité de l'ontologie MATO est effectuée suivant les différentes approches de validation, l'évaluation par la validation automatique, la validation basée sur les métriques et la validation basée sur les connaissances d'experts.

*Première partie :*

*Le Processus de capitalisation et de partage des connaissances : état de l'art*



# *Chapitre 1 : Capitalisation des connaissances dans les systèmes industriels*

## Sommaire du chapitre 1

1.1	Introduction.....	05
1.2	Gestion des connaissances dans l'entreprise.....	05
	1.2.1 Donné, Information et Connaissance.....	05
	1.2.2 Connaissance tacite et explicite.....	06
	1.2.3 Management des connaissances.....	07
1.3	Modèle de création de la connaissance.....	09
	1.3.1 Externalisation.....	09
	1.3.2 Socialisation.....	09
	1.3.3 Internalisation.....	10
	1.3.4 Combinaison.....	10
1.4	Techniques d'acquisition des Connaissances.....	11
	1.4.1 Interviews.....	13
	1.4.2 Classification des concepts (card sorting).....	14
	1.4.3 Techniques d'observation (observation technique).....	15
	1.4.4 Information limitée (limited information).....	15
	1.4.5 Analyse du protocole (Protocol analysis).....	16
	1.4.6 Méthodologie MACTAK.....	16
1.5	Processus de capitalisation des connaissances.....	17
1.6	Système à Base de Connaissance.....	19
1.7	Ingénierie des connaissances.....	20
	1.7.1 Méthode KADS.....	21
	1.7.2 Méthodologie CommonKADS.....	21
1.8	Techniques et outils de modélisation d'une Base de connaissances.....	24
	1.8.1 Systèmes experts.....	25
	1.8.2 Systèmes linguistiques.....	26
	1.8.3 Ontologies.....	27
	1.8.4 Systèmes cognitifs.....	27
1.9	Capitalisation des connaissances dans les systèmes industriels.....	27
1.10	Conclusion.....	32

## 1.1 Introduction

Capitaliser les connaissances devient un objectif stratégique pour de nombreuses entreprises. Il ne s'agit pas d'un simple projet technique, mais surtout d'une fonction de gestion importante. Cependant, avant de développer des solutions, les entreprises doivent déterminer les connaissances sur lesquelles leurs efforts de «capitalisation des connaissances» doivent être concentrés.

Ce premier chapitre présente un état de l'art sur un certain nombre de recherches récentes sur la capitalisation et le partage des connaissances. La section 2 présente un ensemble de notions de base sur la connaissance tacite et explicite, la gestion des connaissances comme concept général et sa relation avec le système de production. La section 3 explique le modèle de création des connaissances. La section 4 définit les techniques d'acquisition des connaissances. Cela conduit alors à une explication d'efforts encouragés par les chercheurs et cadres d'entreprise, puisque ce travail de recherche met également l'accent sur l'amélioration de ces processus pour la réutilisation industrielle. La section 5 est dédiée à l'identification du processus de capitalisation des connaissances. La section 6 traite le Système à Base de Connaissance. La section 7 explique les méthodes de l'ingénierie des connaissances qui existent dans la littérature pour la conception d'un système à base de connaissances. La section 8 est un état de l'art sur les différentes catégories de modélisation des connaissances et ses techniques de manipulation. Enfin, on aborde un ensemble de modèles de connaissances appliqués dans les systèmes de production (section 1.9).

## 1.2 La gestion des connaissances dans l'entreprise

### 1.2.1 Donnée, information et connaissance

Les données sont des faits objectifs qui relatent, une valeur, souvent une mesure ou une description sans contexte. Comme des observations simples, les données n'ont aucun sens et aucune signification au-delà de leur existence (Jifa, 2013 ; Bellinger et al., 2004).

L'information est une collecte de données bien organisées et structurées qui donnent un sens par des moyens de connexion relationnelle. Les informations sont des données contextuelles, pertinentes (Cooper, 2014 ; Jifa, 2013)

La connaissance quant à elle, tire son origine du cerveau des individus, et se construit à partir de l'information qui est transformée et enrichie par l'expérience personnelle, la connaissance est du savoir, du savoir-faire de l'expérience (Bellinger et al., 2004).

On peut distinguer entre donnée, information et connaissance. La figure 1.1 représente un modèle hiérarchique de la connaissance qui nous permet de faire une distinction entre donnée, information et connaissance.

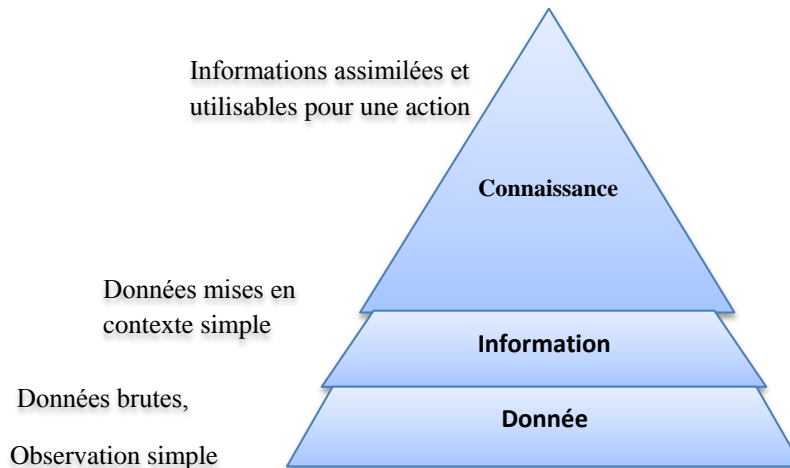


Figure 1.1. Modèle hiérarchique de la connaissance et outils associés

### 1.2.2 Connaissance tacite et explicite

Dans le domaine de la gestion des connaissances, une grande partie des connaissances n'est pas explicite, on dit que ces connaissances sont tacites. Selon (Nonaka., 1991 ), il y a deux catégories générales de la connaissance qui sont les connaissances tacites et explicites.

La connaissance explicite est une connaissance transmissible (formalisable) dans un langage formel et systématique. Pour cette raison, elle peut être facilement communiquée et partagée, dans des spécifications de produit ou une formule scientifique ou un programme informatique (Nonaka., 1991 ; Nonaka et Takeuchi., 2007).

La connaissance tacite (implicite, non formalisable) est un aspect personnel qui la rend difficile à formaliser et à communiquer (Nonaka et Takeuchi., 2007). La connaissance tacite se compose en partie de compétences techniques, le type de compétences informelles, difficiles à cerner, saisies dans le terme «savoir-faire». Un



maître-artisan, après des années d'expérience, développe une richesse d'expertise «au bout de ses doigts» (Nonaka et Takeuchi., 2007).

### 1.2.3 Gestion des connaissances

De nombreuses recherches ont discuté de la question de la gestion des connaissances ou « management des connaissances ». Plusieurs définitions ont été données. La connaissance se définit dans la littérature et dans certains ouvrages de référence comme :

Selon le dictionnaire [le ROBERT 1994](#) « ce qui est connu, est présent à l'esprit ; ce que l'on sait pour l'avoir appris».

Selon Nonaka et Takeuchi ([Nonaka et H.Takeuchi 1995](#)), « la connaissance est du savoir, du savoir-faire, de l'expérience, voire du savoir-être. La connaissance peut être tacite ou explicite». Selon l'AFNOR 2002, elle est «un ensemble de représentations, idées ou perceptions acquises par l'étude ou par l'expérience».

La gestion des connaissances (GC) est donc une stratégie consciente visant à transmettre les bonnes connaissances aux bonnes personnes au bon moment et à aider les gens à partager et à mettre en pratique les informations de manière à améliorer les performances organisationnelles ([O'Dell et Grayson, 1998](#)). (USA, Management).

Azmee ([Azmee et al., 2017](#)) définit la GC comme étant la bonne connaissance donnée aux bonnes personnes, au bon moment, pour leur permettre de prendre une excellente décision.

La gestion des connaissances est passée d'un concept prématuré à une nécessité organisationnelle générale. Au fil du temps, la nature exacte du terme a évolué. Au cours de la dernière décennie, la responsabilité d'établir un sens appliqué précis du domaine est passée des universitaires aux praticiens. Ces derniers ont massé les premières définitions académiques pour répondre aux désirs et aux besoins de leurs électeurs particuliers. Ce projet est une première tentative d'enregistrer les définitions appliquées qui ont progressé pour s'assurer qu'elles sont disponibles pour les universitaires et les praticiens ([Girard, 2015](#)).

Selon Davenport et Prusak ([Davenport et Prusak 1998, p. 163](#)), la gestion des connaissances s'appuie sur les ressources existantes qu'une organisation peut déjà

avoir en place : une bonne gestion des systèmes d'information, une gestion du changement organisationnel et des pratiques de gestion des ressources humaines.

L'une des définitions les plus connues est celle de Petrash (Petrash, 1996), considérant la gestion des connaissances comme fournissant les bonnes connaissances aux bonnes personnes au bon moment, afin de les réutiliser et de profiter des connaissances existantes. Dans cette définition de la gestion des connaissances, l'auteur souligne l'importance de fournir, de tirer profit et de réutiliser les connaissances. Ces énoncés sont considérés cibler deux étapes de la gestion des connaissances: l'identification (fournir) et la réutilisation (profit, réutilisation). La gestion des connaissances est une gestion explicite et systématique des processus permettant d'identifier, de créer, de stocker, de partager et d'utiliser des ressources vitales de connaissances individuelles et collectives. Son expression pratique est la fusion de la gestion de l'information et de l'apprentissage organisationnel (Serrat, 2009).

La gestion des connaissances est une discipline qui favorise une approche intégrée de la création, de la capture, de l'organisation, de l'accès et de l'utilisation des actifs informationnels d'une organisation. Ces actifs comprennent des bases de données structurées, des informations textuelles telles que des documents de politique et de procédure et, surtout, les connaissances et l'expertise tacites résidant dans les chefs des employés individuels («Qu'est-ce que la gestion des connaissances?»).

La gestion des connaissances est le transfert des connaissances tacites en connaissances explicites et leur partage au sein de l'organisation (Uriarte, 2008, p. 13).

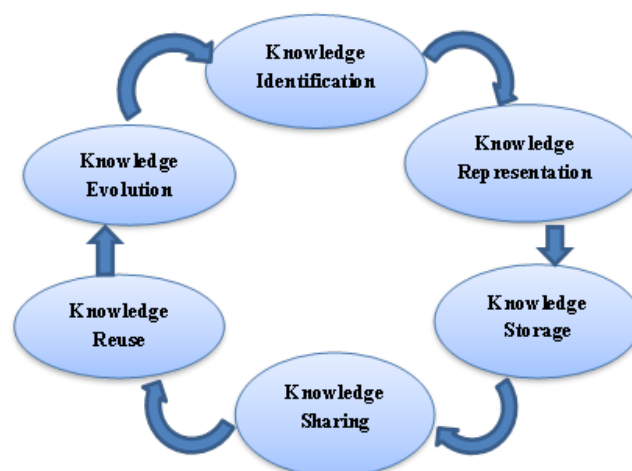


Figure 1.2 Cycle de gestion des connaissances: (Girard, 2015)

En conséquence, la figure 1.2 résume les différents composants nécessaires pour garantir le cycle de gestion des connaissances efficace:

- *L'identification des connaissances*: c'est la collecte de connaissances à capitaliser.
- *La représentation*: structure les connaissances à capitaliser.
- *Le stockage des connaissances* : consiste à sauvegarder des connaissances capitalisées dans un espace de travail.
- *Le partage des connaissances*: fournit des solutions pour diffuser des connaissances capitalisées.
- *La réutilisation des connaissances*: c'est la capacité d'appliquer des connaissances capitalisées dans un contexte donné.
- *L'évolution des connaissances*: c'est la capacité de maintenir des connaissances capitalisées.

### **1.3 Modèle de création de la connaissance**

Les connaissances peuvent être classées en deux catégories: la connaissance tacite et la connaissance explicite. En outre, la connaissance tacite est difficile à utiliser et à cacher dans les modèles de cerveau personnel (Chen, 2010). En fait, les connaissances tacites sont personnelles et difficiles à formaliser. Aussi, Nonaka et al. (Nonaka et al., 2000) ont développé le modèle en spirale de la connaissance, ce modèle contient quatre modes de conversion des connaissances (figure 1.3) appelés SECI (socialisation, externalisation, combinaison, internalisation).

#### **1.3.1 Externalisation (Tacite vers Explicite)**

L'externalisation est un processus qui permet le passage des connaissances tacites en connaissances explicites, sous forme de concepts, modèles ou hypothèses. La modélisation d'un concept est très souvent déclenchée par le dialogue et l'échange avec d'autres individus.

#### **1.3.2 Socialisation (Tacite vers Tacite)**

Dans cette conversion, un individu partage ses connaissances tacites directement avec un autre. Ils font partie de sa propre base de connaissances tacites. Autrement dit, elle est «socialisée» dans le métier (Nonaka et Takeuchi., 1995). Ce processus s'effectue généralement par l'observation, l'imitation et surtout la pratique. Dans ce cas, l'apprenti apprend les compétences du maître, mais ni l'apprenti ni le maître

n'obtiennent un aperçu systématique de leurs connaissances artisanales. Parce que leurs connaissances ne deviennent jamais explicites, elles ne peuvent pas être facilement exploitées par l'organisation dans son ensemble. La socialisation est une forme plutôt limitée de la création de connaissances.

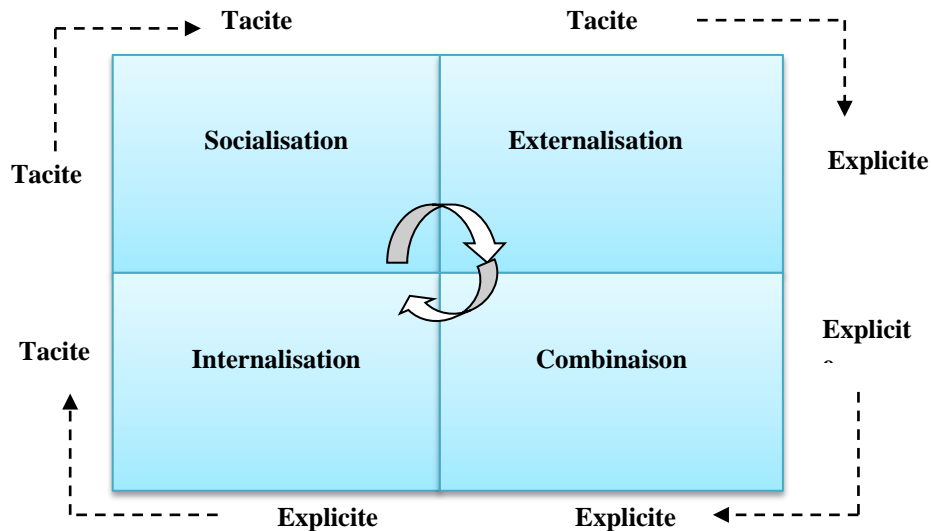


Figure 1.3. Le modèle de création de Connaissance. Source: (Nonaka et Takeuchi, 2000)

### 1.3.3 Internalisation (Explicite vers Tacite)

L'internalisation est le processus de conversion de connaissances explicites en connaissances tacites. De plus, au fur et à mesure que de nouvelles connaissances explicites sont partagées dans toute une organisation, d'autres employés commencent à les internaliser, c'est-à-dire qu'elles les utilisent pour élargir, étendre et recadrer leurs propres connaissances tacites. Les compétences requises sont progressivement traduites, par essai/erreur et interaction, en compétences tacites.

### 1.3.4 Combinaison (Explicite vers Explicite)

Dans cette conversion, un individu peut également combiner des éléments discrets de connaissances explicites dans un nouvel ensemble. Le modèle de la création de la connaissance développée par Nonaka (Nonaka et Takeuchi, 1995) considère que la première fonction de l'entreprise est de créer un avantage concurrentiel basé sur le savoir collectif pour la création de la connaissance.

Au cours des années précédentes, Senker [Senker, 1995] a ajouté à la littérature croissante concernée par les aspects tacites des activités innovantes et a suggéré une définition fonctionnelle de la connaissance tacite dans ce contexte. Ses principales préoccupations sont de préciser si l'importance du savoir tacite diffère selon le secteur

industriel ou la technologie étudiée. En outre, Szulanski (Szulanski, 2003) parle de la «vulnérabilité de la connaissance» lors de la transformation du savoir d'une pratique à l'autre. Cette méthode implique la source de la connaissance.

#### 1.4 Techniques d'acquisition des Connaissances

Le processus d'acquisition de connaissances (Knowledge Acquisition) KA ne consiste pas à externaliser la connaissance de la tête de l'expert ni à rédiger sous forme de règles de production pour la construction d'un système à base de connaissance (KBS). L'acquisition de connaissances a commencé comme une tentative de résoudre le principal goulot d'étranglement dans le développement de systèmes à base de connaissances (KBS) (Cairó & Guardati., 2012). Les ingénieurs des connaissances étaient les professionnels qualifiés du transfert d'expertise vers une base de règles.

Selon Musen (Musen, 2013), l'acquisition de connaissances n'est pas facile à obtenir des experts du domaine et des ingénieurs du savoir qui parlent de même langage.

L'acquisition de connaissances est difficile car le domaine des experts et les ingénieurs du savoir doivent créer ensemble une langue pour définir un modèle d'expertise professionnelle qui n'a jamais existé auparavant dans un sens formel. Comme Breuker, Muse et Motta (Breuker, 2013 ; Musen, 2013 ; Motta, 2013) l'ont déclaré, la résolution de problèmes n'a pas reçu l'attention voulue contrairement à son homologue la connaissance du domaine. La résolution des problèmes est un processus comprenant des étapes d'identification, d'analyse, de diagnostic et de proposition de solutions.

Avant la modélisation de la résolution des problèmes, Schreiber (Schreiber,1999) a utilisé la connaissance des tâches pour permettre le raisonnement avec la connaissance du domaine qui a été proposée et détaillée pour le développement de KBS. Cependant, la recherche est passée radicalement de la préoccupation de susciter des connaissances à celle de construire des modèles cohérents et robustes de résolution de problèmes.

Comme contribution de Gaines (Gaines, 2013), une étude de 25 ans passée sur le domaine d'acquisition des connaissances, nous rappelle que l'acquisition de

connaissances a toujours été un processus clé pour les sociétés humaines, une nouvelle vision dans ce domaine est présentée comme suit :

- D'après une analyse de la relation entre KA et le Web du (Schreiber., 2013), le Web a eu un impact important sur l'acquisition de connaissances, mais aussi l'inverse,
- D'après (Breuker, 2013 ; Musen, 2013 et Wielinga, 2013), l'acquisition des connaissances est centrée sur des approches de modélisation.
- une approche de (Compton, 2013) qui se concentre sur la question de la cognition située, par le développement de solutions opérationnelles basées sur des règles pour les systèmes d'aide à la décision.
- une analyse de (Shadbolt, 2013) centrée sur «la montée des machines sociales», où le crowdsourcing web 2.0 devient une architecture système et les technologies KA sont exploitées pour les scénarios de calcul social distribués activés par le Web.
- un essai de (Gruber., 2013) qui se concentre sur l'une des clés défis pour la recherche en intelligence artificielle (IA), les cinquante dernières années: comment construire véritablement un système non fragile, capable d'acquérir de nouvelles connaissances et de les utiliser efficacement dans des tâches concrètes scenarios.

Le processus KA a définitivement changé. Aujourd'hui, l'acquisition de connaissances est considérée comme un processus cognitif qui implique à la fois des activités de modélisation dynamique et de génération de connaissances. Le processus KA doit être considéré comme une spirale de contenu épistémologique et ontologique qui grandit en transformant la connaissance tacite en connaissance explicite (Titah et al., 2017),

Cependant, le processus d'acquisition de connaissances est similaire aux ontologies où il dépend fortement des connaissances d'experts volatiles du domaine, qui sont parfois complètes et incomplètes, précises et imprécises ou certaines et incertaines (Liu, Liu, Lin et Liu, 2013). Il est difficile de saisir entièrement les connaissances d'un expert, car la plupart de ses connaissances sont cachées dans ses compétences.

### 1.4.1 Interviews

L'interview est la première technique naturelle d'acquisition des connaissances, presque tout le monde commence à obtenir des connaissances avec un ou plusieurs entretiens. Il y a deux types d'interview : les interviews non-structurées et les interviews structurées.

#### 1.4.1.1 Interview non-structurée

Selon (Liou, 1990; Schrieber et al., 2000), l'interview non structurée n'a pas d'agenda (ou, du moins, pas d'agenda détaillé) défini par l'ingénieur du savoir ou par l'expert. Bien entendu, cela ne signifie pas que l'ingénieur du savoir n'a pas d'objectif pour l'entretien, mais cela signifie qu'il dispose d'une marge de manœuvre considérable: il y a peu de contraintes. Les avantages de cette approche proviennent de ce manque de contraintes. Premièrement, l'approche peut être utilisée chaque fois que l'un des objectifs de l'entretien est que l'expert et l'ingénieur du savoir établissent une bonne relation. Il n'y a pas d'obstacle formel à la discussion, que les deux participants jugent bon. Deuxièmement, l'ingénieur peut avoir facilement une vue d'ensemble du sujet; il peut "combler les lacunes" dans sa propre perception du domaine, se sentant ainsi plus à l'aise avec son modèle mental. Troisièmement, l'expert peut décrire le domaine de manière familière, en abordant des sujets qu'il considère importants et en ignorant ceux qui le concernent.

#### 1.4.1.2 Interview structurée

L'entretien structuré est une version formelle de l'entretien tant que l'ingénieur du savoir planifie et dirige la session. L'entretien structuré présente l'avantage de proposer des transcriptions structurées plus faciles à analyser que les discussions non structurées. L'ingénieur des connaissances planifie et dirige la session, prend la forme d'un dialogue fournisseur-avocat, fournit des données d'expertise plus ciblées, souvent utilisées pour "combler les lacunes" dans la base de connaissances, phase de raffinement des connaissances, utile également à la fin de l'identification ou du début de la spécification de la connaissance. Toujours créer une transcription.

L'acquisition de connaissances dans les systèmes flous peut provenir d'experts humains ou pilotée par les données (Zajaczowski & Verma, 2012; Zhang et Mahfouf, 2011).

**Tableau 1.1** Le processus d'interview

<p><b>A. Préparation de l'interview</b></p> <ul style="list-style-type: none"> <li>• établir le but de la session</li> <li>• Envisager une valeur ajoutée pour les experts</li> <li>• Décrivez vous-même le profil de l'expert</li> <li>• Lister les questions pertinentes</li> <li>• Noter les déclarations d'ouverture et de clôture</li> <li>• Vérifier le matériel d'enregistrement - l'enregistrement audio est généralement suffisant</li> <li>• Assurez-vous que l'expert connaît le contexte de la session: objectif, durée, suivi, etc.</li> </ul>	<p><b>B. Début d'interview</b></p> <ul style="list-style-type: none"> <li>• Présentez-vous (si nécessaire)</li> <li>• Clarifier les objectifs et les attentes</li> <li>• Indiquez comment les résultats seront utilisés</li> <li>• Problèmes de confidentialité</li> <li>• Vérifiez s'il reste des questions à l'expert</li> <li>• Créer autant que possible une confiance mutuelle</li> </ul>
<p><b>C. Durant l'interview</b></p> <ul style="list-style-type: none"> <li>• éviter les questions suggestives</li> <li>• Clarifier le motif de la question</li> <li>• questions de phrase en termes de sondes - par exemple, «pourquoi...»</li> <li>• Faites attention aux aspects non verbaux</li> <li>• Soyez conscient des préjugés personnels</li> <li>• Donner des résumés à des points intermédiaires</li> </ul>	<p><b>D. Fin de l'interview</b></p> <ul style="list-style-type: none"> <li>• Reformuler l'objectif de la session</li> <li>• Demander des informations supplémentaires / éligibles</li> <li>• Indiquez quelles seront les prochaines étapes</li> <li>• Prenez rendez-vous pour les prochaines réunions</li> <li>• Traitez les résultats des entretiens dès que possible.</li> <li>• Organiser un tour de rétroaction avec un expert</li> <li>• Distribuez les résultats de la session</li> </ul>

L'approche de l'expert humain se prête à une conception manuelle de modèles flous basés sur les connaissances existantes récupérées auprès d'un expert à travers des entretiens et des questions ouvertes.

### 1.4.2 Classification des concepts (card sorting)

Le tri de concepts est une technique utile lorsque nous souhaitons découvrir les différentes manières dont un expert perçoit les relations entre un ensemble fixe de concepts. Dans la version la plus simple, un expert se voit présenter un certain nombre de cartes sur lesquelles est imprimé un mot concept. Les cartes sont mélangées et on demande à l'expert de les trier soit en un nombre fixe de piles, soit en un nombre quelconque de piles qu'il juge approprié. Ce processus est répété plusieurs fois.

Cette tâche permet d'obtenir plusieurs vues de l'organisation structurelle de connaissances en demandant à l'expert de faire la même tâche encore et encore. Chaque fois que l'expert trie les cartes, il doit créer au moins une pile qui diffère des tris précédents. L'expert doit également fournir une étiquette de nom ou de catégorie pour chaque pile de chaque type.



Les avantages du tri par concept peuvent être caractérisés comme suit: Il est rapide à appliquer et facile à analyser et il force dans un format explicite les constructions qui sous-tendent à la compréhension de l'expert. En fait, il est souvent instructif pour l'expert. Une sorte peut mener l'expert à voir la structure dans sa vision du domaine que lui-même n'a pas consciemment articulé avant. Enfin, dans les domaines où les concepts sont de nature perceptuelle (rayons X, mises en page et images de types divers), les cartes peuvent alors être utilisées comme moyen de présenter ces images et d'essayer d'obtenir des noms pour les catégories et les relations. Cela pourrait les lier. Il faut bien sûr se méfier des caractéristiques avec ce type de technique. Les experts peuvent confondre souvent les dimensions en n'appliquant pas systématiquement les mêmes distinctions sémantiques au cours d'une séance d'élicitation. Ils peuvent aussi simplifier à l'excès la catégorisation des éléments, en omettant d'importantes mises en garde. Une astuce importante avec toutes les techniques que nous examinons est toujours d'enregistrer sur bande audio ces sessions. Un expert fait de nombreux apartés, commentaires et qualifications dans le cas du classement, etc. En fait, on peut choisir d'utiliser les méthodes artificielles pour réaliser des entretiens auxiliaires structurés. La structure cette fois est centrée sur l'activité de la technique.

#### **1.4.3 Techniques d'observation (observation technique)**

Grâce à cette méthode, l'expert vient avec la solution du problème et le chercheur observe le processus (Liou., 1990). L'expert, ici, est généralement plus à l'aise une fois qu'il peut dans son environnement de travail, effectuer ses tâches quotidiennes. L'observation peut être utilisée pour identifier des stratégies pour résoudre un problème dont le sujet n'est pas au courant. Il peut également être mis en œuvre pour étudier la motricité ou des procédures automatiques, pour identifier les tâches impliquées dans la résolution d'un problème.

#### **1.4.4 Information limitée**

Selon (Hoffman et al., 1995), l'approche d'information limitée (également connue sous le nom de focalisation contextes) est de présenter un problème fictif à l'expert, dans l'attente qu'il décrirait une méthode pour sa solution. C'est possible d'adopter une approche qui limite progressivement la disponibilité de l'information. De cette façon, l'expert révèle le niveau minimum d'informations dont il a besoin pour

prendre une décision. L'expert peut aussi être explicitement chargé d'adopter une stratégie particulière. L'expérience consiste à montrer plusieurs photographies sur des sujets avec lesquels l'expert a l'habitude de travailler, ayant une bien plus grande restriction de temps. Après cela, il lui est demandé d'externaliser tout ce qu'il peut se rappeler de ces photos et fournir son interprétation sur elles.

#### 1.4.5 Analyse du protocole

Selon (Schrieber et al., 2000), l'analyse de protocole (PA) est un terme générique désignant un certain nombre de façons différentes d'analyser le (s) expert (s) entrain de résoudre des problèmes du domaine. Dans tous les cas, l'ingénieur enregistre ce que l'expert fait - de préférence par vidéo ou bande audio - ou au moins par notes écrites. Les protocoles sont ensuite créés à partir de ces enregistrements et l'ingénieur du savoir tente d'extraire une structure et des règles significatives des protocoles.

Selon (Cooke., 1994), certaines techniques sont utilisées afin d'analyser les protocoles, de la manière la plus fiable possible au domaine, et aux processus mentaux originaux de l'expert humain, Elles sont:

- L'analyse de contenu: c'est une manière systématique et objective d'organiser une grande quantité de matériel en identifiant certains de ses caractéristiques spécifiques. Il est possible de classer des phrases ou leurs parties par type des connaissances qu'ils démontrent: concepts, règles, procédures, autres.
- L'analyse d'interaction: c'est un moyen d'analyser presque grammaticalement l'interaction entre l'intervieweur et l'expert pour identifier les catégories de phrases. Dans ce cas, les catégories créées sont orientées vers la conversation,

Par exemple: négociation, orientation, explication, autre. Il permet au chercheur d'éliminer ou de confirmer certaines catégories de réponses des experts qu'il considère comme le plus important à un moment donné au cours de la transcription.

- Les outils d'analyse de protocole: ce sont des outils qui utilisent des techniques pour analyser les protocoles automatiquement. L'objectif principal du protocole outils d'analyse est d'identifier et de créer des catégories de propositions et l'organisation sémantique des segments de texte.

#### 1.4.6 Méthodologie MACTAK

La méthodologie MACTAK (Methodology for Acquisition of Collective tacit knowledge) développée par (Claudio Roberto et al., 2015) est une méthodologie d'acquisition des connaissances tacites collectives en connaissances explicites. Elle combine les méthodes d'acquisition des connaissances avec les outils du contrôle de

qualité, cette méthodologie basée sur cinq étapes successives : l'interview non structurée, la classification des concepts, Les techniques d'observation, L'information limitée et L'analyse du protocole. Associes avec les techniques de contrôle de qualité : le brainstorming, le diagramme de Pareto et diagramme Ishikawa (voir figure 1.4).

### 1.5 Le Processus de capitalisation des connaissances

Il existe plusieurs notions de la capitalisation des connaissances dans la littérature. Les problèmes et difficultés des pratiques et solutions actuelles, liées à ses différentes phases. Les entreprises industrielles d'aujourd'hui demandent la capitalisation du savoir-faire des experts.

Capitaliser les connaissances des entreprises devient un objectif stratégique pour de nombreuses entreprises. Il ne s'agit pas d'un simple projet technique, mais surtout d'une fonction de gestion importante.

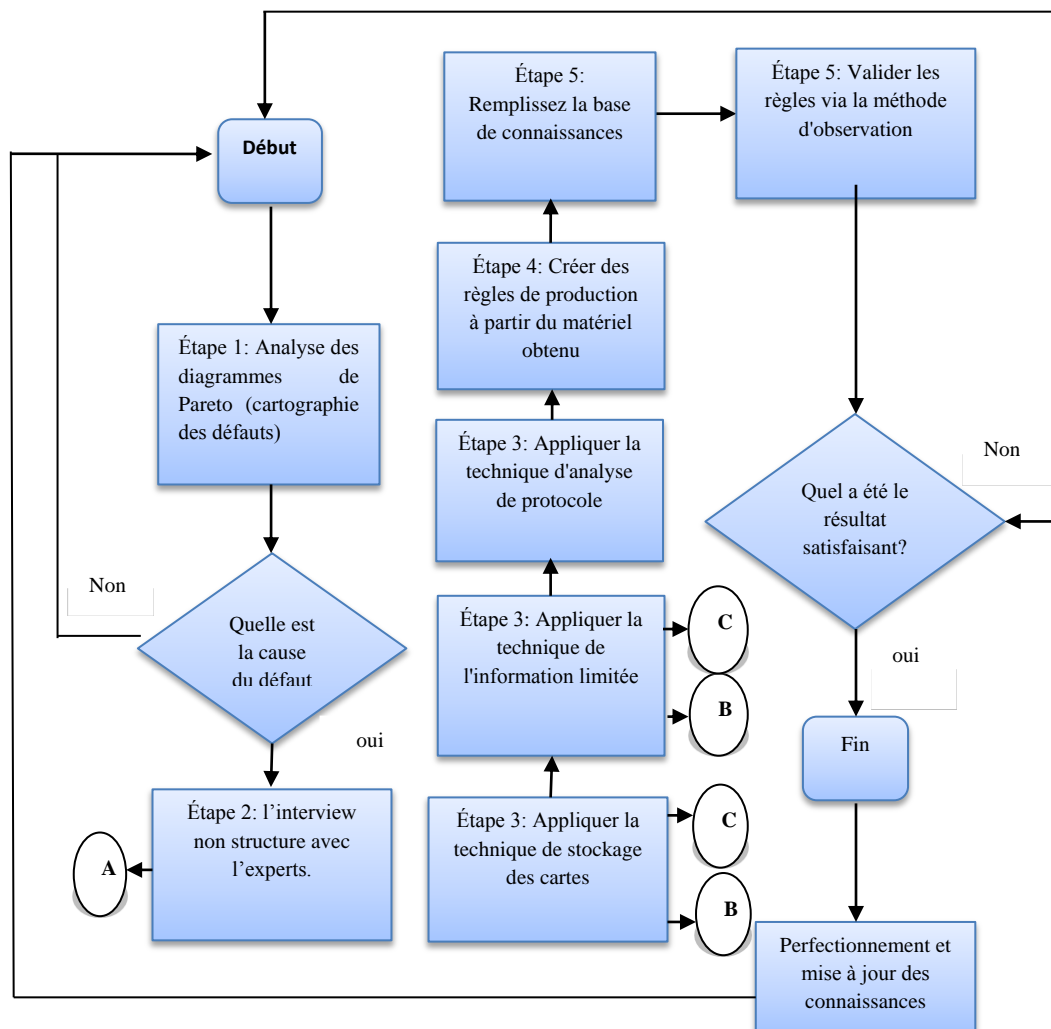


Figure 1.4 la méthodologie MACTAK. Source: (Claudio Roberto et al., 2015)

Cependant, avant de développer des solutions liées au problème de capitalisation des connaissances, les entreprises doivent déterminer les connaissances sur lesquelles leurs efforts de «capitalisation des connaissances» doivent être concentrés (Grundstein, 1996).

Selon (Matta et al., 2001) La capitalisation de la connaissance est :« une formalisation d'expérience gagnée dans un domaine spécifique ».

Autre définition de Capitalisation de (Grundstein, 1996) : « considérer certaines connaissances utilisées et produites par l'entreprise comme un ensemble de richesses et en tirer des intérêts contribuant à augmenter la valeur de ce capital ». Cette définition appelle trois aspects qui concernent respectivement: les deux principales catégories des connaissances d'entreprise, le caractère collectif et privé du savoir individuel et le problème de la capitalisation de la connaissance d'entreprise.

Un cycle de capitalisation des connaissances est proposé par (Grundstein, 2000) présenté dans la figure 1.5. Il est composé de quatre phases de repérage, préservation, valorisation et actualisation et représente également les besoins du développement d'un système à base de connaissances.

Le but est de « localiser et rendre visibles les connaissances de l'entreprise, être capable de les conserver, y accéder et les actualiser, savoir comment les diffuser et mieux les utiliser, les mettre en synergie et les valoriser » (Grundstein, 2000).

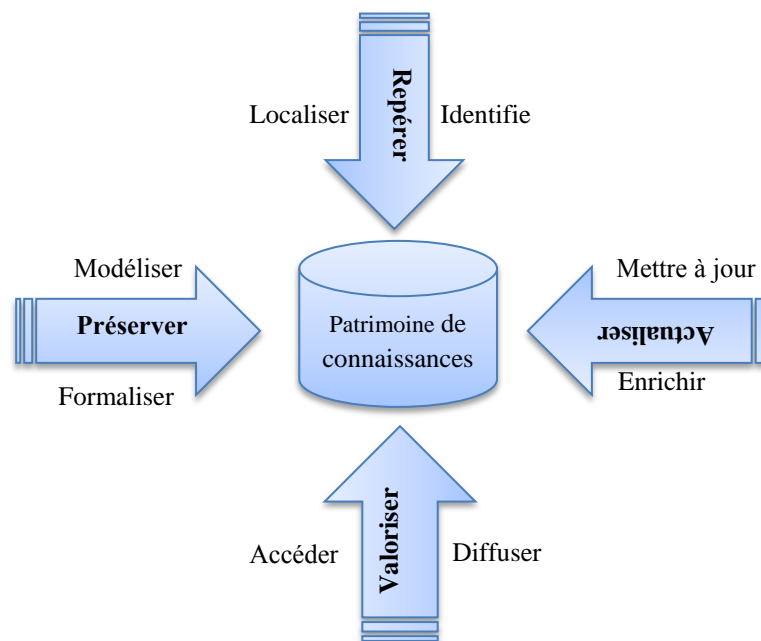


Figure 1.5 Cycle de capitalisation des connaissances (Grundstein, 1996)

Notons que, dans nos recherches, la capitalisation des connaissances implique l'externalisation et la représentation des connaissances, le stockage dans une base de connaissances. De plus, nous considérons l'évolution des connaissances comme une étape nécessaire pour une capitalisation efficace des connaissances.

- Le repérage des connaissances, consiste en la collecte des savoirs et savoir-faire, les identifier, les localiser, les caractériser, en faire des cartographies, estimer leur valeur économique et les hiérarchiser.

- La préservation des connaissances : modéliser les connaissances, les formaliser et les conserver.

- La valorisation des connaissances : rendre les connaissances accessibles selon les règles de confidentialité et de sécurité, les diffuser, les partager, les exploiter.

- L'actualisation des connaissances : évaluer les connaissances, de la création de connaissances nouvelles.

## 1.6 Système à Base de Connaissance

[Dignum et van de Riet \(1991\)](#) ont défini une base de connaissances comme "un ensemble de déclarations qui décrivent le savoir sur les vérités du monde réel plus un ensemble de contraintes qui décrivent des déclarations qui doivent être vraies dans tous les mondes et déclarations possibles qui devraient être vraies dans tous les mondes possibles " (p. 4). Tout système qui représente la connaissance est appelé système à base de connaissances ou en anglais *Knowledge-based system* (KBS).

Dans le passé, le développement de KBS a été considéré comme un transfert de connaissances humaines dans la base de connaissances mises en œuvre ([Wielinga, et al, 1992](#)). Ce point de vue était basé sur la théorie selon laquelle les connaissances requises existent déjà et doivent seulement être collectées et mises en œuvre. Habituellement, les connaissances requises sont acquises en interviewant un expert et mises en œuvre sous la forme de règles de production. Cependant, cette approche ne soutient pas une présentation adéquate des différents types de connaissances ([Studer et al., 1998](#)). L'existence de différents types de connaissances et l'absence de justifications satisfaisantes des règles rendent le processus de maintenance difficile et prend du temps.

## 1.7 Ingénierie des connaissances

L'ingénierie des connaissances ou Knowledge Engineering (KE), a été définie en 1983 par Edward Feigenbaum et Pamela McCorduck en tant que discipline d'ingénierie qui implique une connaissance intégrée des systèmes informatiques afin de résoudre des problèmes complexes nécessitant normalement un haut niveau d'expertise humaine (George et Karapistolis, 2014). Les systèmes basés sur la connaissance (KBS) ont évolué à partir de programmes informatiques qui automatisent partiellement la création de solutions spécifiques pour des problèmes de conception, anciennement connus sous le nom de systèmes experts, pour le développement de systèmes intégrés qui prennent en charge les décisions des concepteurs le long du cycle de vie du produit (Hunter Alarcon et al., 2010).

Dans le domaine de l'ingénierie du savoir (KE), un certain nombre d'approches et de méthodologies sont discutées dans la littérature de recherche, par exemple: KOD (Knowledge Oriented Design) par (Vogel, 1990), CBR (Razement Case-Based) par (Aamodt and Plaza, 1994) MASK (Méthodologie pour analyser et structurer les connaissances) par (Ermine, 1996). CIMOSA par (Kosanke et al., 1999). MOKA (Méthodologie et outils orientés vers les applications d'ingénierie basées sur le savoir) par (Stokes, 2001). CommonKADS (acquisition de connaissances communes et support de conception) par (Schreiber, 2002). La principale différence entre ces approches réside dans le type de projets de modélisation de la connaissance qu'ils soutiennent. CommonKADS est orienté vers des connaissances très spécialisées pour créer des systèmes experts en utilisant des modèles spécifiques. MOKA est plus concentré sur le développement d'un logiciel de conception assistée par ordinateur (CAO) (Campuzano et al., 2014). CBR est une méthode de résolution de problèmes et d'apprentissage provenant de la psychologie cognitive, qui imite le processus cognitif des êtres humains. Le processus CBR comprend principalement la récupération des cas, la réutilisation des cas, la révision des cas et les étapes de rétention des cas (Aamodt et Plaza, 1994). MASK permet de maîtriser la complexité dans les projets de gestion de la connaissance; Le KOD est basé sur une approche inductive pour expliciter le modèle conceptuel pour la facilité de l'analyse sémantique de la terminologie.

### 1.7.1 Méthode KADS

La méthode KADS (Knowledge Acquisition and Design Structuring) développé par (Wielinga et al., 1992) dans le cadre du projet (ESPRIT-I P1098). Ce projet développe une méthodologie complète et commercialement pour la construction d'un système basé sur la connaissance (KBS) est considéré comme une activité de modélisation.

KADS est une méthode pour le développement structuré et systématique de systèmes basés sur la connaissance, qui vise à fournir un soutien en ingénierie logicielle pour le processus d'ingénierie des connaissances - des premiers entretiens exploratoires d'experts au développement d'un système opérationnel. KADS est une méthode manuelle et non formelle. Des travaux sont en cours pour développer des établis de plus en plus sophistiqués d'outils informatiques qui aideront les ingénieurs du savoir à appliquer les principes de KADS.

- Le modèle d'interprétation: Le modèle d'interprétation fournit un cadre indépendant du domaine pour aider l'ingénieur des connaissances dans l'analyse.
- Le modèle conceptuel : Une fois qu'un modèle d'interprétation a été sélectionné, il guide l'ingénieur des connaissances dans le processus d'acquisition des connaissances en fournissant une structure et des conseils. La connaissance du domaine acquise, ainsi que le modèle d'interprétation, est appelée modèle conceptuel (également appelé modèle d'expertise).
- Le modèle de conception : Le modèle conceptuel est transformé en un ensemble de techniques AI appropriées.

C'est un modèle qui comprend trois couches : les concepts du domaine d'application représentés par le niveau domaine, leurs transformations représentées par le niveau inférence et la structure de contrôle représentée par le niveau tâche. voir figure.1.6.

### 1.7.2 Méthodologie CommonKADS

La méthodologie CommonKADS est le résultat du projet européen ESPRIT KADS-I et KADS-II (Schreiber, 2000). Il sert à modéliser les systèmes experts en termes de trois types de connaissances, de connaissances de domaine, de connaissances d'inférence et de connaissances de tâches (Prat et al., 2012).

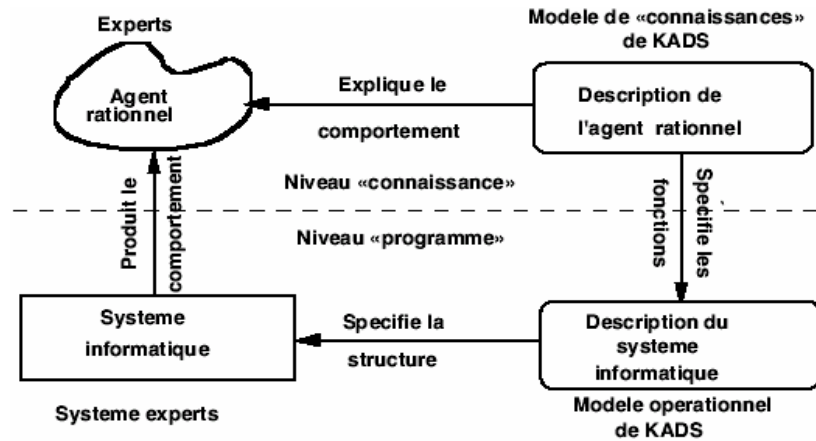


Figure 1.6 Méthodologie de KADS (Wielinga et al., 1992).

Il existe trois structures de couche dans CommonKADS: méthodologie, contexte, concept et niveau d'artefact. Il utilise six modèles: modèles d'organisation, d'agent et de tâche dans le contexte, modèles de communication et de connaissances en niveau conceptuel, modèle de conception en niveau artefact (Figure 1.7).

Une première analyse soutient et définit le problème, les opportunités pour le système de connaissances par modèle organisationnel. Une deuxième analyse décrit les tâches globales réalisées par le modèle de tâche.

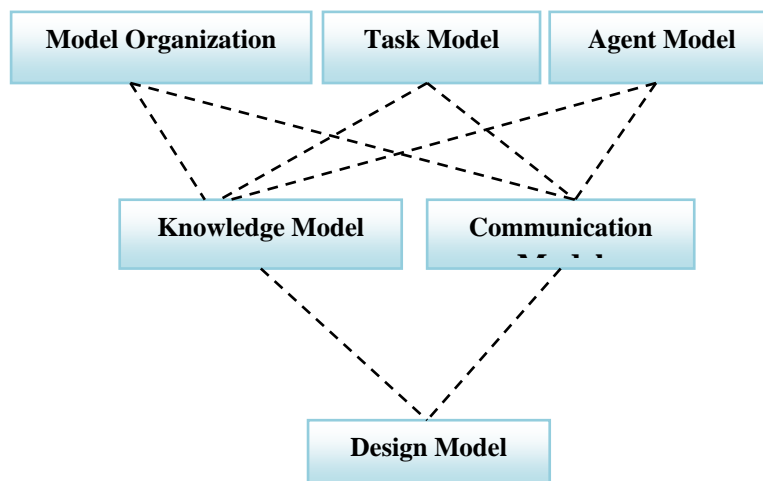


Figure 1.7 Modèles CommonKADS (Schreiber, 2000).

Le modèle d'agent décrit tous les agents et exécutants des tâches, le modèle de connaissance pour décrire en détail les structures et les types de connaissances dans la tâche exécutée. Les transactions communicatives entre les agents donnés par le modèle de communication, le modèle de conception décrivent une spécification



d'architecture et de plate-forme d'implémentation, modules logiciels, dans les modèles de connaissance et de communication.

Le modèle de connaissance facilite la structure d'une tâche de traitement de l'information à forte intensité de connaissance. Ce modèle est structuré de manière similaire aux modèles d'analyse traditionnels en génie logiciel. Il se compose de trois catégories de connaissances: la connaissance des tâches, la connaissance d'inférence et la connaissance du domaine. La connaissance du domaine est clairement le domaine de l'expert du domaine et peut être provoquée par des techniques assez simples. Le schéma de domaine et la connaissance d'inférence sont le domaine de l'ingénieur du savoir qui interprète les étapes de raisonnement de l'expert humain en termes de processus cognitifs (Wielinga, 2013). CommonKADS a tenté de fournir un cadre pour décrire les structures du savoir et les processus de raisonnement correspondant dans les modèles semi-formels au niveau du savoir. De tels modèles ont été couronnés de succès dans des applications basées sur le savoir à petite échelle (Wielinga, 2013).

Notre travail (Titah et al., 2014) a consisté à externaliser les connaissances tacites en connaissances explicites dans une centrale thermique. Pour notre analyse, nous avons utilisé la méthodologie d'acquisition de connaissances «CommonKADS», mais les points faibles sont (i) l'absence d'outil de mise en œuvre de cette méthode, (ii) le langage de modélisation CML (Conceptual Modeling Language) est faible, car il est un langage semi-formel, et (iii) le manque d'inférence (le rôle de basé sur la connaissance). Nous avons donc proposé l'expert générateur de système G2 comme modèle d'expertise informatisé pour cette méthodologie; c'est un développement très efficace assistant de systèmes basés sur la connaissance.

Cela vient du fait qu'il contient un langage naturel et formel. Il est structurée et permet de définir tous les éléments de la méthodologie CommonKADS, il offre plus de possibilités qu'un moteur d'inférence. La centrale thermique étudiée utilise un système de surveillance en ligne; il fait la détection des signes qui montrent des anomalies à l'aide d'alarmes. Nous avons proposé un système basé sur la connaissance qui suit la détection pour diagnostiquer en temps réel le système de gestion des alarmes, et se traduit par la qualité d'équipement de surveillance et de diagnostics rapides.

Une expertise sauvegardée devrait permettre un meilleur ajustement des interventions. Notre contribution est en la conduite et l'accompagnement du diagnostic d'un système de production. La proposition est un outil de mise en œuvre de CommonKADS, basé sur l'amélioration de ses faiblesses.

Notre travail (Titah et al., 2014) est appliqué à une tâche spécifique, et présente les points faibles de cette méthodologie, le travail de (Said Saleh et al. 2018) est basé sur la, génération de notre application , il ont amélioré la réutilisabilité de ses applications est introduite. Cette amélioration contient une adaptation de la méthodologie CommonKADS et l'utilisation de l'architecture orientée services (SOA) en tant que technologie d'ingénierie logicielle prometteuse. Un exemple d'application, un système expert basé sur CommonKADS de la pomme de terre, a été utilisé pour évaluer pour le modèle proposé.

### 1.8 Techniques et outils de modélisation d'une Base de connaissances

Une étude a faite par (Bimba et al., 2016) dans l'objectif d'identifier toutes les différentes techniques de modélisation et les technologies de manipulation d'une base de connaissances avec ses outils techniques et informatiques, ils ont partagé ces méthodes et techniques de modélisation en quatre catégories (voir la figure 1.8);

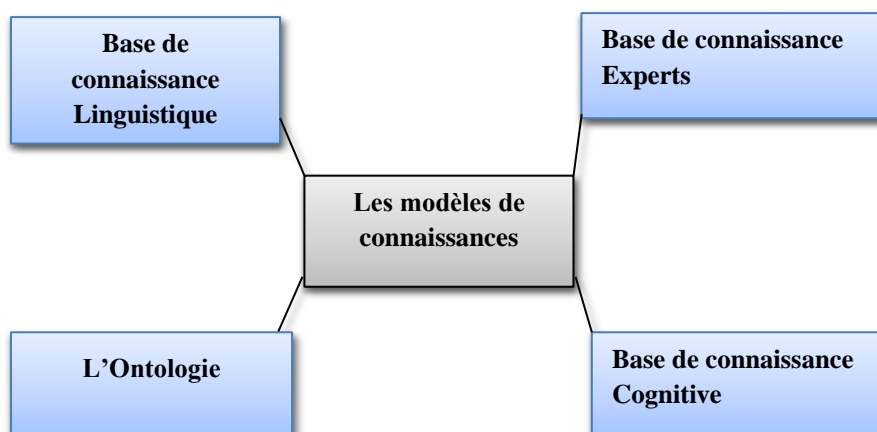


Figure 1.8 Classification des modèles de connaissance et techniques manipulation. (Bimba et al., 2016)

- 1) modélisation à base des systèmes experts;
- 2) modélisation à base des systèmes linguistiques;
- 3) modélisation à base des ontologies
- 4) modélisation à base des systèmes cognitifs.

Cette étude est axée sur les différentes catégories de technologies de modélisation d'une base de connaissances, leurs théories sous-jacentes, la technique de

représentation du savoir, la technique d'acquisition de connaissances, les défis, les applications, les outils de développement et les langages de développement.

### 1.8.1 Systèmes experts

Le modèle de connaissance basé sur le système expert se compose de connaissances de domaine utiles pour la résolution de problèmes. Cette connaissance est représentée comme un ensemble de règles dans la base de connaissances spécialisées. Les règles sont considérées comme une technique relativement simple pour représenter la connaissance.

Toute règle se compose de deux parties: la partie IF appelé l'antécédent et la partie THEN appelée suite. Pratiquement, une règle peut avoir plusieurs antécédents reliés par une (AND) ou une disjonction (OR) ou une combinaison des deux. L'antécédent d'une règle comprend deux parties; un objet linguistique et sa valeur. L'objet et sa valeur sont liés par un opérateur. L'opérateur identifie l'objet linguistique et attribue la valeur. Les règles peuvent représenter les relations, les recommandations, les directives, les stratégies et les heuristiques (Negnevitsky, 2005). Les systèmes experts sont classés en tant que :

- Les systèmes à base de règles logiques «logical rule-based System» (LRS).
- Les systèmes à base de règles floues «fuzzy rule-based System» (FRB).

#### 1.8.1.1 Systèmes à base de règles logiques

La connaissance est représentée comme une logique binaire dans un système basé sur une règle logique (Selva, et al., 2014). Dans ce système, si l'antécédent est vrai, le résultat est également vrai (Negnevitsky, 2005). La plupart du temps, dans une LRS, les connaissances sont acquises manuellement par des experts du domaine par des entretiens, où les experts communiquent leurs connaissances en utilisant des questionnaires (Selva & Crawley, 2012).

#### 1.8.1.2 Systèmes à base de règles floues

Les ensembles flous sont utilisés pour représenter les connaissances dans un système basé sur des règles floues. Contrairement aux systèmes basés sur des règles, dans le système à base de fuzzy rule, si l'antécédent de la règle est vrai, alors le

résultat pourrait être partiellement vrai. La logique floue est développée comme une méthode pour exprimer et appliquer la connaissance humaine sous une forme qui reflète la pensée d'un expert, qui est représentée par des termes tels que, en général, rarement, parfois, souvent et occasionnellement (Negnevitsky, 2005). L'acquisition de connaissances dans les systèmes flous peut être d'origine humaine experte ou informatisée (Zhang & Mahfouf, 2011).

### **1.8.2 Systèmes linguistiques**

La base linguistique est un moyen par lequel les pensées exprimées par l'homme, donc pour modéliser les processus cognitifs humains, un lexique est nécessaire. Cependant, à l'origine, les linguistes ont divisé la connaissance humaine de la grammaire en phonologie, morphologie, syntaxe, sémantique et le lexique. Ces aspects de la grammaire qui sont dirigés par des règles claires autorisent la linguistique computationnelle à concevoir des analyseurs, qui identifient des constituants de phrases, en leur attribuant une structure de phrases. Mais, en raison de la grande taille du lexique. Les bases de connaissances linguistiques typiques sont des bases de données lexicales telles que FrameNet, WordNet et ConceptNet.

- La base Frame Net (FN) : a été développée dans le but de construire un lexique de l'Anglais qui est compréhensible par l'Homme et la machine, en utilisant la théorie de la sémantique des trames et soutenue au moyen d'un capot annoté d'éléments lexicaux (Lakhfif et Laskri, 2015).
- La base Concept Net (CN) : est une base de connaissances à grande échelle qui comprend le savoir humain et son expression sous forme de graphique et sémantique (Agarwal et al., 2015).
- La base Word Net (WN) : est une autre base de données lexicale électronique à grande échelle conçue sur la base de l'organisation sémantique humaine, où les mots et leurs significations sont liés les uns aux autres via des similitudes sémantiques et flexibles (Fellbaum et al., 2006).

### 1.8.3 Ontologies

L'ontologie est une approche de la modélisation des connaissances en termes de modélisation et une des importantes composantes de la technologie sémantique, elle est largement utilisée pour le partage des connaissances dans les organisations. Alors que les modélisations à base linguistiques représentent la connaissance par des relations lexicales et sémantiques, la modélisation à base des systèmes experts représente les connaissances par des règles logiques et floues, les ontologies représentent la connaissance comme taxonomie des concepts avec leurs attributs, leurs valeurs et leurs relations (Studer et al., 1998). La classification des ontologies selon (Van Heijst et al. 1997) est basée sur l'objet de la conceptualisation, ils ont classifié les ontologies en quatre catégories principales : les ontologies d'application, les ontologies de domaine, les ontologies génériques, les ontologies de représentation. Dans cette section nous avons présenté l'ontologie comme une des techniques de modélisation d'une base de connaissances. Pour plus de détails, voir chapitre 2.section2).

### 1.8.4 Systèmes cognitifs

La modélisation de la base de connaissances, telles que la base de connaissances linguistiques, la base de connaissances experte et l'ontologie, sont fabriquées par l'Homme plutôt que par la machine.

La base de connaissances cognitives est une structure qui définit la connaissance comme un réseau de concept dynamique comme le traitement de la connaissance humaine (Wang, 2008). Le développement des systèmes cognitifs est en cours de développement. Cette modélisation est basée sur l'algèbre conceptuelle.

## 1.9 Capitalisation des connaissances dans les systèmes industriels

Un ensemble de modèles de connaissances appliqués dans les systèmes de production industrielle est présentée dans le tableau 1.2.

Tableau 1.2 Les modèles de connaissances récents dans l'industrie manufacturière

L'auteur	Description	Technique de modélisation	Type de connaissances	Cas d'application	limitations
Chen et al., 2005	Proposer un modèle de connaissance pour gérer et générer des connaissances à partir d'informations sur l'expérience	Les ontologies	explicite	Maintenance des ponts roulants	Problème dans le cas utilisant des bases de données plus complexes avec plus d'attributs pour décrire une expérience.
Rasovska et al., 2008	Contribution à une méthodologie de capitalisation des connaissances basée sur le raisonnement à partir de cas	CBR/	explicite	diagnostic dans une plateforme d'e-maintenance.	Problème de capture des connaissances.
Chen 2010	Il a développé une méthode qui adopte un langage d'ontologie Web(OWL) pour représenter les connaissances empiriques.	Une ontologie Représentation/ raisonnement	explicite	diagnostic Financier en fabrication	Problème de raisonnement et d'inférence raisonnement.
Hunter Alarcón et al. (2010)	Une approche proposée pour développer un modèle de connaissances appliqué à la conception des fixations.	CommonKADS, MOKA, UML, IDEF0	explicite	Conception du processus de fixation.	Problème de la mise en œuvre d'algorithmes pour évaluer les contraintes.
Zhou et al. (2011)	Propose une fonctionnalité basée sur la méthodologie conception de l'industriel pour la fabrication des pièces structurelles d'avions.	Case Base Reasoning CBR.	explicite	Conception du processus de fixation pour la fabrication des avions.	Difficulté de capturer et de réutiliser les connaissances.

Tableau 1.2 Les modèles de connaissances récents dans l'industrie manufacturière (Suite)

L'auteur	Description	Technique de modélisation	Type de connaissances	Cas d'application	limitations
Louis-Sidney et al., 2012	Proposition d'un modèle et outil de capitalisation des connaissances en conception : contribution au management et à l'ingénierie des connaissances	MASK/ UML	explicite	L'industrie automobile Renault - DCT	Capture des connaissances. Perte de connaissances tacites.
Maksim et al. (2014)	Défis industriels dans la gestion des connaissances en développement de produits du point de vue des concepteurs et ingénieurs.	le partage des connaissances, la capture des connaissances.	explicite	Développement de produits.	Problème d'adaptation et de validation industrielle des aspects critiques des activités du cycle de vie des connaissances.
Mourtzis and Doukas (2014)	Capture et réutilisation des connaissances pour soutenir la fabrication de produits sur mesure: une étude de cas de la fabrication de moules	Un cadre de travail basé sur les connaissances pour la fabrication avancée.	Explicite	l'industrie de fabrication de moules	L'absence d'ontologies ou de bases de cas propres au domaine en termes de niveau de détails.
Ruiz et al. (2014)	Propose un cadre pour gérer et générer des connaissances à partir d'informations sur l'expérience passée.	Les ontologies (graphe conceptuel)	Explicite	Maintenance des ponts roulants	Problème dans le cas utilisant des bases de données plus complexes avec plus d'attributs pour décrire une expérience.
Titah et al., 2014	consisté à externaliser les connaissances tacites en connaissances explicites dans une centrale thermique.	Système expert en/Commonkads/CML	Tacite /Explicite	Gestion des alarmes à la centrale thermique.	Problème de partage des connaissances. la modélisation logique. Perte d la connaissance tacite.

**Tableau 1.2** Les modèles de connaissances récents dans l'industrie manufacturière (Suite)

L'auteur	Description	Technique de modélisation	Type de connaissances	Cas d'application	limitations
do Rosário et al. (2015)	Construire la méthodologie, qui à transformer la connaissance tacite collective en explicite en utilisant les techniques d'externalisation des connaissances.	La méthodologie MACTAK./système expert logique et flou.	tacite	Fabrication métallurgique	L'implicité aligné avec Internet de des objets. Perte de sens de l'expertise. Problème de partage des connaissances.
Manel Bricni et al.,2015	propose un cycle d'amélioration continue de capitalisation des connaissances. Son objectif est de capitaliser efficacement et en permanence les connaissances tout en ciblant les besoins de l'entreprise et en fournissant une solution évolutive.	STIKI	explicite	Un cas d'étude à STMicroelectronic	Problème de représentation des connaissances. Problème d'intégration des big data dans ce modèle.
Ben Saïd et al.,2015	Gestion dynamique des connaissances de maintenance pour des environnements de production de haute technologie à fort mix produit	les réseaux bayésiens et la méthode AMDEC	explicite	Cas d'application d'apprentissage de nouvelles structures causales	Problème d'acquisition et réutilisation des connaissances
Menaouer et al.,2015	Une approche d'accompagnement à l'innovation guidée par la capitalisation des connaissances.	MASK/TRIZ	explicite	L'industrie des engrais	Problème de partage des connaissances. Problème d'acquisition des connaissances.



Tableau 1.2 Les modèles de connaissances récents dans l'industrie manufacturière (Suite)

L'auteur	Description	Technique de modélisation	Type de connaissances	Cas d'application	limitations
Thramboulidis and Christoulakis (2016)	Une approche basée sur un profil UML pour l'internet des objets (IoT) est présenté pour automatiser entièrement le processus de génération de la couche compatible IoT requise pour le composant cyber-physique.	UML4IoT - une approche basée sur UML.	explicite	Un prototype de mise en œuvre du système de laboratoire de production.	Le problème de la mise en œuvre des études de cas du monde réel.
Mourtzis et al. (2016)	Proposition d'une résolution collaborative de problèmes basée sur les réseaux sociaux utilisant l'analyse des causes profondes (RCA) accessible via des appareils mobiles.	Réseaux Sociaux/(Root Cause Analysis) RCA	Tacite/explicite	Domaine de construction d'automobile.	Problème d'engagement à signaler de nouveaux problèmes et à ne pas les distraire avec l'utilisation d'appareils mobiles en atelier.

## **1.10 Conclusion**

Dans ce chapitre, nous avons présenté un état de l'art sur la capitalisation des connaissances par les techniques d'acquisition et de modélisation des connaissances. A travers ce chapitre, on s'est concentré sur : (1) la gestion des connaissances dans l'entreprise, la notion des données, information et connaissance, (2) la notion des connaissances tacites et explicites, (3) les techniques d'acquisition des connaissances, (4) l'ingénierie des connaissances, (5) techniques et outils de modélisation d'une base de connaissances et (6) la revue de capitalisation des connaissances dans les systèmes industriels.

Les outils actuels existants ne traitent pas efficacement le processus de capitalisation des connaissances. Le raisonnement humain ou perte des connaissances tacites des experts entraîne des coûts énormes pour les organisations. De plus, ils ont besoin d'un modèle qui converge vers la représentation de la connaissance comme raisonnement de l'être humain.

Les principales limites dans les travaux de recherche sont également présentées dans ce chapitre.

Celles-ci sont énumérées ci-dessous :

- Absence de méthodologie de développement et d'externalisation des connaissances tacites en connaissances explicites suffisamment validée et généralisée.
- Difficulté à capturer des connaissances d'experts.
- Fragilité des règles.
- Difficulté à maintenir une large base de règles.
- Problème d'efficacité d'inférence.
- Difficulté à capturer les connaissances d'experts.

Le chapitre 3 propose un modèle de capitalisation des connaissances basé sur l'ingénierie des connaissances des tâches manufacturières pour traiter ces limites dans la recherche.

## ***Chapitre 2 : Processus du partage des connaissances manufacturiers : Approche ontologique***

### Sommaire du chapitre 2 :

2.1	Introduction.....	33
2.2	Partage des connaissances dans l'entreprise.....	33
2.3	Partage des connaissances par les ontologies.....	34
2.3.1	Notion d'ontologie.....	34
2.3.2	Composants d'ontologie.....	36
2.3.3	Classification d'ontologies.....	36
2.3.3.1	Classification selon la structure de conceptualisation.....	37
2.3.3.2	Classification selon l'objet de conceptualisation.....	39
2.3.4	Ingénierie des ontologies.....	41
2.3.4.1	Méthodologie d'Uschold et King.....	41
2.3.4.2	Méthodologie de Gruninger et Fox.....	42
2.3.4.3	METHONTOLOGY.....	43
2.3.4.4	Méthodologie Noy et McGuinness.....	43
2.3.5	Ontologies et raisonnement.....	45
2.3.6	Formalisme d'ontologies.....	45
2.3.6.1	Langage RDF.....	46
2.3.6.2	Langage OWL.....	48
2.3.6.3	Langage SWRL.....	48
2.3.7	Outils d'ontologie.....	48
2.4	Interrogation de l'ontologie.....	50
2.4.1	Requêtes SQWRL.....	50
2.4.2	Requêtes SPARQL.....	50
2.5	Ontologies aux systèmes industriels.....	51
2.6	Conclusion.....	62

## **2.1 Introduction**

Actuellement, le monde vit une évolution rapide des technologies de l'information, cette évolution influe sur les entreprises industrielles, les chercheurs en industries intègrent cette technologie pour l'amélioration de la performance des systèmes de production, telle que l'ontologie manufacturière 'Manufacturing ontology' (Usman et al., 2013), Cloud manufacturier (Xun Xu, 2012), l'industrie sociale 'Social manufacturing' (Markko Hamalainen and Jesse Karjalainen, 2017) et l'industrie des objets 'Industrial Internet of Things' (Sylwia Gieriej, 2017). De part cette évolution, nous devons capitaliser et partager les connaissances dans les entreprises manufacturières, en particulier pour améliorer la résolution des problèmes. Ainsi, le processus de connaissance doit s'intégrer à cette évolution. Du point de vue gestion des connaissances, la connaissance explicite documentée est facile à partager et à gérer grâce à l'utilisation des technologies de l'information et de la communication (TIC) (Lai., 2005). Considérant que la nature non structurée des connaissances tacites rend difficile leur gestion et leur partage aisé par au moins les systèmes traditionnels de gestion des connaissances (KMS) (Lai., 2005).

Ce chapitre comporte cinq sections qui traitent d'un état de l'art sur le partage des connaissances manufacturières par les ontologies, le partage des connaissances par les ontologies, l'interrogation des ontologies par les requêtes SQWRL et SPARQL et se termine par les modèles du partage des connaissances liés aux systèmes industriels.

## **2.2 Partage des connaissances dans l'entreprise**

Le partage des connaissances est le domaine le plus important de la gestion des connaissances, plusieurs définitions existent dans la littérature :

Selon (Davenport and Prusak, 1998, Snejina Michailova and Akshey Gupta 2005) le partage des connaissances est « le fait de transmettre ses connaissances aux autres et de recevoir les connaissances des autres». Aussi, (Bukowitz and Williams.; 1999) ont défini le partage des connaissances comme étant « une activité par laquelle les connaissances (telles que l'information, les compétences ou l'expertise) sont échangées entre des personnes, des amis, des familles, des communautés ou des organisations» (Mustafa I.M. Eid and Ibrahim M. Al-Jabri., 2016).

Selon [Lee \(2001\)](#), le partage des connaissances a été défini comme étant « des activités de transfert ou de diffusion des connaissances d'une personne, d'un groupe ou d'une organisation à une autre ». Selon ([Hooff and Ridder, 2004](#)) le partage des connaissances est : « le processus par lequel les individus échangent mutuellement leurs connaissances et créent conjointement de nouvelles connaissances ».

Le processus du partage des connaissances est un facteur important pour améliorer la performance des entreprises, particulièrement dans le domaine industriel.

Les difficultés du partage des connaissances entre les individus de différents départements d'une organisation sont causées par des différences dans leurs langues, leurs contextes de travail et la compréhension du produit de leurs propres points de vue ([Benchky., 2003](#)).

### **2.3 Partage des connaissances par les ontologies**

La notion d'ontologie a été proposée depuis le début des années 90, elle a suscité l'intérêt des chercheurs dans plusieurs pays. Les ontologies sont incluses dans le domaine de l'Intelligence Artificielle (IA) et particulièrement dans le domaine de l'Ingénierie des Connaissances (IC). Elle fait partie du web sémantique.

L'objectif principal de l'ontologie est de fournir une plate-forme qui facilite le partage et la réutilisation des connaissances entre les groupes sous une forme computationnelle ([Apisakmontri, Nantajeewarawat, Ikeda et Buranarach, 2016](#)). C'est aussi une approche de modélisation et représentation des connaissances.

#### **2.3.1 Notion d'ontologie**

Dans la littérature, Il existe plusieurs définitions de la notion d'ontologie. Selon ([Gruber, 1993](#)) « l'ontologie est une spécification explicite d'une conceptualisation ». Le terme est emprunté de la philosophie, où l'ontologie est un ensemble de choses existantes. Pour les systèmes d'IA, « ce qui existe est ce qui peut être représenté ». Aussi, Studer ([Studer et al 1998](#)) considère que : « l'ontologie est une spécification formelle et explicite d'une conceptualisation partagée » ([Bimba et al 2016](#)). La «conceptualisation» désigne une identification des concepts pertinents d'un modèle abstrait de certains phénomènes dans le monde. Alors que «Explicite» signifie que les concepts identifiés et les contraintes de leur utilisation sont explicitement

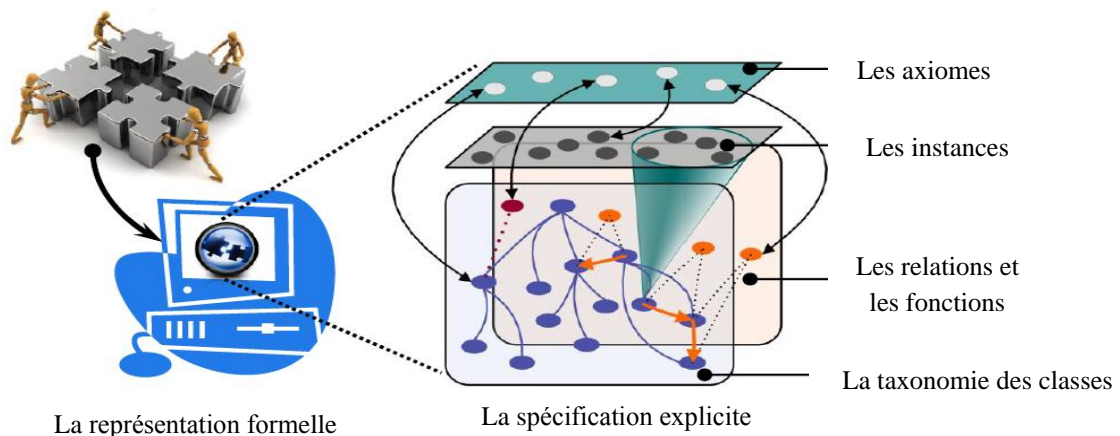
définis (Kimble, de Vasconcelos et Rocha, 2016). Donc, l'ontologie est un terme commun qui présente le sens des concepts et les relations entre ces concepts.

Dans le domaine de la communication et la technologie d'information (CTI), « l'ontologie est une représentation d'information du domaine ». L'objectif est d'assurer la spécification explicite des connaissances au niveau conceptuel en utilisant un langage formel offrant une sémantique qui peut être plus ou moins rigoureuse, permettant une utilisation non ambiguë des connaissances du domaine (Bahloul 2006).

Une autre définition pertinente pour l'ontologie est celle fournie dans l'ISO 18629 (2005), affirmant qu'une ontologie est « un lexique de la terminologie spécialisée avec quelques spécifications de la signification des termes dans le lexique ».

Une définition basée sur la vue structurelle d'une ontologie selon (Labrou, 2002, Gómez-Pérez et al, 2004) « une ontologie est considérée comme étant un modèle multidimensionnel de domaines d'intérêt ». Cette définition regroupe des éléments de différentes définitions. La figure 2.1 identifie la nature multidimensionnelle d'une ontologie.

La Conceptualisation partagée



**Figure 2.1** La nature multidimensionnelle d'une ontologie

### 2.3.2 Composants d'une ontologie

Les principaux composants d'une ontologie sont : des concepts et des relations ainsi que des propriétés et des axiomes.

- **Les concepts** : les concepts ou objets sont des notions qui représentent la description d'une action ou d'une tâche ou d'une fonction ou stratégie ou un mode de raisonnement. Ils peuvent être abstraits ou concrets (Bahloul 2006).
- **Les relations** : les relations représentent les liens entre les concepts.
- **Les propriétés** : les propriétés ou attributs sont des restrictions des concepts ou des relations.
- **Les axiomes** : les axiomes représentent des formules logiques permettant de définir la sémantique des classes et relations et leurs propriétés.
- **Les instances** : les instances ou individus sont utilisés pour représenter les éléments du concept.

### 2.3.3 Classification d'ontologies

Dans la littérature, il existe plusieurs types d'ontologies concernant différents critères. Ces critères de catégorisation sont classés selon : la formalité, le type de structure de conceptualisation, le sujet de Conceptualisation, la généralité et la spécificité des connaissances. Un résumé des types et des critères des classifications des ontologies développé par (Zhou and Dieng-Kuntz 2004) est présenté dans le tableau 2.1.

Dans le cadre de cette thèse, deux critères principaux pour la classification des ontologies sont traités : (1) le critère de structure de conceptualisation et (2) le critère de sujet de conceptualisation. Le tableau 2.2 présente les types d'ontologies par rapport à ces critères. Une discussion sur ces types est présentée dans les sections 2.4. 1 et 2.4.2.

Table 2.1: les types d'ontologies adapté de (Zhou and Dieng-Kuntz 2004)

Les Auteurs	Les critères de classification	Les types d'ontologies
Mizoguchi 1995.	Critère de Réutilisation	L'ontologie du contenu
	Critère de Partage	L'ontologie du Communication
	Critère de récupération	L'ontologie d'Indexation
	Critère de Représentation	Meta ontologie
Uschol et al. 1996.	Critère de Formalité	L'ontologie Très informelle
		L'ontologie Semi-informel
		L'ontologie Semi-formel
		L'ontologie très formel
Van Heijst et al, 1997.	Critère de structure de conceptualisation	L'ontologie de Terminologique
		L'ontologie d'Information
		L'ontologie de Modélisation des connaissances
	Critère de Sujet de Conceptualisation	L'Ontologies d'application
		L'Ontologies de domaine
		L'Ontologies génériques
		L'Ontologies de représentation
	Guarino 1998.	Critère de Généralité
L'ontologie de Tâche		
L'ontologie de Domaine		
L'ontologie d'Application		
Gomez-Perez et al. 2004.	Critère de Formalité	Ontologies Légères
		Ontologies lourdes
Gangemi & Borgo., 2004.	Critère de spécificité	L'ontologie de Fondations
		L'ontologie Principale
		L'ontologie de Domaine

### 2.3.3.1 Classification selon la structure de conceptualisation

Van Heijst (Van Heijst et al. 1997) ont classé l'ontologie selon l'objet de la conceptualisation. Ils ont réparti les ontologies en trois catégories principales: (i) les ontologies terminologiques, (ii) les ontologies d'information et (iii) les ontologies de modélisation des connaissances.



Tableau 2.2: classification des ontologies selon (Van Heijst et al. 1997)

Van Heijst et al. 1997	Critère de structure de conceptualisation	L'ontologie de Terminologique
		L'ontologie d'Information
		L'ontologie de Modélisation des connaissances
	Critère d'objet de Conceptualisation	L'Ontologies d'application
		L'Ontologies de domaine
		L'Ontologies génériques
		L'Ontologies de représentation

### 2.3.3.1.1 Ontologies terminologiques

Les Ontologies terminologiques tels que les lexiques, les termes utilisés pour représenter la connaissance dans le domaine du discours et communication sont précis. Un exemple d'une telle ontologie dans le domaine médical est le réseau sémantique en UMLS (Unified Medical Language System).

### 2.3.3.1.2 Ontologies de l'information

Les ontologies d'informations concernant la modélisation et le développement du système d'information sont spécifiques à la structure d'enregistrement des bases de données. Les schémas de base de données sont un exemple de cette classe d'ontologies. Le modèle PEN & PAD proposé par (Rector et al, 1993), permet de modéliser les dossiers médicaux des patients. Le niveau 1 de ce modèle est un exemple typique d'une telle ontologie dans le domaine médical. A ce niveau, le modèle fournit un cadre pour enregistrer les observations de base des patients, mais il ne fait pas de distinction entre symptômes, signes, traitements, etc.

### 2.3.3.1.3 Ontologies de modélisation des connaissances

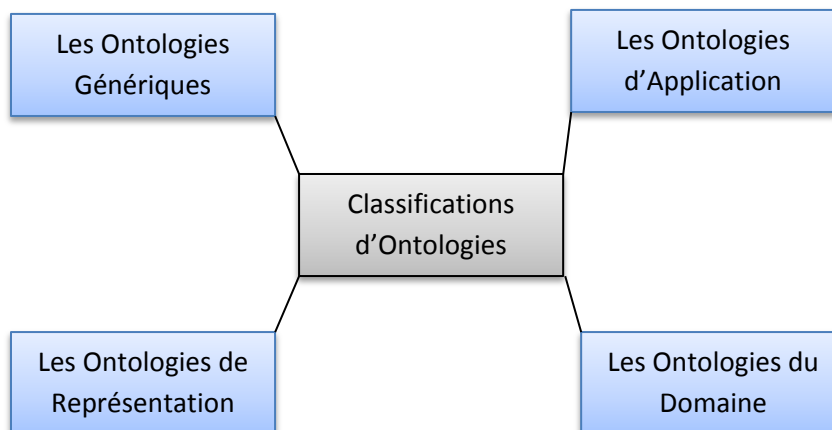
Les ontologies de modélisation des connaissances concernant la modélisation et le développement du système à base de connaissance spécifient des conceptualisations des connaissances. Par rapport aux ontologies de l'information, les ontologies de modélisation des connaissances ont généralement une structure interne plus riche.

En outre, ces ontologies sont souvent adaptées au développement du système à base de connaissances à une tâche spécifique. Dans le contexte de conception de KBS, par

exemple la méthodologie CommonKADS (SCHREIBER et al. 1994). Les ontologies de modélisation des connaissances sont celles qui nous intéressent le plus.

### 2.3.3.2 Classification selon l'objet de conceptualisation

Le deuxième critère de classification selon (Van Heijst et al. 1997) est basé sur l'objet de la conceptualisation. Ils ont classé les ontologies dans quatre catégories principales (i) les ontologies d'application, (ii) les ontologies de domaine, (iii) les ontologies génériques et (iv) les ontologies de représentation (voir figure 2.2). Les études de Bimba et al. (Bimba et al 2016) classent les ontologies selon ce critère et utilisent l'ontologie comme une des techniques de modélisation d'une base de connaissances.



**Figure 2.2** Classification d'ontologie

#### 2.3.3.2.1 Ontologies d'application

Les ontologies d'application contiennent toutes les caractéristiques nécessaires pour modéliser les connaissances pour un domaine particulier. En règle générale, les ontologies d'application sont un mélange de concepts inspirés d'ontologies de domaine et d'ontologies génériques (décrites ci-dessous). De plus, les ontologies d'application contiennent des extensions spécifiques à une application et à une tâche, elles ne sont pas spécifiques ni réutilisables (Van Heijst et al 1997 ;Savonnet et al 2015; Bimba et al 2016). Une ontologie d'application décrit la relation entre les concepts basés sur des tâches spécifiques (Liu et al 2010). L'acquisition de connaissances dans les ontologies d'application implique l'élimination des

mécanismes de raisonnement utilisés et utilisent des interviews directs avec les experts.

### **2.3.3.2 Ontologies du domaine**

Les ontologies de domaine représentent des conceptualisations spécifiques à un domaine particulier (Van Heijst et al. 1997 ;Bimba et al 2016 ), par exemple, le domaine de la santé (Batet, Sanchez et Valls, 2011; Castilho, Lopes, & Tacla, 2008), le domaine manufacturier (Lemaignan et al., 2006 ; Chang et al., 2010 ; El-Diraby et Osman, 2011 ; Panetto et al., 2012 ; Gunel et al ., 2013 ; Imran and Young., 2013 ; Mazzola et al ., 2016). Ces ontologies sont réutilisables pour plusieurs applications de ce domaine. Plusieurs techniques ont été proposées pour l'acquisition de connaissances dans la construction de l'ontologie de domaine. La méthodologie CommonKADS a été utilisée pour formaliser le processus d'acquisition de connaissances où les principaux concepts et caractéristiques du domaine sont extraits manuellement d'experts à l'aide d'un questionnaire (Labidi & Sergio, 2000 ;Titah et al 2014 ), ou bien avec des techniques d'externalisation des connaissances (Titah et al 2017).

### **2.3.3.2.3 Ontologies génériques**

Les ontologies génériques sont similaires aux ontologies de domaine, mais les concepts qu'elles définissent sont considérés comme génériques dans plusieurs et différents domaines (Van Heijst et al 1997 ;Xing, Li et Liu, 2009; Bimba et al 2016). Les ontologies génériques définissent généralement des concepts tels que état, événement, processus, action, composant, etc. (Van Heijst et al 1997). Les concepts des ontologies de domaine sont souvent définis comme des spécialisations de concepts dans les ontologies génériques.

Par exemple, les ontologies génériques sont utilisées dans les systèmes multi-agents, permettant aux agents d'interagir et de coopérer à travers une ontologie commune (Su, Matskin, & Rao, 2003). PRIMA est une autre ontologie générique qui contient un ensemble de concepts et de relations décrivant le risque et son incorporation dans différents processus de travail (Makki, Alquier et Prince, 2008). L'acquisition de connaissances dans les ontologies génériques est difficile à intégrer complètement, car

elle implique des informations dispersées dans différents systèmes et services. Ainsi, il est principalement semi-automatique, impliquant un expert humain combiné avec des techniques statistiques, syntaxiques et semi-techniques (Makki et al 2008).

#### **2.3.3.2.4 Ontologies de représentation**

Les ontologies de représentation expliquent les conceptualisations sous forme de représentation des connaissances (Van Heijst et al. 1997). Les ontologies de domaine et les ontologies génériques sont décrites à l'aide des primitives fournies par les ontologies de représentation. Un exemple d'ontologie de représentation est proposé par (Benslimane, Bensaber, & Malki, 2005) pour l'interopérabilité sémantique des sources d'information. Il utilise la médiation sémantique pour permettre la définition du système de partage, de l'interrogation, de l'accès uniforme et transparent à diverses sources d'information. L'acquisition de connaissances dans ces ontologies est principalement effectuée manuellement. Traditionnellement, la création du contenu d'ontologies, tels que les concepts et les relations, est faite par des spécialistes, des ingénieurs de la connaissance ou des experts du domaine (Zhou, 2007).

### **2.3.4 Ingénierie des ontologies**

L'ingénierie des ontologies est les méthodologies de développement d'une ontologie, plusieurs différentes méthodes existent dans la littérature. Cette section présente un ensemble des différentes méthodologies de développement d'ontologies.

#### **2.3.4.1 Méthodologie d'Uschold and King**

Cette méthodologie a été développée par (Uschold and King., 1995). Les principales étapes de cette ontologie sont : 1) l'identification du but, 2) la construction de l'ontologie 3) l'évaluation et 4) la documentation.

- 1) l'identification : la première étape consiste à identifier le but potentiel de l'ontologie à développer et la portée de ses utilisateurs prévus.
- 2) la construction de l'ontologie : L'ontologie de construction consiste en trois sous-étapes qui incluent: (i) la capture de l'ontologie (ii) le codage et (iii) l'intégration de l'ontologie existante.

- 3) l'évaluation : cette étape nécessite l'évaluation des ontologies par rapport à un cadre de référence, par exemple. les spécifications de l'exigence, puis en adaptant les ontologies en conséquence.
- 4) la documentation : cette étape vise à développer une documentation adéquate pour faciliter le partage des connaissances.

#### **2.3.4.2 Méthodologie de Gruninger and Fox**

Cette méthodologie a été proposée par (Gruninger & Fox, 1995). Elle contient six étapes : 1) le scénario de motivation, 2) les questions de compétences informelles, 3) la terminologie, 4) les questions de compétences formelles, 5) les axiomes, 6) le théorème de complétude.

- 1) le scénario de motivation : le besoin de développer des ontologies découle des scénarios motivants particulièrement attirés par les problèmes industriels. ces problèmes industriels existent normalement sous la forme de problèmes.
- 2) les questions de compétences informelles : la compétence informelle est un ensemble de requêtes (déclenchées par le scénario de motivation) auxquelles la nouvelle ontologie doit répondre.
- 3) la terminologie : Une fois les questions de compétences informelles proposées, la terminologie de l'ontologie devrait être exprimée dans la logique du premier ordre.
- 4) les questions de compétences formelles : après que les questions de compétence sont définies de manière informelle et que la terminologie de l'ontologie est définie.
- 5) les axiomes : les axiomes logiques du premier ordre définissent des termes ontologiques et aident à appliquer des contraintes à ces termes. Les axiomes constituent une partie essentielle de l'ontologie et décrivent la sémantique des termes utilisés dans la logique du premier ordre.
- 6) le théorème de complétude : une fois les questions de compétence formellement définies, les conditions qui répondent à la solution des questions de compétence sont précisées.

### **2.3.4.3 METHONTOLOGY**

METHONTOLOGY est une méthodologie pour le développement d'ontologies a été proposée par (Fernandez et al., 1997) au laboratoire d'intelligence artificielle (Ontology Engineering Group) à l'université technique de Madrid . Cette méthodologie a été créée dans le domaine des produits chimiques mais elle peut aussi être utilisée comme référence pour d'autres domaines. Les principales étapes de cette méthodologie sont les suivantes :1) Spécification, 2) Conceptualisation, 3) Formalisation, 4) Intégration 5) Implémentation 6) Evaluation 7) Documentation.

- 1) la Spécification : le but de cette étape est de créer un document de spécification exprimé en langage naturel en utilisant les questions de compétences ou en utilisant un ensemble de représentations de concepts intermédiaires.
- 2) la Conceptualisation : l'obtention d'un modèle du domaine au niveau de la connaissance.
- 3) la Formalisation : la transformation du modèle conceptuel en modèle formel
- 4) l'Intégration : la réutilisation d'autres ontologies avec les ontologies existantes.
- 5) l'Implémentation : la phase d'implémentation nécessite un environnement de développement d'ontologies pouvant supporter l'ontologie formelle.
- 6) l'Evaluation : le but de la phase d'évaluation est de faire «un jugement technique des ontologies, de leurs environnements logiciels et des documentations» en référence au document de spécification.
- 7) la Documentation : cette phase nécessite la phase de documentation dans le cycle de développement de l'ontologie, car aucune directive consensuelle n'est disponible pour aider les développeurs à documenter le processus de développement de l'ontologie.

### **2.3.4.4 Méthodologie Noy and McGuinness**

Une méthodologie d'ingénierie d'ontologie simple proposée par (Noy et McGuinness., 2000) décrit une méthode itérative pour la construction d'une

ontologie ; ce processus a énuméré les étapes dans le développement de l'ontologie et les problèmes complexes simplifiés de la définition des hiérarchies de classes, des propriétés des classes et des instances. Il comprend les étapes suivantes : 1) déterminer le domaine et la portée de l'ontologie, 2) envisager de réutiliser les ontologies existantes, 3) énumérer des termes importants dans l'ontologie, 4) définir les classes et la hiérarchie des classes, 5) Définir les relations et les fonctions et 6) Créer des instances.

- 1) déterminer le domaine et la portée de l'ontologie : la première étape de cette méthodologie est de définir le domaine et la portée de l'ontologie, une liste de questions de compétences peut être préparée à cette fin. ces questions peuvent relier le domaine d'intérêt, le but de l'ontologie et le type de requêtes auxquelles l'ontologie devrait répondre.
- 2) envisager de réutiliser les ontologies existantes : la deuxième étape de cette méthodologie suggère la réutilisation d'ontologies existantes. Beaucoup d'ontologies existantes sont sous forme électronique et peuvent être importées dans l'environnement de développement d'ontologies.
- 3) énumérer des termes importants dans l'ontologie : la troisième étape de cette méthodologie nécessite l'énumération de tous les termes connexes pour le domaine d'intérêt. Parfois, ces termes se chevauchent, mais les termes qui se chevauchent peuvent également être résolus dans les étapes ultérieures.
- 4) définir les classes et les classes hiérarchies : Une fois les termes importants répertoriés, l'étape suivante consiste à définir les classes et leurs hiérarchies. Les approches descendantes, ascendantes et combinatoires (de haut en bas et de bas en haut) peuvent être utilisées pour développer les hiérarchies de classes.
- 5) Définir les relations et les fonctions : Une fois la hiérarchie des classes et des classes définie, l'étape suivante consiste à définir les relations et les fonctions. les relations et les fonctions aident à ajouter plus d'informations aux classes. la cardinalité des relations et des

fonctions est également définie pour spécifier l'ordre des relations et des fonctions, c'est-à-dire unaire, binaire, etc.

- 6) Créer des instances : la dernière étape de cette méthodologie de développement d'ontologies consiste à créer des instances des classes définies dans l'ontologie.

### **2.3.5 Ontologies et raisonnement**

L'utilisation efficace des ontologies nécessite non seulement un langage bien conçu, mais également un raisonnement logique rigoureux. Par conséquent, construits sur des langages et des théories logiques de description bien définis, les bases de connaissances ontologiques sont dotées d'un formalisme terminologique, appelé «T-Box», et d'un formalisme assertionnel, «A-Box» (Brachman et al., 1983). Pour les ontologies informatiques dans les systèmes d'information, les ontologies signifient deux choses liées (Chandrasekaran et al., 1999): un vocabulaire de représentation (T-box), qui fournit un ensemble de termes permettant de décrire les faits dans un domaine donné; et un ensemble de connaissances (A-box), c'est-à-dire le fait associé à un modèle conceptuel ou à des ontologies au sein d'une base de connaissances. Le raisonnement ontologique réduit la redondance des informations dans la base de connaissances et permet de résoudre les conflits de contenu des connaissances.

«Un raisonneur sémantique, un moteur de raisonnement, un moteur de règles, ou simplement un raisonneur, est un logiciel capable d'inférer des conséquences logiques à partir d'un ensemble de faits ou d'axiomes affirmés». Le Tableau 2.3 présente les différents raisonneurs d'une ontologie

### **2.3.6 Formalisme d'ontologies**

Les ontologies peuvent être représentées sous différentes formes en fonction du niveau d'abstraction ou du degré de formalité. Elles peuvent être exprimées sous forme d'ensembles d'énoncés déclaratifs dans les langues naturelles. Cependant, il est impossible pour les ordinateurs de traiter des déclarations en langage naturel. Pour des représentations plus formelles, le langage d'ontologie Web (OWL) est largement utilisé dans la pratique, appuyé par des outils de modélisation ontologique, tels que Protégé (Musen, 2015).



**Tableau 2.3** les différents raisonneurs d'une ontologie

Le raisonneur d'ontologie	Description
Pellet	Pellet est un raisonneur OWL-DL open source développé par le groupe « Mind Swap » Il est basé sur l'algorithme de tableau et supporte les logiques de description expressives. Pellet prend en charge les profils OWL 2. Il raisonne les ontologies à travers Jena ainsi que les interfaces OWL-API (Sirin et al.2007).
Racer	Le raisonneur RACER (renommé ABoxes et Concept Expression Reasoner). RACER, également connu sous le nom de RacerPro est le premier raisonneur OWL. RACER implémente TBox et ABox reasoner pour la logique SHIQ. Ce raisonneur est développé par (Horrocks et al. 2011). <a href="http://www.franz.com/agraph/cresources/white_papers">www.franz.com/agraph/cresources/white_papers</a> .
FaCT++	FaCT++ est un classificateur de logique de description et pour le test de satisfiabilité de la logique modale. Le système FaCT possède un algorithme de tableaux complet pour les logiques de description expressives. Ce raisonneur utilise le même algorithme que dans FaCT, mais avec une structure interne différente. Il est implémenté en C ++. FaCT ++ implémente une procédure de décision basée sur les tableaux (Sattler, 2007). <a href="http://www.cs.man.ac.uk/~sattler/reasoners.html">http://www.cs.man.ac.uk/~sattler/reasoners.html</a>
Hermit	Hermit est le premier raisonneur OWL accessible au public. Il est écrit en utilisant OWL. Hermit peut vérifier les fichiers OWL pour déterminer la cohérence des ontologies et identifier les relations hiérarchiques entre les classes. Ce raisonneur est basé sur l'hypertableau calculus. Il fournit également le processus plus rapide de classification des ontologies. <a href="http://hermit-reasoner.com/">http://hermit-reasoner.com/</a> .
Snorocket	Snorocket a été développé dans le cadre du programme de recherche sur l'informatique de santé et la terminologie clinique du CSIRO, Il est implémenté en Java (Lawley and Bousquet 2010). <a href="http://research.ict.csiro.au/software/snorocket">http://research.ict.csiro.au/software/snorocket</a> .

Des ontologies bien structurées et bien développées permettent différents types d'examen de cohérence logique et améliorent également l'interopérabilité entre différentes applications.

Plusieurs langages d'ontologie, également appelés formalismes ontologiques ou formalismes de représentation des connaissances, sont aujourd'hui disponibles pour construire des ontologies. Le tableau 2.4 présente différents langages de développement d'une ontologie. Notre travail est basé sur le langage OWL.

### 2.3.6.1 Langage RDF

RDF (Resource description Framework ) est un langage pour représenter des informations sur les ressources dans le web (Brickley., 1998), et aussi une infrastructure permettant d'encoder, échanger, et réutiliser des métadonnées sur le

web. et faciliter l'intégration des données entre différentes sources, même si elles sont hétérogènes. Il fournit la description du sens d'une ressource par des triplets, où chaque triplet associant un sujet (la ressource), à un prédicat (la propriété descriptive) et à un objet (un littéral ou une ressource) dans une phrase élémentaire.

**Tableau 2.4** les différents langages des développements d'une ontologie

Langage de développement d'ontologie	description
OWL	OWL le langage de l'ontologie web est un langage de balisage permettant de développer des ontologies pour le web sémantique, développé par W3C.
XML	XML fournit la syntaxe de la pile de langages sémantiques Web.
RDF, RDF Schema, RDF(S)	la structure de description de ressource est utilisée pour la modélisation des informations à déployer en tant que ressource Web.
UML	UML est une notation graphique orientée objet pour modéliser l'information
F-Logic	la logique de trame est une approche orientée objet de FOL utilisée pour les bases de données déductives et orientées objet
OCML	OCML le langage de modélisation conceptuel opérationnel est un langage qui prend en charge la définition des axiomes du 1er et 2ème ordre.
KIF	KIF format d'échange de connaissances fournit une syntaxe pour laquelle on peut comprendre par ordinateur
CL	CL La logique commune est décrite comme un cadre pour une famille de langage basée sur la logique.
CGIF	CGIF format d'échange de graphes conceptuels est un dialecte de CL utilisé pour représenter des graphes conceptuels.
CLIF	CLIF Le format d'échange de logique courant est la syntaxe de type KIF utilisée dans CL
KFL	KFL le langage du cadre de connaissances est basé sur une implémentation étendue de CLIF appelée ECLIF

On peut écrire ces triplets en utilisant des balises XML. RDF représente la connaissance par des triplets <sujet — prédicat — objet> :

- a) les ressources (Sujet) : Cela peut être tous objets identifiés avec un URI, qu'il concerne le web. Une ressource peut être un objet, une page web ou même un lieu, une personne.
- b) les propriétés (prédicat) : une propriété permettant de caractériser une ressource, attribut ou relation qui peut décrire la ressource.
- c) les valeurs (objet) : est une valeur prise par une propriété pour une ressource donnée qui sera affectée à la propriété de la ressource.

### **2.3.6.2 Langage OWL**

OWL (Ontology Web Language) est une sorte de langage d'ontologie de réseau développé par le W3C (World Wide Web Consortium), qui est utilisé pour vérifier la cohérence des connaissances ou rendre explicite les connaissances implicites (Bechhofer., et al 2004). OWL offre une meilleure interprétabilité du contenu Web par machine que d'autres langages Web sémantiques, tels que XML (Extensible Markup Language), RDF (Resource Description Framework) et RDFS (RDF Schema). Il a la capacité d'exprimer non seulement une sémantique plus puissante mais aussi le raisonnement de la logique de description (DL). Le W3C a conçu trois sous-langages de plus en plus expressifs de OWL, concernant le besoin de toutes sortes de fonctionnalités, à savoir, le sous-langage OWL Lite, OWL DL et OWL Full.

### **2.3.6.3 Langage SWRL**

SWRL (Semantic Web Rule Language) (Horrocks., et al 2004) est basé sur la combinaison de OWL-DL et OWL-Lite, sous-langage de OWL, avec Unary / Binary Datalog RuleML, un sous-langage de RuleML) et a été introduit par le W3C en 2004. SWRL étend l'ensemble d'axiomes OWL permettant de combiner une règle avec une base de connaissances OWL. La syntaxe du langage de règles est relativement similaire à RuleML. Ils peuvent également interagir les uns avec les autres.

Les opérateurs logiques et les supports de quantification de SWRL sont les mêmes que ceux de RuleML. En outre, le contenu RuleML peut faire partie du contenu SWRL. Les axiomes peuvent être constitués d'axiomes RDF, OWL et de règles. Une relation peut être un IRI, une plage de données, une propriété OWL ou une relation intégrée. Un objet peut être une variable, un individu, une valeur littérale ou un nœud vide. De plus, le langage des règles fournit de nombreux ensembles de fonctions intégrées, par exemple fonctions de chaîne et fonctions mathématiques.

### **2.3.7 Outils d'ontologie**

Divers outils de développement d'ontologies ont été mis en place. Un résumé sur les outils de développement d'ontologies est présenté dans le tableau 2.5. Protégé est l'outil utilisé dans cette thèse, il est discuté en détail.

**Tableau 2.5** les différents outils de développement d'une ontologie

Outils d'ontologie	Description/ Développeur
Protégé	Protégé est l'outil de développement d'ontologies le plus utilisé. Il supporte les langages de développement d'ontologies tels que RDF et OWL. Il a une interface personnalisable. Il possède également une architecture de plug-in puissante qui permet l'intégration avec d'autres applications. Cet outil est développé pour les ontologies et les bases de connaissances biomédicales soutenues par l'Institut national des sciences de la médecine générale. <a href="https://protege.stanford.edu/">https://protege.stanford.edu/</a>
OntoEdit	OntoEdit est un éditeur d'ontologies graphique qui intègre de nombreux aspects de l'ingénierie ontologique, et un utilisateur de l'infrastructure de gestion d'ontologies KAON. Il est basé sur la méthodologie commoKADS.
Ontolingua	Ontolingua est un environnement collaboratif distribué pour, créer, éditer, modifier et utiliser des ontologies. il est basé sur le langage KIF. Il peut transférer vers DL. Il n'a pas de moteur d'inférence. Cet outil est développé par laboratoire de système de connaissance, université de Sandford <a href="http://www.ksl.stanford.edu/software/ontolingua/">http://www.ksl.stanford.edu/software/ontolingua/</a>
IODE	IODE - Environnement de développement d'ontologies intégrées Avec le logiciel IODE d'Ontology Works, les utilisateurs développent des modèles de domaine (ontologies) robustes, flexibles et extensibles en fournissant:- Une logique expressive puissante dans laquelle encoder l'ontologie de la matière, quelle que soit la complexité de votre domaine. - Validation du modèle en développement pour assurer la cohérence logique interne, c'est le seul outil commercial, de développement d'ontologie. <a href="http://www.ontologyworks.com/products/iode">http://www.ontologyworks.com/products/iode</a>

Protégé est un éditeur d'ontologie libre et à source ouverte et une plateforme pour la construction de systèmes intelligents. Protégé s'appuie sur une solide communauté d'utilisateurs issus des universités, des gouvernements et des entreprises, qui utilisent Protégé pour créer des solutions basées sur la connaissance dans des domaines aussi divers tels que la biomédecine, le commerce électronique et la modélisation organisationnelle.

L'architecture de plug-in de Protégé peut être adaptée pour créer des applications à base d'ontologies simples et complexes. Les développeurs peuvent intégrer la sortie de Protégé à des systèmes de règles ou à d'autres solutions du problème afin de construire un large éventail de systèmes intelligents.

Protégé est basé sur Java, est extensible et fournit un environnement plug-and-play qui en fait une base flexible pour le prototypage rapide et le développement d'applications.

## 2.4 Interrogation de l'ontologie

### 2.4.1 Requêtes SQWRL

SQWRL « Semantic Query-Enhanced Web Rule Language » est un langage de requête basé sur le langage SWRL pour les ontologies faites en OWL (O'Connor & Das, 2009). SQWRL est construit sur le langage de règles SWRL, SQWRL prend un antécédent de règle SWRL standard et le traite efficacement comme une spécification de modèle. Il fournit un ensemble de fonctions semblable à « SQL » pour extraire des connaissances d'une ontologie en OWL, SQWRL utilise la fonction intégrée de SWRL comme point d'extension. A l'aide de fonctions intégrées, il définit un ensemble d'opérateurs pouvant être utilisés pour construire des spécifications de récupération. L'attrait de cette approche est qu'aucune extension syntaxique n'est requise pour SWRL. Ainsi, les éditeurs SWRL existants peuvent être utilisés pour générer et éditer des requêtes SQWRL. En outre, des mécanismes de sérialisation SWRL standard peuvent être utilisés afin que les requêtes puissent être stockées dans des ontologies OWL. Ces requêtes peuvent être écrites via la SWRLTab de PROTÉGÉ. Le langage SQWRL définit quatre types de requêtes : sqwrl:select, sqwrl:count, sqwrl:orderBy, sqwrl:columnNames.

### 2.4.2 Requêtes SPARQL

Le langage SPARQL (Simple Protocol And RDF Query Language) est une recommandation W3C largement utilisée pour l'interrogation des données en langage RDF (Arias., et al 2011), SPARQL c'est un langage de requête pour RDF, tout comme les données RDF, il est basé sur la notion de graphes de triplets et ils peuvent aussi être représentées par des graphes, les requêtes sont décrites par des motifs (patterns) et des variables. Le langage SPARQL définit quatre types de requêtes :

- ASK : pour tester si un modèle de requête a une solution. Aucune information n'est renvoyée sur les solutions de requête possibles, que la solution existe ou pas et renvoie un booléen, vrai ou faux,
- DESCRIBE : Le modèle de requête est utilisé pour créer un jeu de résultats. Le formulaire DESCRIBE prend chacune des ressources identifiées dans une solution, il retourne des informations sur les résultats de la requête, ainsi que

toutes les ressources directement nommées par IRI, et assemble un seul graphique RDF en prenant une "description" qui peut provenir de toute information disponible, y compris le jeu de données RDF cible. La syntaxe DESCRIBE est une abréviation qui décrit toutes les variables d'une requête.

— CONSTRUCT : Le formulaire de requête CONSTRUCT renvoie un seul graphique RDF spécifié par un modèle de graphique. Le résultat est un graphique RDF formé en prenant chaque solution de requête dans la séquence de solutions, en remplaçant les variables dans le modèle de graphique et en combinant les triplets en un seul graphique RDF par union définie. La structure du graphe résultat est décrite par un patron dans la requête

— SELECT : le modèle de requête SELECT permet d'extraire des informations qui vérifient les contraintes spécifiées dans la requête. La syntaxe SELECT est une abréviation qui sélectionne toutes les variables d'une requête.

## **2.5 Ontologies aux systèmes industriels**

De nombreuses ontologies ont été développées ces dernières années au domaine manufacturier, concernant la conception des produits, le cycle de vie des produits, la tâche spécifique comme la conception, la logistique, la production et la maintenance...etc.

Une ontologie du domaine spécifique à la tâche de production, appelée ADACOR (ADaptive holonic COntrol aRchitecture for distributed manufacturing systems) a été proposée par (Borgo et Leitão 2004). Les concepts ADACOR sont des expressions qui valent pour des entités complexes. Les principaux concepts de l'ADACOR sont :

*“Product”*, *“Raw-material”*, *“Customer order”*, *“Production order”*, *“Work order”*, *“Resource”*, *“Operation”*, *“Disturbance”*, *“Process”* et *“Plan Property”*. L'ontologie d'ADACOR est développée autour d'une ressource (généralement une machine) affectée à un processus spécifique. Cela nécessite que le client ait une connaissance préexistante du processus/service qu'il compte utiliser dans son application. De plus, (Borgo et Leitão 2004) ont été déclaré qu'ADACOR a des instances d'ambiguïté sémantique et de conflits de modélisation.

Les chercheurs (Leimagnan et al. 2006) ont développé une ontologie sémantique manufacturière appelé MASON (MANufacturing's Semantics ONtology), pour permettre la saisie formelle de la sémantique des concepts liés aux industries manufacturières. Une représentation simplifiée du modèle MASON est illustrée dans la figure 2.3.

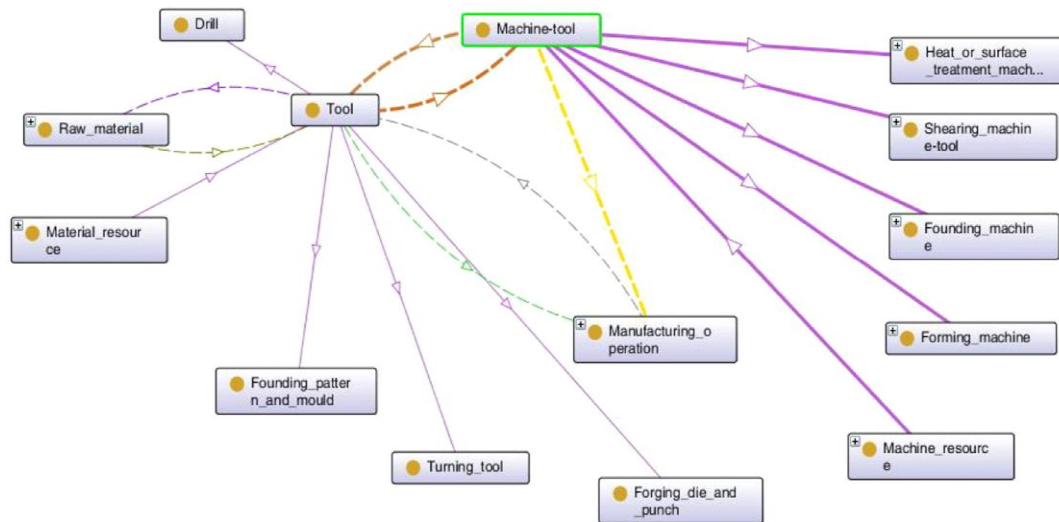


Figure 2.3: les concepts principaux de l'ontologie MASON (Leimagnan et al. 2006)

Les concepts et les relations dans MASON comme présenté par la figure 2.3 peuvent faciliter le développement de la compréhension des concepts et des relations dans les ontologies manufacturières. Les principales classes utilisées sont les classes « entité », « ressource » et « opération ».

La sémantique de ces classes et de leurs relations a été capturée dans une logique formelle à l'aide de langage OWL. L'utilisation OWL permet à MASON d'être plus largement applicable et interopérable. Deux applications de cette ontologie ont été réalisées:

- l'estimation automatique des coûts
- le système de production multi-agents à conscience sémantique.

L'ontologie et l'approche proposée dans MASON, ne fournit pas un système de partage des connaissances durant l'exécution de la tâche, c'est donc un modèle applicable à une tâche spécifique. Cette ontologie est considérée comme une base de développement manufacturières sémantique par les chercheurs de domaine.



Un outil basé sur l'ontologie pour l'échange de données de produit appelé CPM (A Core Model for Product Data) est proposé par (Fenves et al. 2006). CPM est un modèle générique abstrait avec une sémantique générique. Il donne un statut égal à trois aspects d'un produit : sa fonction, sa forme et son comportement. Ainsi, CPM peut prendre en charge:

- un raisonnement purement fonctionnel concernant un produit aux étapes conceptuelles de la conception;
- l'enregistrement et l'analyse de son comportement dans les étapes postérieures à la conception;
- les activités de conception technique «traditionnelles» consistant à générer la forme du produit en réponse à la fonction spécifiée.

Le modèle CDM concentre uniquement sur la capture du modèle de produit. Mais la possibilité de partager des connaissances entre domaines n'a pas été représentée.

Une ontologie d'usinage développée par (Semere et al., 2007), cette ontologie basé sur MASON et étendue pour définir les concepts et les relations pour le domaine de l'usinage. L'ontologie de (Semere et al., 2007) a abordé les problèmes de capture formelle de la sémantique des concepts. Cette ontologie prend en charge la capture des connaissances en usinage seulement.

L'aspect de fabrication des pièces est cependant absent de l'ontologie de Semere, ce qui signifie que les connaissances d'usinage sur les fonctions sont capturées indépendamment de la partie ou de la famille de pièces à laquelle elles peuvent appartenir. Un inconvénient de l'ontologie est que le contexte le partage des connaissances de production dans la conception du produit n'a pas été pris en compte dans cette ontologie.

L'ontologie CAD/CAPP (computer-aided design systems/computer-aided process planning) a été développée par (Dartigues et al (2007) une approche ontologique d'intégration de la conception assistée par ordinateur CAD et de la planification de processus assistée par ordinateur CAPP. Deux applications logicielles commerciales ont été utilisées pour démontrer l'approche. L'approche impliquait le développement d'une ontologie partagée et de méthodologies spécifiques à un domaine utilisant le langage KIF (Knowledge Interchange Format). Les ontologies spécifiques à un domaine, basées sur des fonctionnalités, ont été développées après une analyse détaillée des logiciels de CAD et de CAPP. Le mappage entre les



ontologies de domaine et l'ontologie partagée a été réalisé par plusieurs règles de mappage. Cette ontologie est une intégration entre la tâche de conception et la tâche de planification mais elle reste toujours spécifique entre les deux logiciels commerciaux sans utiliser la connaissance de l'expert humain.

Un modèle d'ingénierie du système manufacturier MSE (Manufacturing System Engineering) a été réalisé par (Lin and Harding 2007). Le modèle d'ontologie MSE est basé sur une technologie complète de Web sémantique par utilisation des ontologies et du langage standard du Web sémantique OWL. Le modèle MSE traite de nombreuses questions inter-entreprises et inter-activités liées aux exigences de l'interopérabilité sémantique de l'information pour le partage de connaissances. Donc il existe le sens de partage des connaissances entre tâche mais une limite pour identifier la connaissance entre deux ontologies.

Une ontologie de la tâche de logistique appelé SCONtology (Supply Chain Ontology) est proposée par (Gonnet et al., 2007). SCONtology est développée pour aider les organisations à obtenir une vue unifiée de la chaîne logistique qui implique la coordination et l'intégration des flux de matériel, d'informations et d'argent, elle est un cadre permettant de décrire formellement un système logistique à différents niveaux d'abstraction, fournit une base pour la spécification des processus de logistique de l'information et propose différents paramètres et concepts liés aux performances, mais une ontologie spécifique à la tâche de logistique ne traite pas les problèmes lié à cette tâche.

Un système de partage des connaissances multicouches basé sur l'ontologie appelé OMKF (ontology-based multi-layered knowledge framework) développé par (Lee et Suh 2008), propose un cadre de connaissances multicouches basé sur des ontologies (OMKF) pour les systèmes de gestion du cycle de vie d'un produit. Ils ont souligné les limites des systèmes de gestion des connaissances actuels en ce qui concerne les ambiguïtés des connaissances partagées.

Ils ont introduit trois types de connaissances, à savoir **les axiomes**, **les cartes de connaissances** et **les connaissances spécialisées** par domaine. Ils ont également identifié quatre types de modèles de produits, à savoir modèle de contexte de produit, modèle spécifique de produit, modèle de planification de produit et modèle de fabrication de produit.

Le modèle de contexte de produit de (OMKF) a formellement défini plusieurs concepts utiles, tels que “Part”, “SolidGeometry”, “GeometricFeature”, “Function”, “ManufacturingProcess” et “ManufacturingMachine”. De tels concepts et leur sémantique formelle peuvent être vraiment utiles pour développer une compréhension des concepts fondamentaux de cette ontologie. L’avantage de ce modèle est d’aider les ingénieurs du savoir à créer, éditer et visualiser les connaissances sur les produits conformément et à inférer de nouvelles connaissances avec un moteur d’inférence PROLOG. Mais, cette ontologie est spécifique au produit et n’a pas de tâche de fabrication. Aussi, manque le mécanisme de raisonnement basé sur les règles de l’expert pour soutenir de meilleures capacités de raisonnement par rapport aux connaissances de fabrication.

Une ontologie de la tâche de conception produit a été développée par (Chang et al., 2008), d’un outil de modélisation graphique destiné à aider les concepteurs au stade de la conception. Afin de fermer la boucle pour aider les concepteurs à générer des concepts de conception de manière flexible, rapide et simple, une approche basée sur une ontologie pour la gestion des connaissances fonctionnant avec l’outil de modélisation graphique est discutée. Mais cette étude concerne seulement la conception des produits et n’a pas traité les problèmes liés à la conception.

Une ontologie de produit pour l’intégration de la tâche de planification avec la tâche de conception de la production a été développée par (Tursi et al., 2009), l’ontologie OPISI (Ontological approach for Products-centric Information System Interoperability) est une nouvelle méthode d’échange d’informations sur les produits pour traiter le produit lui-même comme un système interopérable. Un modèle commun d’intégration des systèmes de planification de la production avec des applications de conception de produits. Les travaux de (Tursi et al., 2009) ont été spécifiquement orientés vers la transformation des nomenclatures avec un accent particulier sur la transformation des nomenclatures d’ingénierie en nomenclatures de fabrication. L’ontologie du produit a été construite par concepts d’emprunt des normes relatives à la gestion des données produits (ISO / TS-10303 2004) et l’intégration d’entreprise (ISO/IEC-6224 2002). Les travaux sont axés sur les nomenclatures, l’assemblage du domaine de planification de la production et de la conception. Cependant, les inclusions de des concepts tels que “Part”, “Product”,

“Component” et “PartVersion” à partir de la norme ISO. L’ontologie des produits de Tursi peut être utile pour identifier des concepts de normes pouvant être intégrées à l’ontologie de fabrication.

Un modèle de machines à outils appelé MTM (Machine-Tools Model) a été développé par (Kjellberg et al 2009) ce modèle joue un rôle important dans le développement de systèmes nouveaux ou déjà existants, allant de l’investissement à la planification, en passant par la planification des processus, la conception, l’utilisation et la fin de vie. Aujourd'hui, de nombreuses informations importantes sur les machines-outils ne sont ni gérées ni stockées, ce qui facilite leur réutilisation. Une approche de modélisation de machine-outil est structurée conformément aux principes des normes génériques utilisant une ontologie de système de fabrication pour la modélisation de concepts de machine-outil. Il constitue une contribution à la gestion de l’information et des connaissances en fabrication.

L’approche proposée est axée sur l’augmentation du niveau de gestion de l’information en vue de la gestion des connaissances en s’appuyant sur des ontologies et des normes d’informations existantes. Le concept principal de modèle MTM est présenté dans la Figure 2.4.

Dans le contexte d’interopérabilité entre la tâche de conception et la tâche de fabrication, un modèle pour partager les connaissances sur les meilleures pratiques de fabrication a été proposé par (Gunendran et Young, 2010), ce modèle permet de capturer les connaissances relatives aux meilleures pratiques de fabrication.

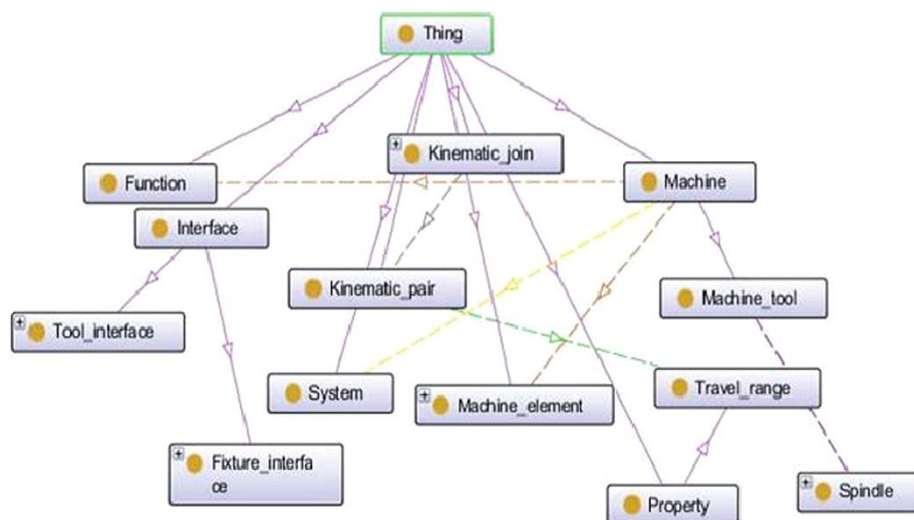


Figure 2.4: Les concepts principaux de modèle MTM adapté de (Kjellberg et al 2009)

Ils ont utilisé des bibliothèques de meilleures pratiques, des modèles de produits et des modèles de fabrication pour prendre en charge cette fonctionnalité. Cela a également exploré les relations entre les caractéristiques de la tâche conception et la tâche de fabrication, et aussi les relations entre les éléments de ces connaissances et leur relation avec les informations produit. Ils ont présenté les problèmes liés aux tâches industrielles. L'une des questions importantes soulignée concernait l'exploitation de cette approche pour identifier les conséquences sur la production lors de la phase de conception du produit. Ils ont également souligné la nécessité d'explorer plus avant les relations entre conception et fabrication. Mais, un problème important consiste à explorer davantage de connaissances en relations étendues entre la fonction de conception et la fabrication. Un problème d'interopérabilité entre les groupes et le système de connaissance, car les connaissances relatives aux meilleures pratiques devraient idéalement être accessibles à partir de sources multiples d'experts.

Une ontologie de conception pour la fabrication appelée DFM (Design for Manufacturing) a été proposée par (Chang et al, 2010), a facilité le développement d'un système d'aide à la décision pour concepteur. Cependant, l'ontologie DFM peut être utilisée pour capturer les connaissances du domaine dans la conception de la fabrication des produits, pour définir un ensemble de données, leur structure et leurs relations pour les concepteurs de produits ayant besoin de partager des informations dans un domaine d'intérêt. Quatre études de cas montrent certaines caractéristiques importantes de l'utilisation de l'ontologie DFM. La pratique a prouvé qu'il est très utile de réutiliser les données de la base de connaissances et de transférer de nouvelles connaissances en utilisant les relations et les axiomes construits dans l'ontologie. L'ontologie DFM aide les concepteurs à prendre des décisions en tenant compte de critères techniques et économiques complexes. Mais cette ontologie ne traite pas les problèmes de cycle de vie tels que la maintenabilité, les facteurs humains et les coûts.

Dans le contexte de la résolution des problèmes, le développement et la maintenance de systèmes de production collaboratifs distribués sont confrontés à des défis croissants causés par les difficultés techniques et technologiques liées à l'informatique distribuée, à l'intégration de ressources et au partage de connaissances sur des plates-formes informatiques hétérogènes. (Yin et al 2010) ont proposé une

grille sémantique basée sur des agents destinés à prendre en charge la fabrication collaborative dans des entreprises virtuelles. Aussi une architecture multi-agents permettant d'ajouter l'autonomie, l'adaptabilité et le contrôle d'intelligence aux services de la grille, l'approche proposée combine les avantages de ces techniques émergentes pour traiter le processus complexe de résolution de problèmes de la fabrication distribuée collaborative dans des environnements dynamiques et ouverts. Mais ne traite pas l'intervention de connaissances tacites pour l'intervention.

Dans le Cadre d'interopérabilité de la fabrication sémantique, des problèmes liés au partage des connaissances dans la conception et la fabrication du produit, (Chungoora et Young 2011) ont proposé un système d'interopérabilité de la fabrication sémantique appelée SMIF (Semantic Manufacturing Interoperability Framework) pour faciliter l'interopérabilité dans la conception du produit et des domaines de production. L'ontologie proposée a permis de saisir la sémantique et de partager les connaissances entre les domaines de la conception et de la production. Cependant, SMIF était limité pour prendre en charge la capture de la sémantique et assurer l'interopérabilité entre conception et production de produits pour les caractéristiques de trous simples uniquement. Cette ontologie développée est limitée en termes pour soutenir le développement de la conception et de la production de produits spécifiques à une application ontologies. Elle a besoin des concepts des niveaux plus détaillés.

Une approche de modélisation des connaissances constructives pour développer systématiquement des ontologies manufacturières a été proposée par (Lin et al. 2011), cette approche a deux outils de modélisation, un paradigme de génie logiciel et un autre à base de Web sémantique. La modélisation d'objet à base d'UML / OCL ( Unified Modeling Language/Object Constraint Language) est d'abord utilisée pour servir de base graphique et structurée à la communication conceptuelle entre experts du domaine et ingénieurs de la connaissance. Le langage OWL / SWRL permet la modélisation des ontologies basées sur les bases, étend ensuite les modèles d'objet basés sur UML / OCL avec la sémantique à l'aide d'une méthode d'acquisition de connaissances progressive et orientée sémantique. Les principaux concepts utilisés sont : '*Manufacturing\_enterprise*', '*Manufacturing\_process*', '*Manufacturing\_equi*

ment”, “Manufacturing\_operation”, “Electronic\_connector” et “Manufacturing\_feature”. Mais il y a un manque des règles d’experts d’inférence du domaine.

Dans le contexte de la formalisation des connaissances pour l’alignement. Une ontologie de référence pour la chaîne d’approvisionnement appelée SCOR (Supply Chain Ontology Reference) a été proposée par (Sakka et al., 2011). Cette ontologie est basée sur les connaissances contenues dans un modèle de référence bien connu. Ce travail est une base pour les études d’alignement sémantique avec les approches d’alignement nécessitant des connaissances commerciales élicitation et cette connaissance peut être contenue dans le modèle d’entreprise, par exemple le modèle SCOR. Mais cette ontologie est spécifique seulement pour la tâche d’approvisionnement et elle ne traite pas l’interopérabilité entre autre tâche.

Une ontologie de la tâche de la maintenance industrielle appelé IMAMO (Industrial MAintenance Management Ontology a été développée par ( Mohamed Hedi Karray et al., 2012), cette ontologie est une plates-forme de maintenance intégrée d’interopérabilité sémantique et de partage des connaissances visant à fournir aux acteurs de la maintenance les informations appropriées exploitables au bon moment. Une approche d’ingénierie en ontologie qui répond à ces deux exigences. En raison de l’absence d’une ontologie de domaine couvrant tous les aspects de la maintenance industrielle, malgré la présence de certaines normes sur le terrain et d’ontologies spécifiques différentes. Les principales classes sont 30 concepts (équipement, acteur, période, activité, processus, ressource, lubrifiant...etc.). Mais cette étude ne traite pas la résolution des problèmes liés à cette tâche et aussi à l’interopérabilité avec d’autres tâches.

Une nouvelle ontologie basée sur le concept de produit comme élément pivot est appelée ONTO-PDM (Ontology Product Data Management systems) a été proposée par (Panetto et al., 2012), une approche pour faciliter l’interopérabilité des systèmes dans un environnement de fabrication. Elle est basée sur un modèle ontologique d’un produit peut être considéré comme un facilitateur pour l’interopérabilité de tous les logiciels d’application qui partagent des informations au cours du cycle de vie d’un produit physique. Le nombre d’applications impliquées dans les entreprises manufacturières peut en fait se référer à la connaissance qui doit y

être intégré, stockant de manière appropriée toutes ses données techniques sur la base d'un modèle commun, mais il n'existe pas une relation entre la production et la standardisation des produits.

Évaluation de données multi capteurs par une approche statistique et sémantique pour surveillance de l'état des fluides dans les systèmes éoliens proposées par (Gunel et al., 2013). Cette approche permettrait d'utiliser l'expérience opérationnelle dans des réseaux pour améliorer les performances et la fiabilité des systèmes de surveillance intelligente. Cette approche peut être appliquée à d'autres domaines aussi bien dans les énergies renouvelables mais ne traite pas la résolution des problèmes du processus complet.

Une ontologie de référence manufacturières appelé MRO (Manufacturing Reference Ontology) a été proposée par (Usman et al., 2013), contribuant au développement d'une ontologie de référence formelle pour la fabrication, qui est considéré comme un élément clé dans les futurs systèmes de fabrication interopérables. Un ensemble de base des concepts de fabrication sont identifiés et leur sémantique a été capturée dans une logique formelle basée sur l'exploitation et l'extension des définitions des normes existantes, Il a été démontré que la MRO avait la capacité de prendre en charge le développement d'ontologies sémantiquement spécifiques à l'application pour la conception et la production de produits, mais ne traite pas la résolution des problèmes des tâches.

Une ontologie de machine de processus appelée MOP (Machine Of Process ) a été développée par (Ramos et al, 2014) une méthode qui intègre la réutilisation d'ontologies avec la validation et l'apprentissage de l'ontologie sont présentées. Cette approche diffère d'autres études dans le domaine de la sémantique. Cette approche permet la réutilisation des ontologies existantes comme MASON (Leimagnan et al. 2006) et MTM (Kjellberg et al 2009). Les principaux concepts de MOP sont : “*Machine*”, “*System*”, “*Interface*”, “*fonction*”, “*Features*”, “*Machine\_Element*”.

Cette ontologie est spécifique à la tâche de la conception assistée par ordinateur (CAO) et les machines de fabrication.



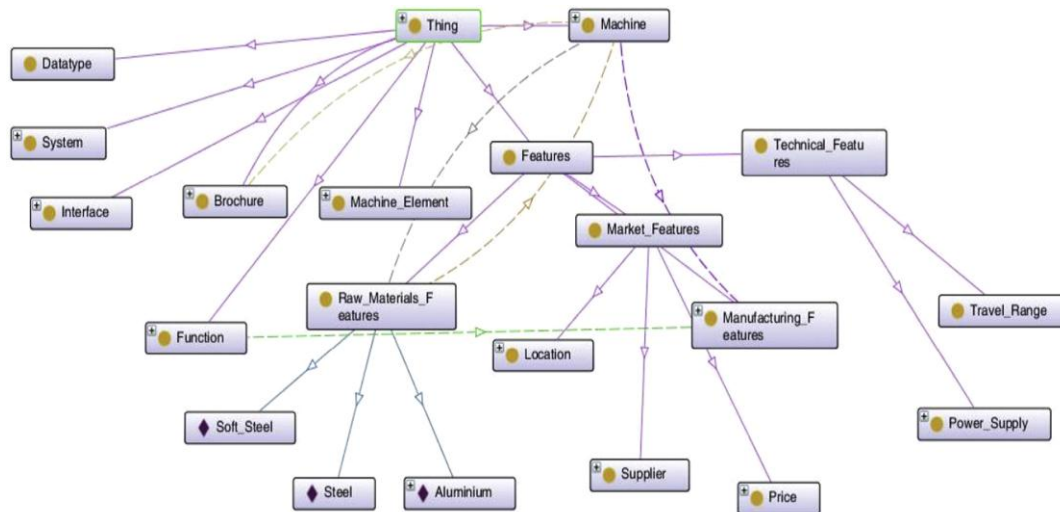


Figure 2.5: Les concepts principaux de modèle MOP adapté de (Ramos et al, 2014)

Une nouvelle approche pour développer une ontologie de processus de fabrication appelée PMPO (Part-focused Manufacturing Process Ontology) a été proposée par (Mesmer and Olewnik 2015) elle s'appuie fortement sur les caractéristiques et les qualités des pièces pour associer un produit à un processus de fabrication donné plutôt que sur des approches plus traditionnelles associant des ressources de fabrication à un processus de fabrication donné. PMPO est capable de permettre aux utilisateurs ayant une connaissance limitée des processus de fabrication de découvrir les fabricants potentiels de leurs produits. Les concepts de PMPO sont dérivés de nombreuses ontologies existantes, l'ontologie contient certaines données du fabricant et ses fonctionnalités sont illustrées à l'aide de divers exemples.

Dans le contexte de partage et de capture des connaissances pour une tâche de planification du processus d'assemblage, une ontologie appelée ARO (Assembly Reference Ontology) a été développée par (Imran and Young., 2015). Cette ontologie pour soutenir la capture et le partage de la connaissance d'assemblage en utilisant des ontologies formelles. Les principaux concepts de ARO sont : “*Assembly feature*”, “*Quantity*”, “*Tolerance*”, “*Tolerance type*”, “*Assembly fits*”, “*Assembly method*”, “*Assembly resource*” et “*Manufacturing facility*”.

Cette ontologie ne traite pas la capture des connaissances et le partage entre les autres tâches, telles que la production, le désassemblage, la qualité.



Une ontologie du domaine manufacturier intègre la tâche de production et la tâche de maintenance appelé **CDM-Core** (CREMA Data Model-Core) a été proposée par (Mazzola et al., 2016), pour améliorer la résolution des problèmes liés au connaissances des tâches dans les systèmes industriels. L'un des objectifs était d'équilibrer la (ré) applicabilité générale et la spécificité des cas d'utilisation, à savoir la production d'échappement automobile et la maintenance de la presse métallique.

## **2.6 Conclusion**

Dans ce chapitre, nous avons présenté un état de l'art sur le partage des connaissances manufacturières par les ontologies et la raison pour laquelle un système industriel a besoin d'ontologies.

A travers ce chapitre, on s'est concentré sur (1) Le partage des connaissances dans l'entreprise et la notion d'interopérabilité, (2) la Classification des ontologies, (3) le développement des ontologies, (4) les Ontologies et le raisonnement, (5) les outils de l'ontologie, (5) la revue des ontologies existantes dans le domaine de la fabrication et les modèles de la gestion des connaissances développés à base des ontologies et, (6) les problèmes et les opportunités de support des connaissances partagées dans le domaine industriel.

Les outils actuels basés sur les TIC ne permettent pas de partager efficacement les connaissances et l'interopérabilité. C'est un problème courant qui entraîne des coûts énormes pour les organisations modernes. De plus, les ontologies sont utilisées pour faciliter l'interopérabilité sémantique entre humains et entre humains et ordinateurs.

**Diverses méthodologies** de développement d'ontologies ont été abordées dans ce chapitre. Toutefois, les méthodologies développées par **Uschold et King (1995)** et **Noy et McGuinness (2001)** semblent être des candidats potentiels pour fournir une méthode permettant de développer une ontologie du domaine.

**Différents langages** de développement d'ontologies ont été explorés, mais le formalisme ontologique OWL est le plus compétent pour la définition des concepts de domaine manufacturier en raison de sa puissante interopérabilité et d'adaptation.

**Parmi les divers outils** de développement d'ontologies étudiés, Protégé est l'outil le plus utilisé dans le développement des ontologies. Il est communément utilisé en raison de son environnement de développement. Il peut exporter dans une variété de formats tels que RDF, RDFS, OWL et schéma XML.

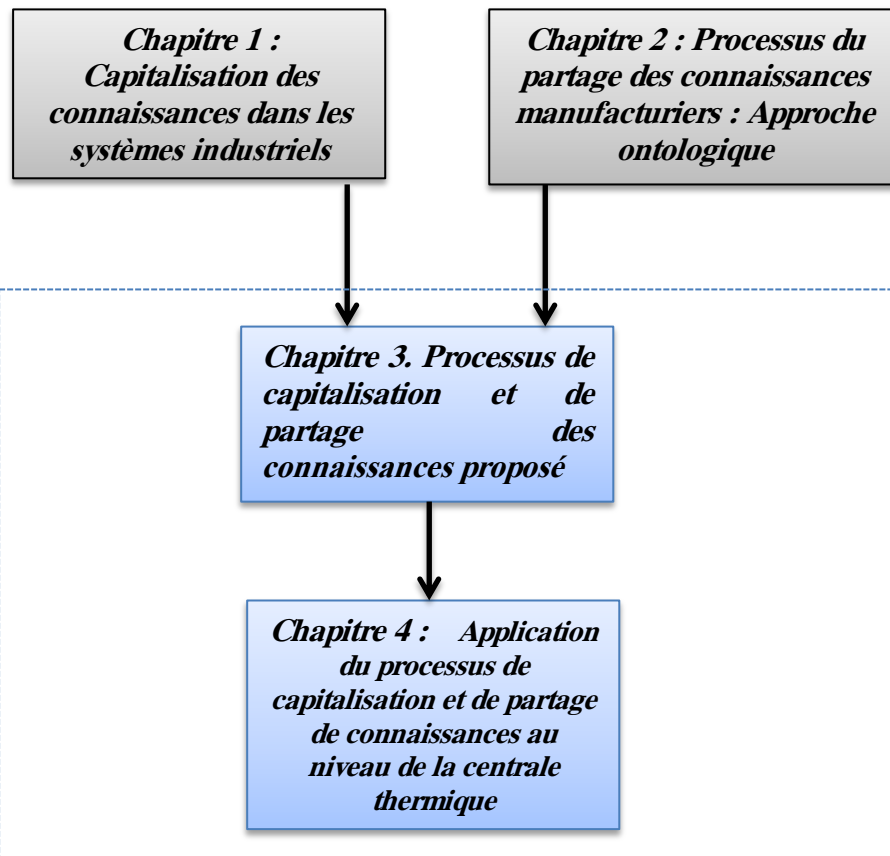
Les fonctions portent plusieurs points de vue qui peuvent résister aux efforts de partage des connaissances, avec les principales limites dans les travaux de recherches présentées dans ce chapitre.

Celles-ci sont énumérées ci-dessous.

- L'ontologie et l'approche proposée dans MASON ne fournissant pas un système de partage des connaissances durant l'exécution de la tâche. Donc, c'est un modèle applicable à une tâche spécifique.
- Quelques ontologies sont spécifiques au produit et ne sont pas de tâche de fabrication. Elles manquent aussi de mécanisme de raisonnement basé sur les règles de l'expert pour soutenir de meilleures capacités de raisonnement par rapport aux connaissances de fabrication.
- Un problème d'interopérabilité se situe entre les groupes et le système de connaissance, car les connaissances relatives aux meilleures pratiques devraient idéalement être accessibles à partir de sources multiples d'experts.
- Manque de connaissances tacites pour l'intervention.
- Manque de la partie caractéristique manufacturière
- Manque d'une terminologie commune entre les experts pour faciliter l'interopérabilité entre les tâches manufacturières telles que : la production, la maintenance, la logistique, la qualité...etc.
- Elle ne traite pas la résolution des problèmes des tâches.
- Elle ne traite pas la capture des connaissances.

Le chapitre suivant propose un processus du partage des connaissances basé sur l'Ontologie des Tâches Manufacturières pour traiter ces limites dans la recherche.

## *Deuxième Partie : Proposition et mise en œuvre d'un modèle de gestion des connaissances au système industriel*



## ***Chapitre 3 : Processus de capitalisation et de partage des connaissances proposées***

### Sommaire du chapitre 3 :

3.1 Introduction.....	64
3.2 Processus de capitalisation des connaissances : Externalisation et Modélisation.....	65
3.2.1 Positionnement vis-à-vis de l'état de l'art.....	65
3.2.2 Conception ET architecture de KCMMT.....	65
3.2.2.1 Phase d'externalisation des connaissances.....	65
3.2.2.2 Phase de modélisation des connaissances.....	74
3.2.3 Synthèse de modèle proposé.....	77
3.3 Processus de partage des connaissances basées sur l'Ontologie des Taches Manufacturières.....	79
3.3.1 Exigences d'une ontologie de tache manufacturière.....	79
3.3.2 Méthodologie de développement de l'ontologie MATO.....	80
3.3.3 Définition des concepts principaux de l'ontologie MATO et leurs relations.....	91
3.3.4 Formalisation de l'ontologie MATO.....	97
3.3.5 Interrogation de l'ontologie de MATO par les requêtes SQWRL.....	102
3.3.6 Synthèse de modèle proposé.....	104
3.4 Conclusion.....	104

### **3.1 Introduction**

Dans ce chapitre, nous présentons notre proposition de modèle de gestion des connaissances aux entreprises industrielles, ce modèle contient deux processus de gestion, le premier est de capitalisation des connaissances basés sur une tâche industrielle, dont l'objectif est d'améliorer la performance du système de production et capitaliser le savoir-faire de l'entreprise. Le deuxième processus sur le partage des connaissances basé sur l'Ontologie des Tâches Manufacturières.

Pour le premier processus concernant la capitalisation des connaissances, nous avons étudié deux phases importantes ; l'externalisation et la modélisation des connaissances, ce processus est spécifique à une tâche industrielle.

Dans la phase d'externalisation, nous avons utilisé une combinaison de deux méthodologies : une méthodologie d'ingénierie des connaissances CommonKADS et une méthodologie d'externalisation des connaissances MACTAK. Dans la phase de modélisation, nous avons utilisé deux différentes techniques de modélisation, une modélisation basée sur l'expert et la deuxième une représentation ontologique.

Pour le deuxième processus concernant le partage des connaissances, nous avons développé un processus de partage basé sur l'Ontologie des Tâches Manufacturières, nous avons développé une ontologie de tâche manufacturière MATO (MANufacturing Task Ontology) en identifiant un ensemble des concepts de fabrication et leurs relations. Cette ontologie proposée facilite le partage des connaissances entre les tâches de fabrication et aide à capturer et à réutiliser les connaissances durant l'exécution des tâches.

Dans la section 2, nous avons proposé un processus de capitalisation des connaissances, nous discuterons d'abord en section 2.1, notre positionnement vis-à-vis de l'état de l'art (chapitre 1). La Conception et l'architecture de modèle proposé est présenté dans la section 2.2. Dans la section 3, nous proposons un processus de partage des connaissances basé sur l'Ontologie des Tâches Manufacturières, cette section traite l'exigence d'une ontologie de tâche manufacturière, puis la méthodologie de développement de l'ontologie MATO. Il est nécessaire aussi de définir tous les concepts principaux de MATO, les relations entre les concepts et la

formalisation de MATO. Nous présentons l'interrogation de l'ontologie de MATO par les requêtes SQWRL.

## **3.2. Processus de capitalisation des connaissances : Externalisation et Modélisation**

### **3.2.1. Positionnement vis-à-vis de l'état de l'art**

Nous avons utilisé la méthodologie CommonKADS (Schreiber et al, 1999), un support de développement du système à base de connaissance en Europe. Cette méthodologie est basée sur six modèles, le modèle d'organisation, modèle d'agent, modèle de la tâche, modèle de la communication, modèle d'expertise et modèle d'implémentation.

L'acquisition des connaissances à travers des interviews directs avec les experts a rendu le processus d'externalisation des connaissances très difficile (Titah et al, 2014), nous avons utilisé la méthodologie MACTAK (Rosario et al, 2015) comme une technique statistique d'externalisation des connaissances (voire la section 3.2.1).

La phase de la modélisation des connaissances par la méthodologie CommonKADS est Indépendante de la phase d'implémentation, mais le problème existant est le langage de modélisation CML développé pour cette méthodologie qui est faible et semi formel. Pour lever cet enjeu, nous avons utilisé les réseaux sémantiques, les modèles logiques et les ontologies comme outil de modélisation des connaissances (la section 3.2.2).

L'objectif de notre modèle est de mémoriser la connaissance explicite de l'entreprise d'une façon cognitive, utilisant les deux modèles : le modèle logique par les règles d'experts et la modélisation sémantique par une représentation ontologique.

### **3.2.2. Conception et architecture de KCMMT**

Le modèle proposé appelé KCMMT (**A knowledge Capitalization Model for Manufacturing Task**) est composé de trois phases : 1) la phase d'externalisation des connaissances, 2) la phase de la modélisation des connaissances, 3) la phase d'implémentation et la réutilisation des connaissances. La conception du modèle KCMMT est présentée dans la figure 3.1

#### **3.2.2.1 Phase d'externalisation des connaissances**

La phase d'externalisation des connaissances combine deux méthodologies, la première est la méthodologie CommonKADS comme une technique d'acquisition des connaissances et la deuxième la méthodologie MACTAK comme technique d'élicitation des connaissances.

Pour l'identification de la tâche, nous appliquons la méthodologie CommonKADS pour développer les modèles d'organisation, de la tâche et d'agent. Voir la figure 3.2.

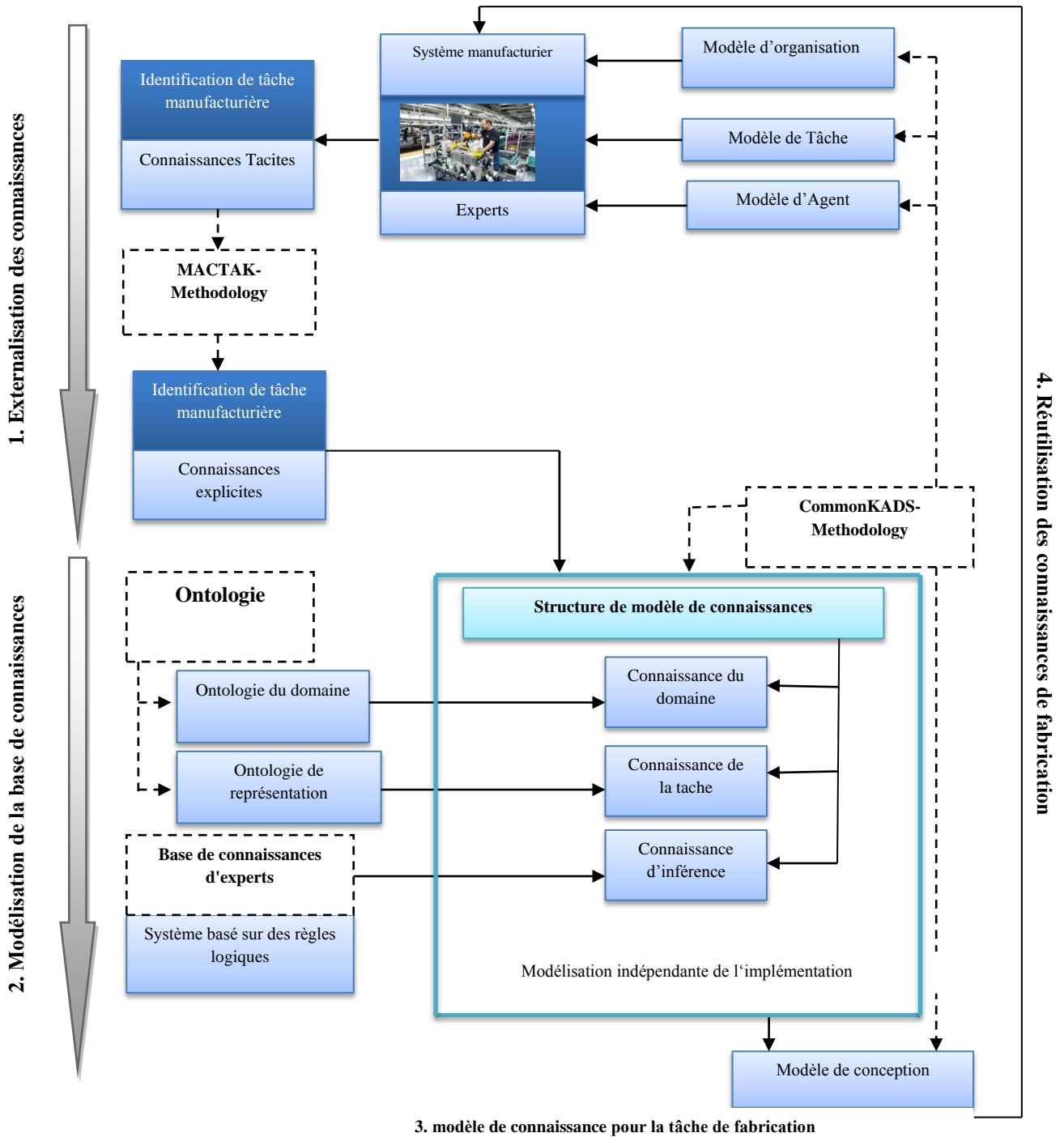


Figure.3.1. Le modèle de la connaissance proposée sur une tâche industrielle (Titah et al., 2017)

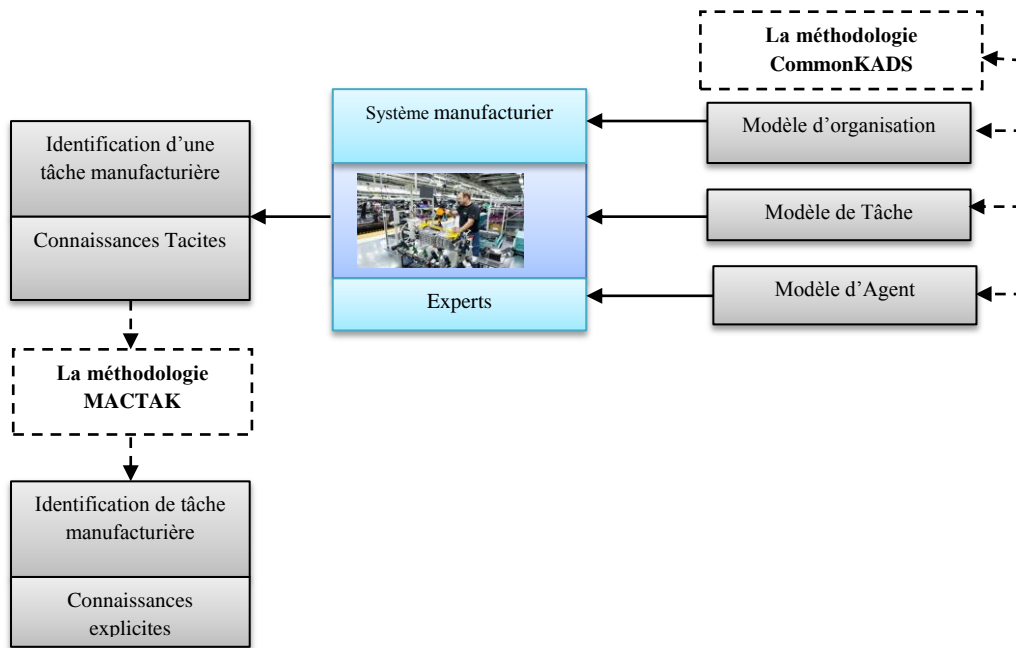


Figure.3.2. Le modèle d'externalisation des connaissances CommonKADS-MACTAK

Pour l'identification de la tâche, nous appliquons la méthodologie CommonKADS pour développer les modèles d'organisation, de la tâche et d'agent. Les éléments de sorties des modèles d'organisation, de tâche et d'agent, de la méthodologie CommonKADS est l'identification d'une tâche spécifique (voire la figure 3.3), aussi détecter les meilleurs experts et exécuteurs de tâche. Ces résultats deviennent les entrées de la méthodologie MACTAK.

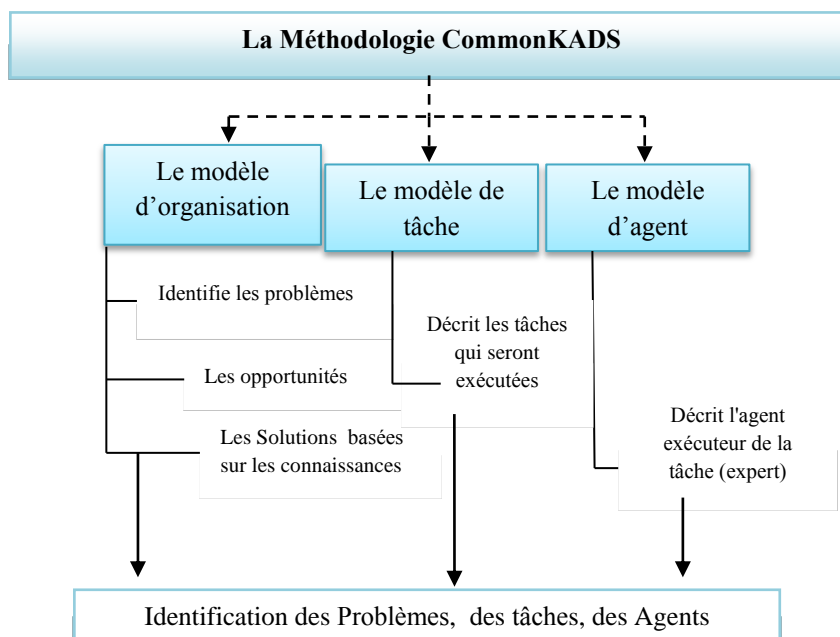


Figure.3.3 l'identification de la tâche



Avant de démarrer le processus d'externalisation, nous avons : un modèle de tâche qui définit la spécification des connaissances utilisées pour une tâche, ainsi que les goulots d'étranglement et les points d'amélioration, et un modèle d'agent qui indique comment l'agent se positionne dans l'organisation telle que définie par le modèle de l'organisation, incluant le type (humain, système d'information), fonction, position dans la structure organisationnelle, aussi une liste de responsabilités que l'agent a dans l'exécution des tâches et les restrictions. Les contraintes peuvent être au niveau de la limite de son autorité.

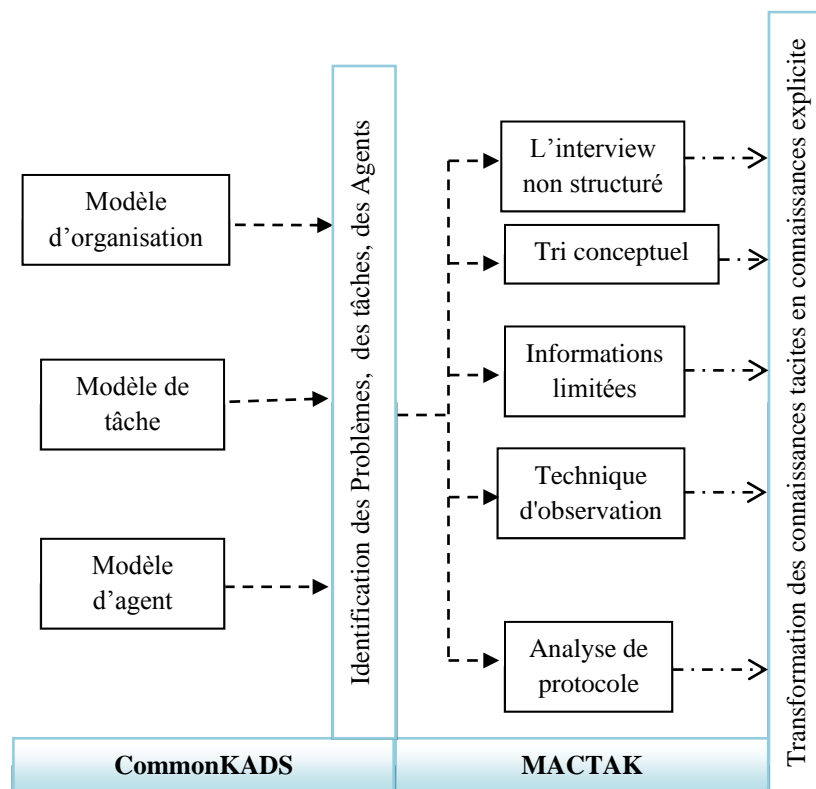


Figure.3.4 Combinaison CommonKADS-MACTAK

Dans cette phase, l'application de deux méthodologies, une méthodologie de développement des systèmes à base de connaissance CommonKADS et une méthodologie d'élicitation des connaissances MACTAK, nous avons proposé un processus d'externalisation à base de cinq étapes successives : 1) identification des problèmes, des tâches et des agents, 2) analyse des problèmes, 3) diagnostic les problèmes analysés, 4) capturer les connaissances tacites, 5) Modèle explicite sous forme de règles de production, comme le montre la figure 3.5.

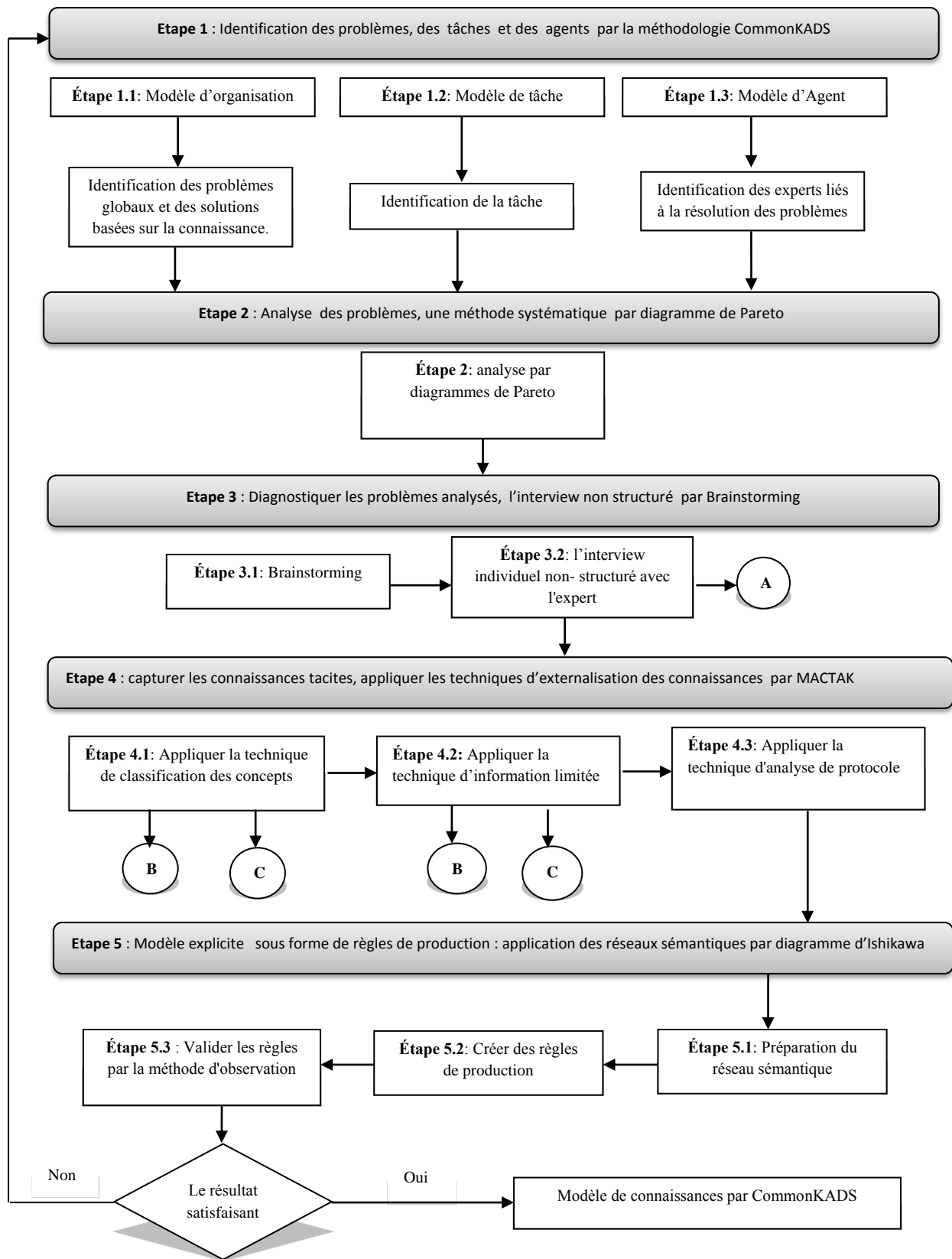


Figure.3.5 La phase d'externalisation des connaissances du modèle KCMMT

### 3.2.2.1.1 Étape 1 : Identification des problèmes, des tâches et des agents

Étape 1.1 Le modèle d'organisation : Le modèle de l'organisation pour identifier les problèmes et les possibilités de solutions basées sur les systèmes de connaissances. Il décrit la structure de l'organisation, les processus, le personnel et les ressources. Il aide à identifier les systèmes de connaissances et la valeur ajoutée à l'organisation. Ce modèle est construit à l'aide de quatre tableaux reliés au Modèle de l'Organisation: MO-1, MO-2, MO-3, et MO-4 décrits à la figure 3.6 et détaillés dans les tableaux I à IV de l'annexe A.

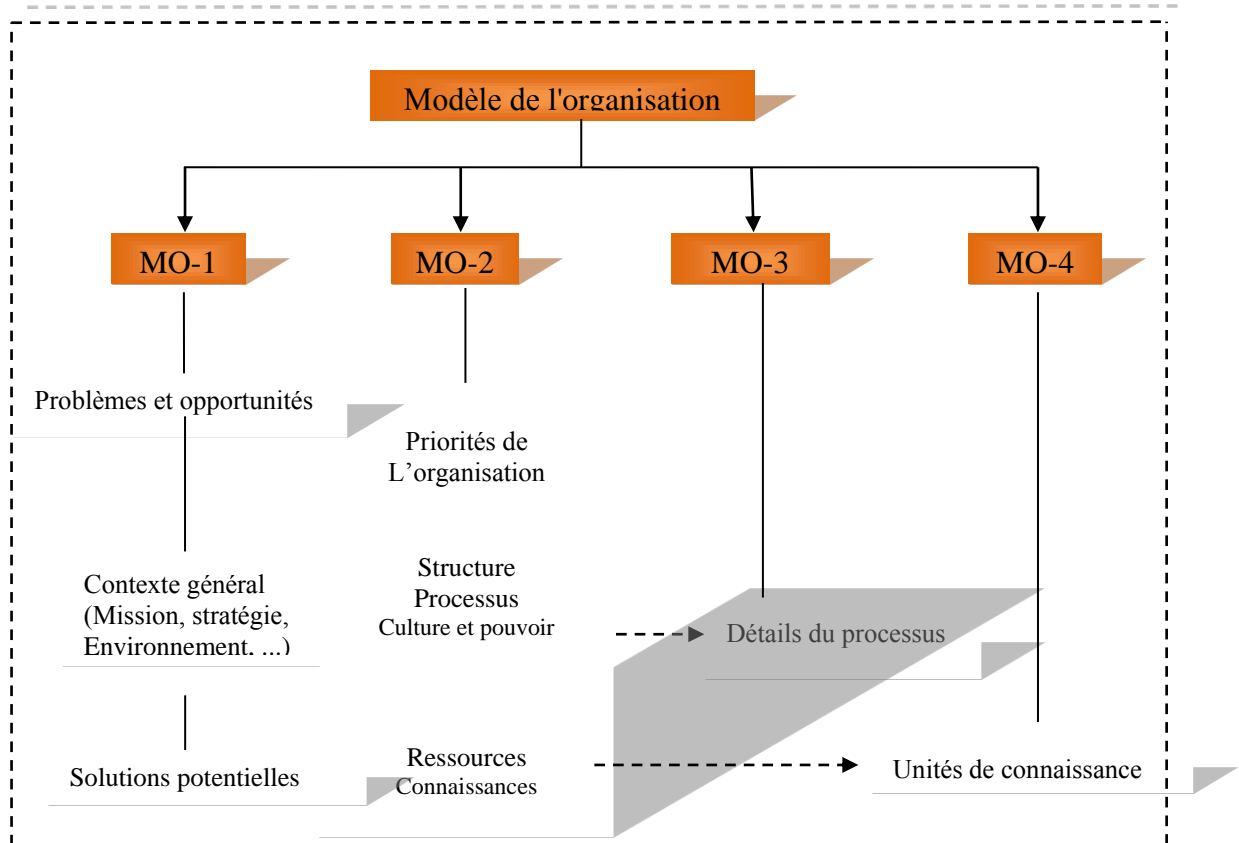


Figure 3.6 Le modèle de l'organisation

Étape 1.2 Le modèle de tâches : Le modèle de tâches permet d'étudier les sous-processus d'entreprise. Il analyse l'organisation globale des tâches, les entrées et les sorties, les pré-conditions et les critères de performance, ainsi que les ressources et compétences nécessaires. Dans CommonKADS, une tâche est un sous-processus du processus d'entreprise qui:

- représente une activité orientée vers un but et qui ajoute de la valeur à l'organisation;
- traite des entrées et fournit des sorties d'une manière structurée et contrôlée;
- utilise des ressources;
- a besoin de connaissances et en fournit ;

Le modèle de tâches peut être vu comme un raffinement du tableau MO-3. Ce raffinement est effectué à l'aide du tableau Modèle de Tâches-1 (MT-1). Le détail de chaque élément de

connaissance associé à une tâche est obtenu au tableau MT-2 (tableaux de tâche de l'annexe A)

*Étape 1.3 Le modèle d'agents* : Après la construction du modèle de tâches, le modèle d'agents décrit les caractéristiques des agents qui peuvent être soit des humains, soit des systèmes d'information, soit d'autres entités capables d'effectuer une tâche.

Les agents sont les exécuteurs des tâches. Le modèle décrit leurs compétences, leur autorité et leurs contraintes ainsi que les liens de communication. Ce modèle est construit à l'aide du tableau Modèle de l'agent-1 (MA-1) (tableau dans l'annexe A).

### **3.2.2.1.2 Étape 2 : Analyse des problèmes**

Le modèle de l'organisation identifie les problèmes d'une façon générale. Il nécessite aussi une connaissance détaillée sur le processus. En d'autres termes, nous avons besoin d'un modèle qui permet de le connaître, de le comprendre et de l'interpréter, de détailler l'analyse d'un processus afin de parvenir à une compréhension efficace du problème. Pour analyser les problèmes dans une situation réelle, le diagramme de Pareto est utilisé lorsqu'il est nécessaire d'analyser les problèmes de manière systématique, ainsi que lorsqu'il existe un grand nombre de problèmes et des ressources limitées pour les résoudre. Le graphique, correctement conçu, indique les zones les plus problématiques suivant un ordre de priorité. De telles priorités peuvent se reproduire de 70% à 80% du problème dans son ensemble.

### **3.2.2.1.3 Étape 3 : Diagnostic des problèmes analysés**

*Étape 3.1 Brainstorming* : D'après la construction du modèle d'organisation, de tâche et d'agent, une séance de brainstorming avec les experts définit le modèle d'agent, qui est un outil qui aide à la découverte des causes des problèmes, en tenant compte de la connaissance des experts sur les problèmes cités dans le modèle d'organisation. Il vise à générer des idées / suggestions avec un groupe des agents industriels exécuteurs des tâches (ingénieur, techniciens, opérateurs....) susceptibles de fournir des avancées significatives dans la résolution de problèmes.

*Étape 3.2 l'interview individuelle non-structuré* : Les entretiens non structurés ont peu de restrictions, ce qui signifie qu'il n'y a pas d'ordre ou de séquence de questions prédéfinies ou de direction de conversation.

Un entretien individuel non structuré après une séance de Brainstorming, En raison des limites de la technique de l'entretien non structuré, Son but n'est pas d'acquérir des

connaissances spécifiques, mais plutôt d'extraire de l'expert son large panorama sur le sujet en question; c'est pourquoi il est si courant de commencer l'entretien en demandant «Comment pouvez-vous résoudre ce problème?»,

#### **3.2.2.1.4 Étape 4 : capturer les connaissances tacites**

##### Étape 4.1: Appliquer la technique de classification des concepts

La méthode de classification des concepts ou «card sorting» vise à identifier et organiser les termes ou concepts et leurs relations dans un domaine particulier, comme la classification des objets, les hiérarchies, les relations entre les concepts et d'autres descriptions statiques des objets du domaine, selon le point de vue de l'expert.

Une liste de termes de domaine est obtenue à partir des entretiens non structurés initiaux. Des cartes avec le nom des objets sont présentées à l'expert, qui est chargé de trier les cartes qui sont similaires ou du même type, afin d'obtenir des catégories de concepts. Les cartes sont mélangées une fois de plus, et l'expert doit les regrouper à nouveau en essayant d'autres critères afin que de nouvelles catégories soient créées. L'expert supprimera certains objets lors de la classification - car ils sont synonymes ou non pertinents - et inclura d'autres concepts, qui avaient été laissés de côté. Ce processus est répété plusieurs fois, mais il peut avoir quelques variations. Le résultat de l'application de la classification des concepts est un groupe de composants qui partagent des attributs communs. Cette technique facilite la tâche de communication avec les experts du domaine.

##### Étape 4.2: Appliquer la technique d'information limitée

Cette étape consiste à présenter un problème réel à l'expert, en s'attendant à ce qu'il propose une méthode pour sa solution. Il est possible d'adopter une approche qui restreint progressivement la disponibilité des informations; l'expert révèle ainsi le niveau minimum d'informations dont il a besoin pour prendre une décision. L'expert peut également être explicitement chargé d'adopter une stratégie particulière. Une expérience assez courante consiste à montrer plusieurs photographies liées à des sujets avec lesquels l'expert a l'habitude de travailler, mais avec une restriction de temps beaucoup plus importante. Après cela, il lui est demandé d'extérioriser tout ce dont il se souvient de ces images et de fournir leur interprétation.

Au cours des entretiens non structurés, certaines questions étaient basées sur des fictifs pour le moment des problèmes d'entretien, mais déjà vécues au quotidien dans la pratique. Les experts ont été amenés à répondre à chacune des raisons lors de l'enquête sur les causes en utilisant les concepts assignés par les experts du domaine. Des photos de défauts ont également été utilisées à cette époque, la visualisation des images était limitée à 10 s chacune.

Étape 4.3: Appliquer la technique d'analyse de protocole

Les entretiens ayant été menés individuellement, il était nécessaire pour effectuer une analyse des informations collectées. L'analyse consistait à vérifier l'existence d'une similarité ou d'un schéma de réponses entre les réponses des experts.

Une analyse de contenu d'une manière systématique et objective est nécessaire.

Il est possible de classer les phrases ou leurs parties selon le type de connaissances qu'elles témoignent: concepts, règles, procédures, autres. Une analyse d'interaction entre l'intervieweur et l'expert pour identifier les catégories de phrases. Dans ce cas, les catégories créées sont orientées vers la conversation, par exemple: négociation, orientation, explication, autre. Le résultat d'un processus peut être attribué à plusieurs facteurs, et une relation de cause à effet peut être trouvée parmi ces facteurs. Il est possible d'utiliser une technique d'analyse de protocole pour déterminer la structure ou la relation de l'effet de cause multiple en observant systématiquement le processus. Il est difficile de résoudre des problèmes complexes sans considérer une telle structure, qui consiste en une chaîne de causes et d'effets, donc le choix du diagramme d'Ichikawa est une méthode simple et facile pour la représentation du phénomène concernant les causes possibles d'un problème.

**3.2.2.1.5 Étape 5 : Modèle explicite sous forme de règles de production**

Étape 5.1: Préparation du réseau sémantique

D'après l'analyse du protocole, la manière de représentation des connaissances collectées peut être comprise comme un moyen systématique de structurer et de codifier les connaissances par un réseau sémantique. Le réseau sémantique est constitué d'un ensemble de nœuds reliés par un ensemble d'arcs. En général, les nœuds représentent des objets et les arcs représentent les relations binaires

entre ces objets. Cependant, les nœuds peuvent également être utilisés pour représenter des prédicats, des classes.

Dans cette étape, le réseau sémantique a été utilisé comme moyen de représenter les connaissances tacites impliquées dans le processus, ainsi qu'un objectif du réseau sémantique est de représenter les connaissances d'une manière systématique pour mieux aider à l'élaboration des règles de production.

Étape 5.2: Créer des règles de production

Après le développement des réseaux sémantiques, la représentation des connaissances, à travers des règles de production, est très facile pour encoder les connaissances. Un système expert, basé sur des règles, se caractérise par: l'intégration de connaissances heuristiques dans les règles IF – THEN; combiner des règles simples pour résoudre des problèmes complexes et pouvoir expliquer toute conclusion à partir du raisonnement appliqué par le système.

Étape 5.3 : Valider les règles par la méthode d'observation

Les règles de production ont été évaluées par une technique d'observation. Grâce à cette étape, l'expert vient avec la solution du problème et observe le processus. L'expert, ici, est généralement plus à l'aise, une fois qu'il est dans son environnement de travail pour effectuer ses tâches quotidiennes. L'observation peut être utilisée pour identifier et vérifier ces connaissances externalisées sous forme de concepts, de procédures, d'hypothèses ou de modèles proposés par les experts du domaine pour résoudre un problème automatique.

**3.2.2.2 Phase de modélisation des connaissances**

Il utilise deux catégories différentes de modélisation des connaissances; la combinaison entre la base de connaissances d'experts et l'ontologie. Il a un modèle logique présenté par des systèmes logiques basés sur des règles et des techniques d'ontologie par relation concept-attribut. Le détail de la modélisation des

connaissances est présenté dans la section 4.2. La connaissance de la modélisation est indépendante des outils d'implémentation.

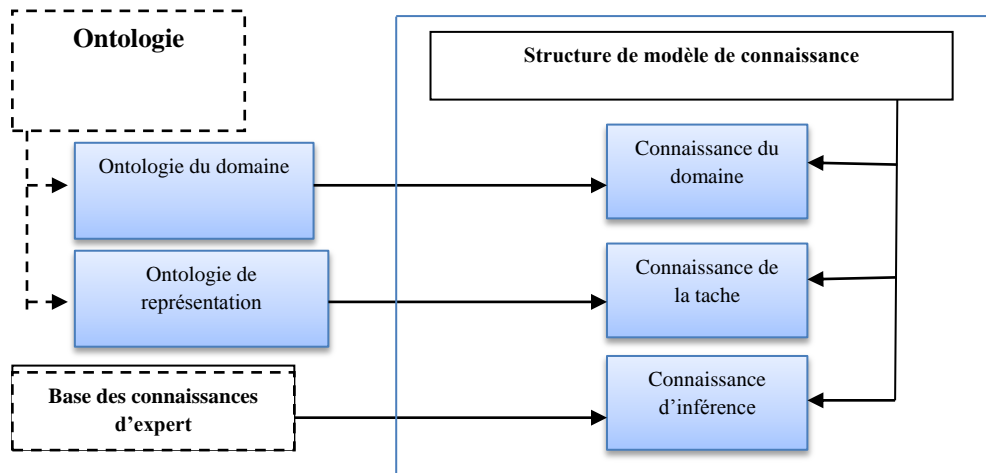


Figure.3.7 La phase de modélisation des connaissances du modèle KCMMT

Après la phase d'externalisation des connaissances tacites en connaissances explicites, la modélisation est la deuxième phase dans notre modèle proposé, il existe trois catégories dans le modèle de connaissance de la méthodologie CommonKADS (voire la figure 3.8) : la connaissance de la tâche, la connaissance de domaine et la connaissance d'inférence. Nous avons utilisé deux types de modélisation des connaissances différents (voir la figure 3.7) :

- une modélisation par une ontologie de représentation pour la connaissance de la tâche et la connaissance de domaine.
- Une modélisation logique à base de connaissances d'experts pour la connaissance d'inférence.

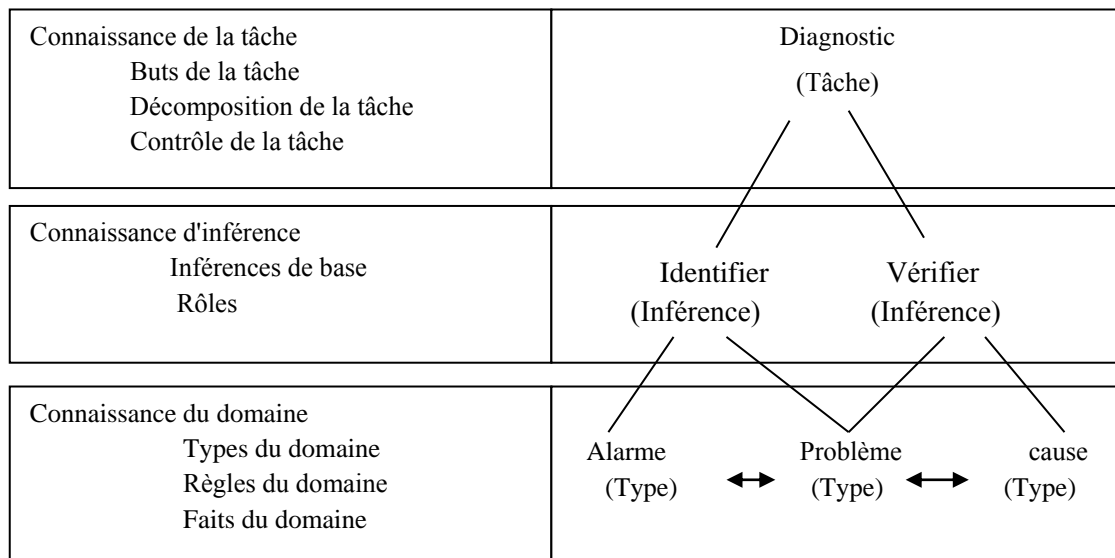


Figure 3.8. Les catégories de connaissance dans un modèle de connaissances



### **3.2.2.2.1 Connaissance du domaine**

La connaissance du domaine dans CommonKADS décrit l'information statique du domaine d'application. Elle est constituée de deux groupes, le schéma du domaine (modèle des données) et la base de connaissances (les instances des types spécifiés au schéma du domaine). Il s'agit ici d'identifier les concepts et les relations, les types-de-règle et de décrire la base de connaissances (BC) pour chaque relation et type-de-règle. La représentation ontologique de modèle de connaissances est présentée au tableau 3.1. Une représentation ontologique peut assurer la consistance et enlever l'ambiguïté dans les descriptions des connaissances pour un domaine spécifique.

### **3.2.2.2.2 Connaissance de la tâche**

La connaissance de la tâche dans le modèle d'expertise de CommonKADS décrit les buts poursuivis par une application et aussi la façon de les atteindre, elle se fait par une décomposition fonctionnelle de la tâche en sous-tâches et en inférences. L'aspect "comment" inclut une description du comportement dynamique des tâches, c-à-d, définition de leur ordre d'exécution. Par exemple la tâche du diagnostic industriel repose sur le principe de l'observation et de l'analyse des symptômes d'une défaillance. Dans ce cas, on peut la décomposer en sous tâches suivantes : identifier les causes, contrôler les paramètres et localiser la cause du problème. La représentation ontologique des connaissances de la tâche est basée sur le langage OWL (voire la figure 3.9) comme suit :

- les concepts sont des notions qui représentent la description d'une tâche ou d'une fonction.
- les propriétés ou attributs sont des restrictions des concepts ou des relations qui représentent les sous tâches, en langage OWL par (object-properties).

### **3.2.2.2.3 Connaissance d'inférence**

Elle décrit les étapes d'inférence qui seront effectuées à l'aide de la connaissance du domaine. Les inférences correspondent au plus bas niveau d'une décomposition fonctionnelle. La figure 3.10 présente les deux parties des connaissances d'inférence, la description d'inférence et le raisonnement logique de l'inférence par les règles de type If-Then. Nous présentons les détails dans le chapitre 4.

Tableau 3.1. Représentation ontologique des connaissances de domaine

Connaissances du domaine	Modélisation CommonKADS	Représentation ontologique par le langage OWL
le schéma du domaine	les concepts : Un concept décrit un ensemble d'objets ou d'instances qui existent dans le domaine d'application et qui ont des caractéristiques similaires. Il correspond à la notion de classe dans la modélisation orientée-objet, à la différence que les fonctions ne sont pas incluses dans la définition d'une classe.	Un concept : est une représentation d'un objet, une notion, une idée. Un concept peut être divisé en trois parties : un terme (est un élément lexical qui permet d'exprimer le concept en langue naturelle), une notion (également appelée intension du concept, contient la sémantique du concept, exprimée en termes de propriétés et attributs) et un ensemble d'objets (appelé extension du concept, ces objets sont appelés instances du concept.).
	les relations : Les liens entre les différents concepts sont définis par une construction de type relation. Une relation peut elle-même avoir des attributs.	Les relations : représentant des types d'interactions entre les concepts. Une relation permet de lier des instances de concepts. Elles sont caractérisées par un terme et une signature qui précise le nombre d'instances de concepts que la relation lie, leurs types et l'ordre des concepts.
	les types-de-règle : Un type-de-règle ressemble à une relation dont les arguments sont des antécédents et des conséquents. Ces arguments ne sont pas des instances de concept, mais plutôt des expressions à propos de ces concepts.	Les types de règle peuvent être représentés par des règles SWRL, une propriété OWL ou une relation intégrée. Le langage des règles fournit de nombreux ensembles de fonctions intégrées, par exemple fonctions de chaîne et fonctions mathématiques.
la base de connaissances	Une BC est typique de la modélisation des connaissances; elle contient les instances sur lesquelles le raisonnement devra s'effectuer. La séparation du schéma du domaine et de la base de connaissances implique que l'acquisition de connaissances se fait en deux étapes. La première consiste à définir les types de connaissances et la deuxième à obtenir les instances de ces connaissances. Une BC contient les instanciations des relations et de type-de-règle.	On peut représenter la base des connaissances par des instances et des axiomes, Instances (ou individus) représentent des éléments spécifiques d'un concept (ou d'une classe). Les axiomes est constitué par des assertions acceptées comme vraies qui s'appliquent sur les classes ou les instances des classes de l'ontologie.

### 3.2.3. Synthèse de modèle proposé

La principale contribution de cette étude est de sauvegarder les connaissances appliquées dans l'industrie manufacturière, en utilisant la nouvelle technique et méthodologie d'externalisation des connaissances.

Notre modèle de connaissances proposé dans ce processus contient deux processus principaux: l'externalisation des connaissances et modélisation des

connaissances. Dans le processus de connaissance de l'externalisation: nous combinons deux méthodes; la méthodologie CommonKADS, une ingénierie des connaissances pour développer KBS à l'aide de modèles spécifiques, et la méthodologie MACTAK, un nouveau MACTAK.

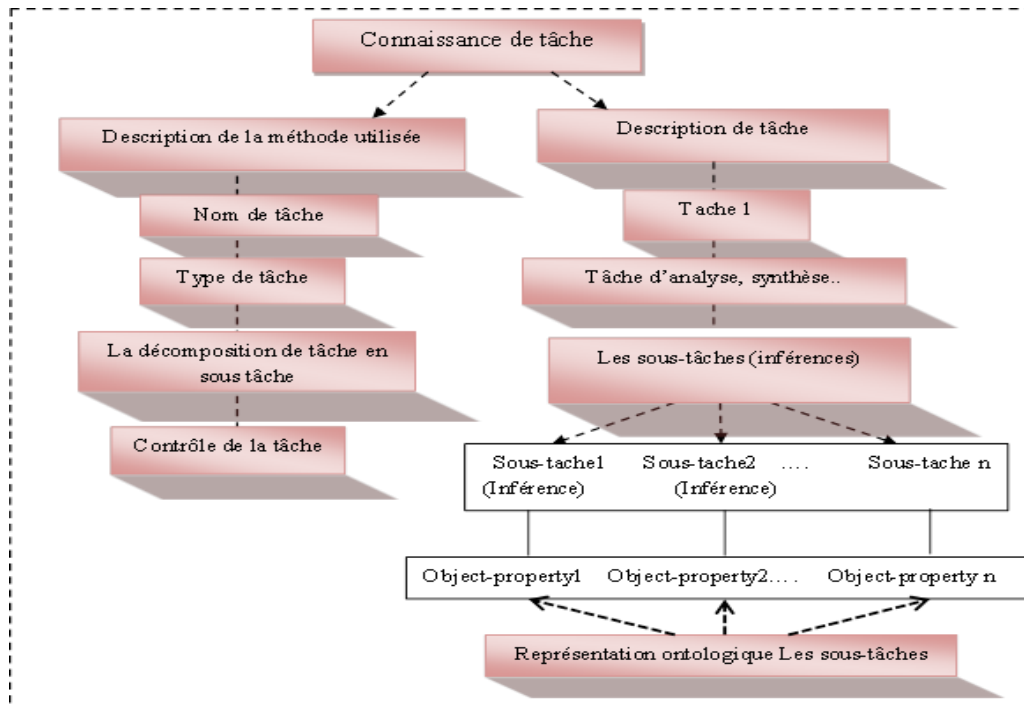


Figure 3.9. La connaissance de la tâche

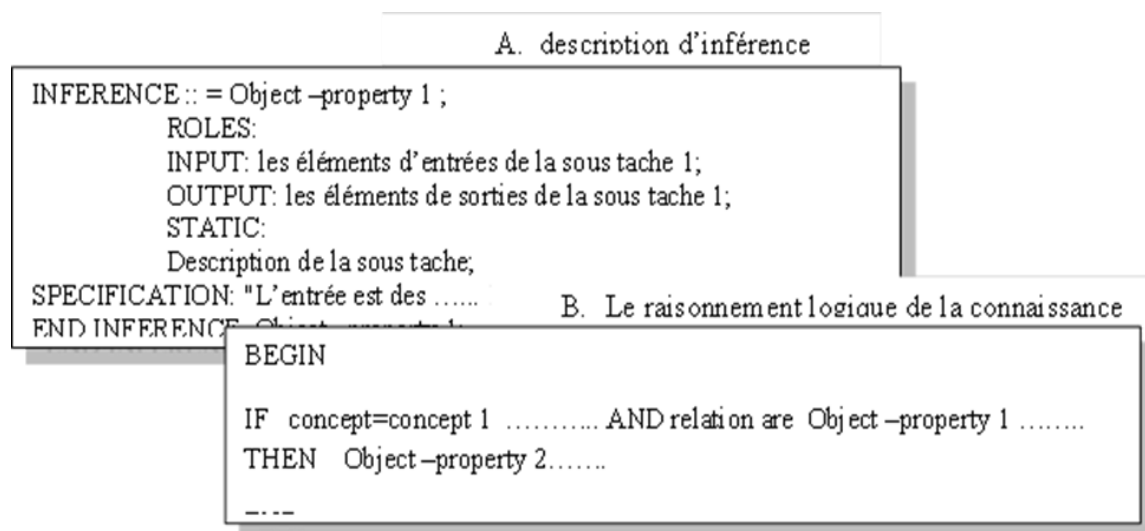


Figure 3.10. La connaissance d'inférence

Le type de connaissances extraites est tiré de données statistiques et de données génériques. Alors que CommonKADS est caractérisé par des données génériques et MACTAK par des données statistiques spécialisées,

Notre modèle facilite la capture des connaissances expertes et transforme les connaissances tacites en explicites avec une maximisation des règles de production, CommonKADS est un complément des techniques de sollicitation. Le modèle d'organisation, le modèle de tâche et le modèle d'agent sont les entrées de la méthodologie MACTAK. Dans lequel il a spécifié une tâche spécifique, et il a détecté les meilleurs experts et exécuteurs de tâche.

Ces entrées définissent l'identification des tâches de fabrication et deviennent les entrées de la méthodologie MACTAK. La combinaison facilite l'application des techniques d'élicitation des connaissances tacites.

Modélisation du processus de connaissances: Le résultat de la méthodologie MACTAK était la connaissance explicite de l'identification des tâches de fabrication, qui deviennent les entrées du modèle de modèle de connaissances de CommonKADS. Nous avons utilisé deux modèles de connaissances différents pour les modèles de modèles de connaissances, l'ontologie par les relations concept-attributs pour la connaissance des tâches et le domaine de connaissances. Base de connaissances d'experts pour les connaissances d'inférence par modèle logique, nous avons également utilisé des réseaux sémantiques pour extraire des règles de production. Ce modèle facilite la représentation des connaissances en termes d'ontologie et facilite le raisonnement en termes de modèle logique; tout cela converge vers la représentation de la connaissance comme raisonnement de l'être humain.

### **3.3 Processus de partage des connaissances basé sur l'Ontologie des Tâches Manufacturières**

#### **3.3.1 Exigences d'une ontologie de tâche manufacturière**

##### **3.3.1.1 Problème industriel lié au partage des connaissances**

Selon notre problématique industrielle, nous avons constaté les problèmes suivants :

- Manque de partages des connaissances entre experts et nouveau employeurs,
- Manque de travail collaboratif dans l'entreprise
- Absence d'un modèle d'interaction entre tâches industrielles
- Manque de coopération entre industriels
- Problème de compréhension des connaissances entre différents départements, (Par exemple production, maintenance et qualité)

Pour répondre à ces problèmes industriels, nous avons besoin d'un processus du partage des connaissances manufacturières.

### **3.3.1.2 Besoin d'une ontologie de tâche manufacturière**

L'ontologie de fabrication, est une ontologie constituée d'un ensemble de concepts de base : relations, instances, axiome et fonctions dans le domaine de fabrication, fonctionnant comme un indice de gestion de la mémoire de fabrication.

L'ontologue manufacturier est une personne chargée d'étudier, construire, utiliser et gérer l'ontologie manufacturière.

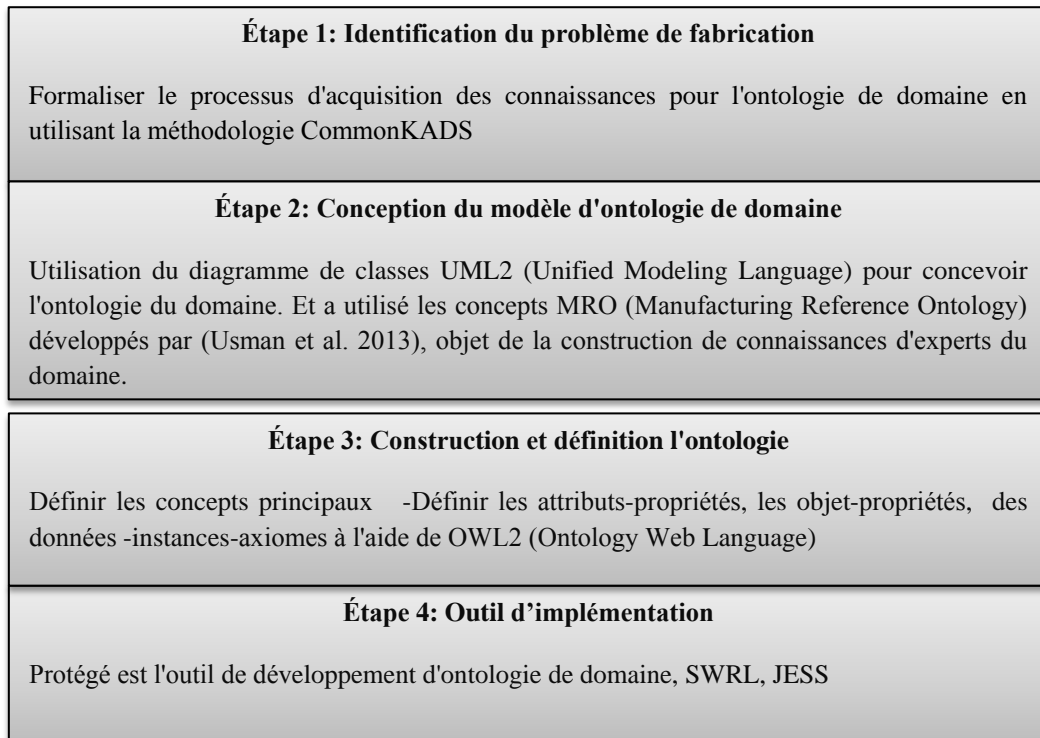
L'ontologie est le principal pilier des technologies sémantiques; l'ontologie a été un mécanisme de partage des connaissances. L'objectif de notre recherche est de développer un modèle de partage des connaissances pour l'amélioration de la résolution de problèmes dans la fabrication. Par conséquent, notre objectif dans cette étude est de faciliter l'échange de connaissances et l'interaction en ligne entre le système basé sur les connaissances, les employés et les experts à l'intérieur et à l'extérieur des entreprises. Plus précisément, nous avons proposé un modèle de partage des connaissances en utilisant l'ontologie manufacturière.

### **3.3.2 Méthodologie de développement de l'ontologie MATO**

Dans cette section, nous présentons et décrivons la création d'un processus de partage des connaissances en fabrication. Cependant, une étape constructive pour développer une ontologie de fabrication en vue de la résolution de problèmes. Généralement, cette étape permet de créer une ontologie de domaine spécifique, en particulier dans une tâche de fabrication spécifique.

Dans ce cadre, nous avons utilisé des étapes successives pour développer une ontologie de fabrication. Alors que nous avons utilisé la méthodologie CommonKADS (Schreiber, G., 2000) pour formaliser l'acquisition de connaissances dans une ontologie de domaine et décrire le besoin d'une ontologie pour la résolution de problèmes en fabrication. Ensuite la conception d'un modèle d'ontologie de domaine utilisant le langage de modélisation unifié UML de (Wei Liau et al., 2003), et le développement des composants de l'ontologie sont basés sur les directives de (Noy et McGuinness, 2000).

Protégé1 est l'outil de l'ontologie de domaine. La méthodologie de développement d'ontologies de fabrication est illustrée dans la figure 3.11.



**Figure 3.11** La méthodologie de developement de l'ontologie

### **3.3.2.1 Étape 1: Identification du problème de fabrication**

Dans la résolution de problèmes de fabrication, l'identification du problème est une étape importante pour la modélisation et la mise en œuvre d'une base de connaissances. Les problèmes d'interopérabilité dans les industries manufacturières ont été très coûteux (Usman et al., 2013). L'ontologie est une plate-forme de partage et de réutilisation des connaissances entre entreprises. Certaines techniques ont été utilisées pour l'acquisition de connaissances dans le domaine de l'ontologie. Cependant, la méthodologie CommonKADS a été utilisée pour formaliser le processus d'acquisition des connaissances (Bimba et al., 2016). Pour la construction d'un système basé sur la connaissance (Titah et al. 2014), a présenté le problème de l'outil de mise en œuvre de la méthodologie CommonKADS. Également, un problème de partage et de réutilisation des connaissances tacites dans les tâches de fabrication (Titah et al., 2017).

Dans la présente étude, la méthode d'ingénierie des connaissances CommonKADS a été utilisée pour soutenir la procédure de préparation pour deux

raisons principales. Premièrement, Elle représente la norme de référence pour l'analyse des connaissances et le développement d'un système de capitalisation de connaissances (voire section 3.2). Il se compose principalement de trois activités:

(1) Identification des connaissances. Cela implique d'identifier les problèmes dans des domaines pertinents. La tâche principale de cette activité consiste à examiner et analyser la littérature existante et les sources sémantiques.

(2) Spécification des connaissances. Au cours de cette étape, un modèle est sélectionné et un modèle semi-formel est développé. Le but de ces activités est de créer une spécification pour le modèle de connaissances.

(3) Perfectionnement des connaissances. Il s'agit de la dernière phase du processus de modélisation des connaissances, et il se compose généralement de deux tâches: validation et perfectionnement du modèle de connaissances. Deuxièmement, la méthodologie CommonKADS intègre de nombreuses activités considérées comme essentielles pour l'ingénierie des connaissances, de l'analyse et de la gestion des connaissances au développement de systèmes à base de connaissances.

Selon le modèle d'organisation de la méthodologie CommonKADS , les problèmes industriels sont:

- Manque de partage des connaissances entre experts et nouveaux employeurs;
- Absence de modèle d'interaction entre les tâches industrielles;
- Manque de coopération entre les fabricants;
- Problème d'interopérabilité entre différents services, par exemple: production, maintenance et qualité).

Le partage des connaissances dans le secteur de la fabrication nécessite un concept de fabrication et un vocabulaire précis; dans cet objet, nous avons proposé une ontologie de tâches de fabrication afin de faciliter le partage et la réutilisation des connaissances pour la résolution de problèmes de fabrication.

### 3.3.2.2 Étape 2: Conception du modèle d'ontologie de domaine

Nous avons conçu une ontologie de fabrication de domaine à l'aide du diagramme de classes UML (Unified Modeling Language) proposé par (Wei Liau et al., 2003) et nous avons utilisé les concepts MRO (Manufacturing Reference Ontology) développés par (Usman et al. 2013, Lin et al. 2011, Young et al. 2007) et (Lemaingnan et al. 2006).

Dans l'objectif de construction et de modélisation, l'expert du domaine de connaissances utilise les principales classes de notre ontologie de tâches de fabrication qui sont: le concept de processus manufacturier « *ManufacturingProcess* », le concept de tâche manufacturière « *ManufacturingTask* », le concept de problème industriel « *ManufacturingProblem* », le concept de Solution industrielle « *ManufacturingSolution* », le concept de caractéristique industriel « *ManufacturingFeature* », le concept d'agent industriel « *ManufacturingAgent* » et le concept d'équipement industriel « *ManufacturingEquipment* ».

Le diagramme de classes UML2 de la fabrication de domaine est présenté à la figure 3.12, le modèle proposé devrait contenir sept classes principales d'ontologies de fabrication: « *ProcessusProduction* », « *Tâche de fabrication* », « *Problème de fabrication* », « *Solution de fabrication* », « *FabricationCaractéristiques* » fabrication, « *Agent de fabrication* » et « *Équipement de fabrication* ». Trois relations sont présentées dans le diagramme de classes d'ontologies, les généralisations, les associations et les relations d'agrégations, la classe « *ManufacturingProcess* » étant associée à la classe « *ManufacturingTask* » via l'affect. La classe « *ManufacturingTask* » est associée à la classe « *ManufacturingFeature* » via l'association de produits. La classe « *ManufacturingAgent* » est associée aux classes « *ManufacturingTask* », « *ManufacturingProblem* » et « *ManufacturingSolution* » avec, respectivement, « *realise* », « *identifie* » et « *propose* » des associations. La classe « *ManufacturingEquipment* » s'associe à la classe « *ManufacturingProblem* » par le biais de l'association de « *causes* ». Inversement, la classe « *ManufacturingProblem* » est associée à la classe « *ManufacturingEquipment* » via l'association « *isCausedBy* ».



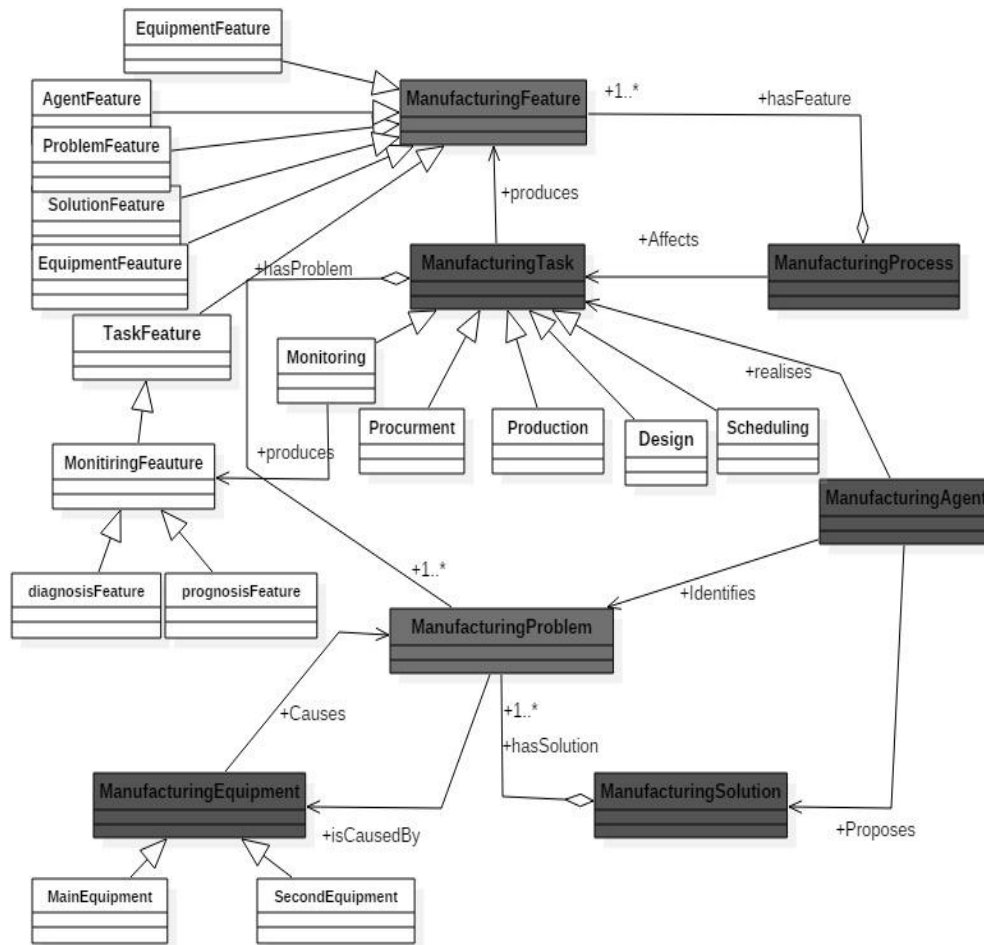


Figure 3.12 Diagramme de classes en UML2 de domaine

Les principales classes de la classe « ManufacturingTask » par le biais des relations de généralisation sont les suivantes: surveillance, approvisionnement, production, planification et conception. De plus, les principales classes de la classe d'intérêt « ManufacturingFeature » sont: « AgentFeature », « ProblemFeature », « SolutionFeature », « TaskFeature », « AgentFeature » et « EquipmentFeature ».

La classe « ManufacturingProcess » contient certaines classes « ManufacturingFeature » via l'association « hasFeature » et les relations d'agrégation. L'une des classes « ManufacturingTask » contient de nombreuses classes « ManufacturingProblem » via l'association « hasProblem » et les relations d'agrégation. La classe « ManufacturingProblem » contient une ou plusieurs classes « ManufacturingSolution » par le biais de l'association « hasSolution » et des relations d'agrégation.

### 3.3.2.3 Etape 3: construire et définir l'ontologie

Nous avons défini le développement des composants d'ontologie en se basant sur les directives de (Noy et McGuinness, 2000) avec un processus itératif:

- 1. Déterminer le domaine et la portée de l'ontologie
- 2. Énumérer les termes importants dans l'ontologie
- 3. Définir les classes et la hiérarchie des classes
- 4. Définir les propriétés des classes - slots
- 5. Définissez les facettes des slots
- 6. Créer des instances.

Nous représentons la connaissance en utilisant OWL2 (langage Web de l'ontologie).

**Les classes :** les classes ont présenté les concepts du domaine de fabrication, notre ontologie proposée contient sept concepts principaux: 'ManufacturingProcess', 'ManufacturingTask', 'ManufacturingProblem', 'ManufacturingSolution', 'ManufacturingFeature', 'ManufacturingAgent' and 'ManufacturingEquipment'. les principales classes et sous-classes sont illustrées dans la figure 3.13.

**Les propriétés d'objet:** nos propositions sur les relations entre les classes sont présentées dans le tableau 3.1.

Nous avons proposé une ontologie de domaine pour la tâche de fabrication, baptisée MATO (MANufacturing Task Ontology). Elle contient sept concepts principaux: Processus de fabrication, Tâche de fabrication, Problème de fabrication, Solution de fabrication, les caractéristiques de Fabrication, Fabrication et Agent de fabrication. Illustré dans les figure.3.13 et 3.14. chaque classe contient des sous classes, la figure 3.15 présente une structure des classes principales et la hiérarchie des sous classes.

- La classe "ManufacturingAgent" (figure 3.15.A), les sous-classes les plus importantes sont : OperatorAgent, ManagerAgent, EngineerAgent et ExpertAgent.
- La classe "ManufacturingEquipment" (figure 3.15.B) comprend deux sous-classes principales: MainEquipment et SecondEquipment.

- La “ManufacturingProcess” (figure 3.15.C), contient un ensemble d’activités et une méthode pour le système de production, deux sous-classes: ProductionSystem et productionMethod.
- La classe “ManufacturingProblem-Solution” (figure 3.15.D), y compris ManufacturingProblem et manufacturingSolution.
- La classe “ManufacturingFeature” (figure 3.15.E), notamment: TaskFeature, AgentFeature, EquipmentFeature, ProblemFeature, SolutionFeature.
- La classe “ManufacturingTask” (figure 3.15.F), cette classe comprenant principalement les tâches de fabrication: conception, surveillance, production, logistique, qualité, achats, ordonnancement, commerce.

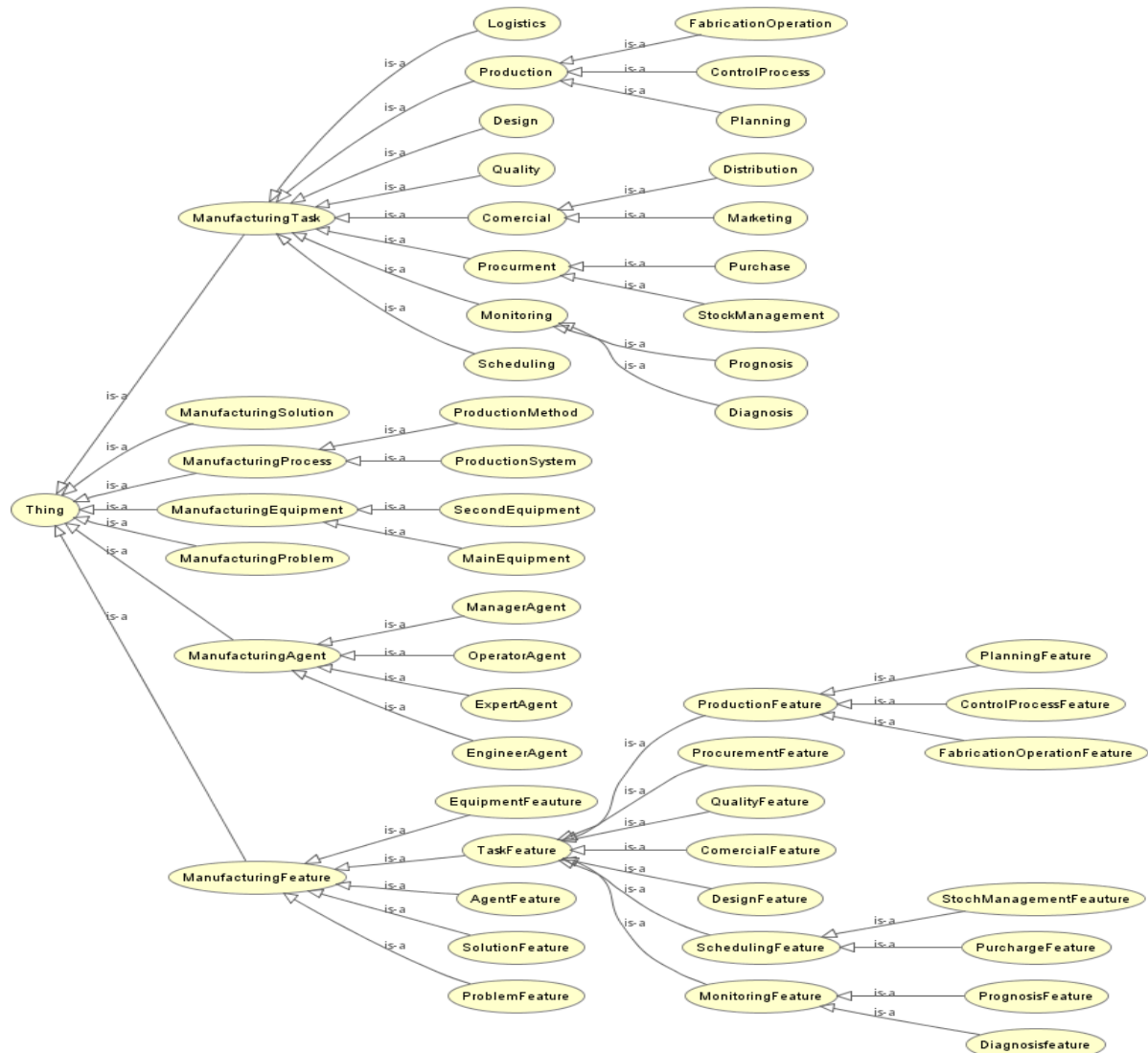


Figure 3.13 les concepts principaux de l’ontologie MATO

**Chapitre 3. Processus de capitalisation et de partage des connaissances proposées**

**Tableau 3.2** Les relations entre les concepts de l'ontologie MATO

Object properties	description	domain	range
affects	Property defines the relationship between ManufacturingProcess(domain) and ManufacturingTask (range)	ManufacturingProcess	ManufacturingTask
causes	Property defines the relationship between ManufacturingEquipment(domain) and ManufacturingProblem(range)	ManufacturingEquipment	ManufacturingProblem
hasFeature	Property defines the relationship between ManufacturingProcess(domain) and ManufacturingFeature (range)	ManufacturingEquipment ManufacturingTask ManufacturingProblem ManufacturingAgent	ManufacturingFeature
hasProblem	Property defines the relationship between ManufacturingTask(domain) and ManufacturingProblem(range)	ManufacturingTask	ManufacturingProblem
hasSolution	Property defines the relationship between ManufacturingProblem(domain) and ManufacturingSolution (range)	ManufacturingProblem	ManufacturingSolution
partOf	Property defines the relationship between ManufacturingFeature(domain) and TaskFeature, ProblemFeature, SolutionFeature, AgentFeature, EquipmentFeature (range)	TaskFeature, ProblemFeature SolutionFeature, AgentFeature, EquipmentFeature.	ManufacturingFeature
hasTask	Property defines the relationship between ManufacturingTask(domain) and Procurement, Scheduling, production, Monitoring, Design, Commercial and Quality (range)	ManufacturingTask ManufacturingAgent	Procurement, Scheduling, production, Monitoring, Design, Commercial and Quality
identifies	Property defines the relationship between ManufacturingAgent(domain) and ManufacturingProblem (range)	ManufacturingAgent	ManufacturingProblem
produces	Property defines the relationship between ManufacturingTask(domain) and ManufacturingFeature (range)	ManufacturingTask	ManufacturingFeature
realizes	Property defines the relationship between ManufacturingAgent(domain) and ManufacturingTask (range)	ManufacturingAgent	ManufacturingTask
proposes	Property defines the relationship between ManufacturingAgent(domain) and ManufacturingSolution (range)	ManufacturingAgent	ManufacturingSolution
isRealisedBy	Property defines the relationship between ManufacturingTask(domain) and ManufacturingAgent(range) (inverse of realises)	ManufacturingTask	ManufacturingAgent
controlOf	Property defines the relationship between ManufacturingAgent(domain) and ManufacturingTask, ManufacturingEquipment, ManufacturingProcess (range)	ManufacturingAgent	ManufacturingTask ManufacturingEquipment ManufacturingProcess
checkOf	Property defines the relationship between ManufacturingAgent(domain) and ManufacturingSystem, ManufacturingProblem, ManufacturingEquipment (range)	ManufacturingAgent	ManufacturingProcess ManufacturingProblem ManufacturingEquipment
isCausedBy	Property defines the relationship between ManufacturingProblem (domain) and ManufacturingEquipment (range) (inverse of causes)	ManufacturingProblem	ManufacturingEquipment
isProposedBy	Property defines the relationship between ManufacturingSolution(domain) and ManufacturingAgent(range) (inverse of proposes)	ManufacturingSolution	ManufacturingAgent
collaborateWith	Property defines the relationship between ManufacturingAgent(domain) and ManufacturingAgent(range)	ManufacturingAgent	ManufacturingAgent
isInterventionBy	Property defines the relationship between ManufacturingAgent(domain) and ManufacturingProblem, ManufacturingSolution(range)	ManufacturingProblem ManufacturingSolution	ManufacturingAgent

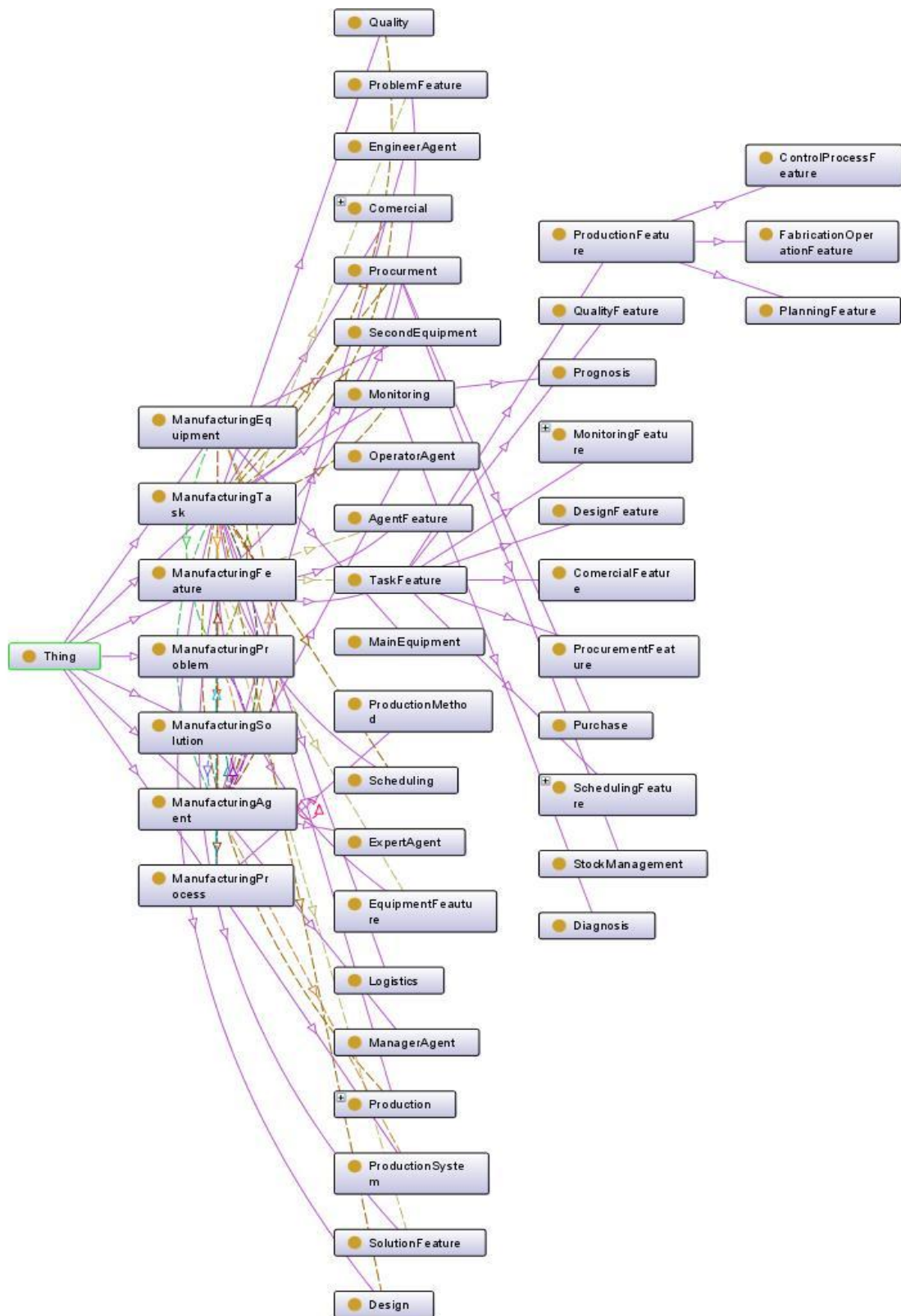


Figure 3.14 les concepts principaux de l'ontologie MATO par Ontograph

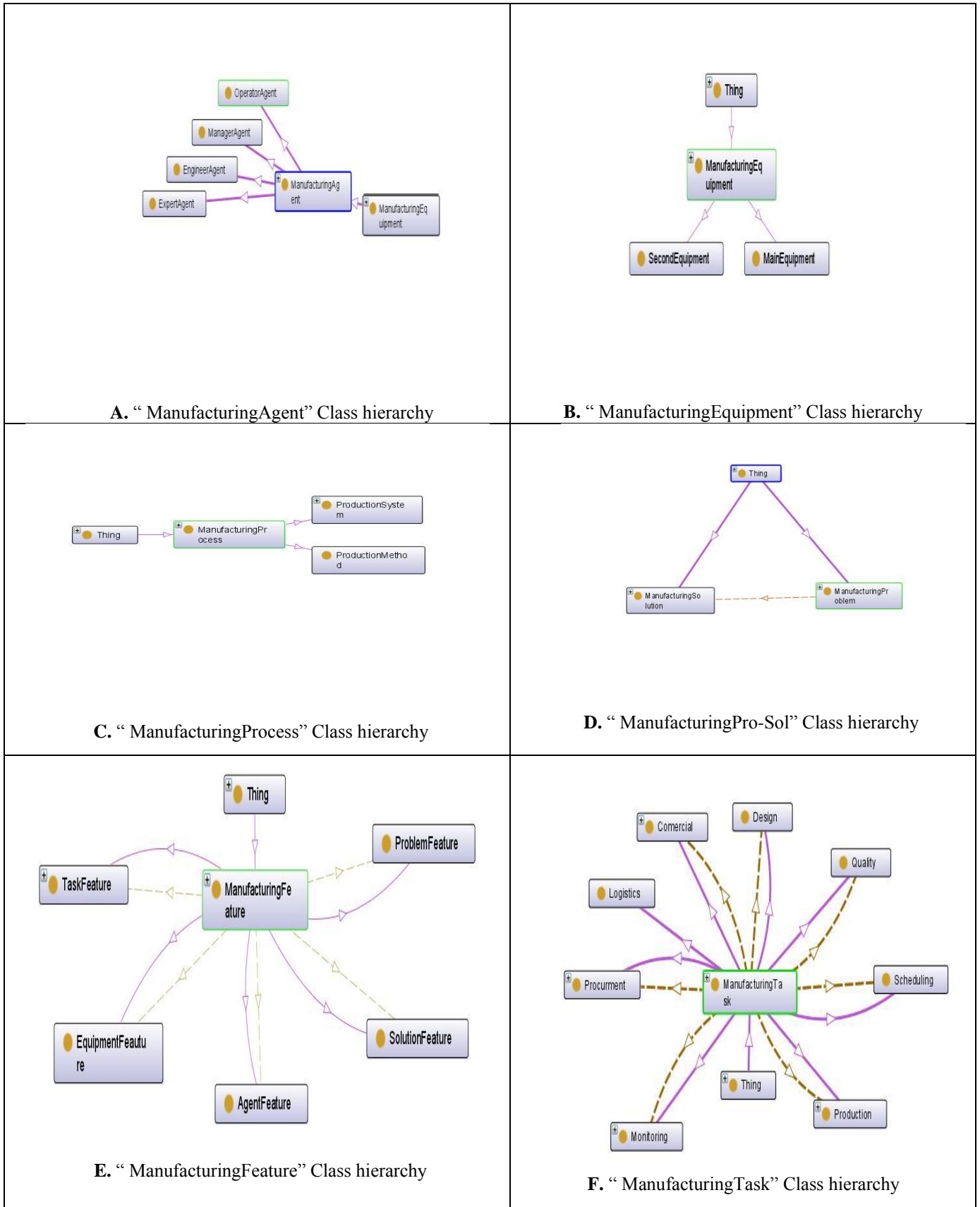


Figure 3.15 Structure des classes principales et la hiérarchie des classes



Les principaux objectifs de MATO pour la résolution de problèmes de fabrication sont les suivants:

- Faciliter le partage des connaissances entre les tâches de fabrication.
- Aider à capturer et à réutiliser les connaissances lors de l'exécution de tâches.
- Réduction du temps nécessaire à la description et à la localisation des problèmes de fabrication.
- Définir tous les agents manufacturiers (experts ou non experts) capables d'intervenir dans un domaine de tâche spécifique.
- Faciliter l'intervention des agents de fabrication et la description des équipements de fabrication.

Les caractéristiques de MATO sont la description de la connaissance de fabrication par Know-What, la connaissance de fabrication causale par Know-Why, la connaissance procédurale par Know-How et la connaissance relationnelle par Know-with.

- Know-what: description des connaissances relatives aux tâches de fabrication dans les principales classes: «ManufacturingAgent», «ManufacturingEquipment», «ManufacturingProcess», «ManufacturingProblem-Solution», «une partie de» «ManufacturingFeature», «ManufacturingTask» et des relations entre classes: “hasTask”, “hasProblem”, “hasSolution” et “hasFeature”.

- Savoir-why: présenter une relation de cause à effet entre la tâche de fabrication et la connaissance basée sur les propriétés d'un objet: «causes» et «est causée par».

- Know-how: présente une procédure de connaissance de la tâche de fabrication utilisant les propriétés de l'objet principal: «affecte», «identifie», «produit», «réalise», «propose», «controlOf», «sRealisedBy», «checkOf» et “IsInterventionBy”.

- Know-with: définir une connaissance relationnelle des tâches de fabrication, une relation entre une tâche de fabrication ou des relations entre l'agent de fabrication et les propriétés d'objet: «collaborateWith».

### **3.3.3 Définition des concepts principaux de l'ontologie MATO et leurs relations**

#### **3.3.3.1 Concept de processus de fabrication « *ManufacturingProcess* »**

La classe « *ManufacturingProcess* » a été reprise des concepts MRO (Manufacturing Reference Ontology) proposés par (Usman et al., 2013). Les processus de fabrication sont les étapes et les méthodes par lesquelles les intrants sont transformés en extrants afin de produire un produit final. Le processus de fabrication affecte de nombreuses tâches de fabrication, la conception, la production, la planification, la qualité, la surveillance, etc. Le processus de fabrication comporte des caractéristiques.

#### **3.3.3.2 Concept de tâche manufacturière « *ManufacturingTask* »**

La classe « *ManufacturingTask* » a été reprise des concepts « *manufacturing-operation* » de (Lin et al.2011) et « *Operation class* » de (Lemaingnan et al., 2006). Dans les domaines de fabrication, le besoin de partage de connaissances lors de l'exécution d'une tâche est primordial. Par contre, certaines tâches nécessitent un partage de connaissances en hauteur à partir d'une autre tâche pour son exécution. La description de concept « *ManufacturingTask* » et leurs classes hiérarchies est représentée dans la figure.3.16.

#### **3.3.3.3 Concept d'équipements manufacturiers « *ManufacturingEquipment* »**

La classe « *ManufacturingEquipment* » a été adoptée dans (Lin et al., 2011) et (Mohamed Hedi Karray et al., 2012). Dans notre ontologie proposée, le concept « *ManufacturingEquipment* » englobe tous les équipements, outils et machines utilisés dans le système de fabrication pour réaliser une tâche de fabrication. Ils contiennent deux concepts: «*MainEuipeument*» et «*secondlyEquipment*».

Ces informations concernent les ressources d'équipements, à savoir les Classes Équipement, l'équipement et le test de capacité de l'équipement qui peuvent être échangés entre les systèmes de l'entreprise et le système d'exécution de fabrication. La hiérarchie des ressources matérielles est représentée dans la Figure 3.17.

L'équipement peut être la définition de sites, de régions, de lignes de production, d'unités de production, de cellules de travail, de cellules ou de processus de l'unité. L'équipement peut être constitué d'autres équipements, avec des propriétés et des



capacités distinctes, par exemple une ligne de production peut être constituée de cellules de travail.

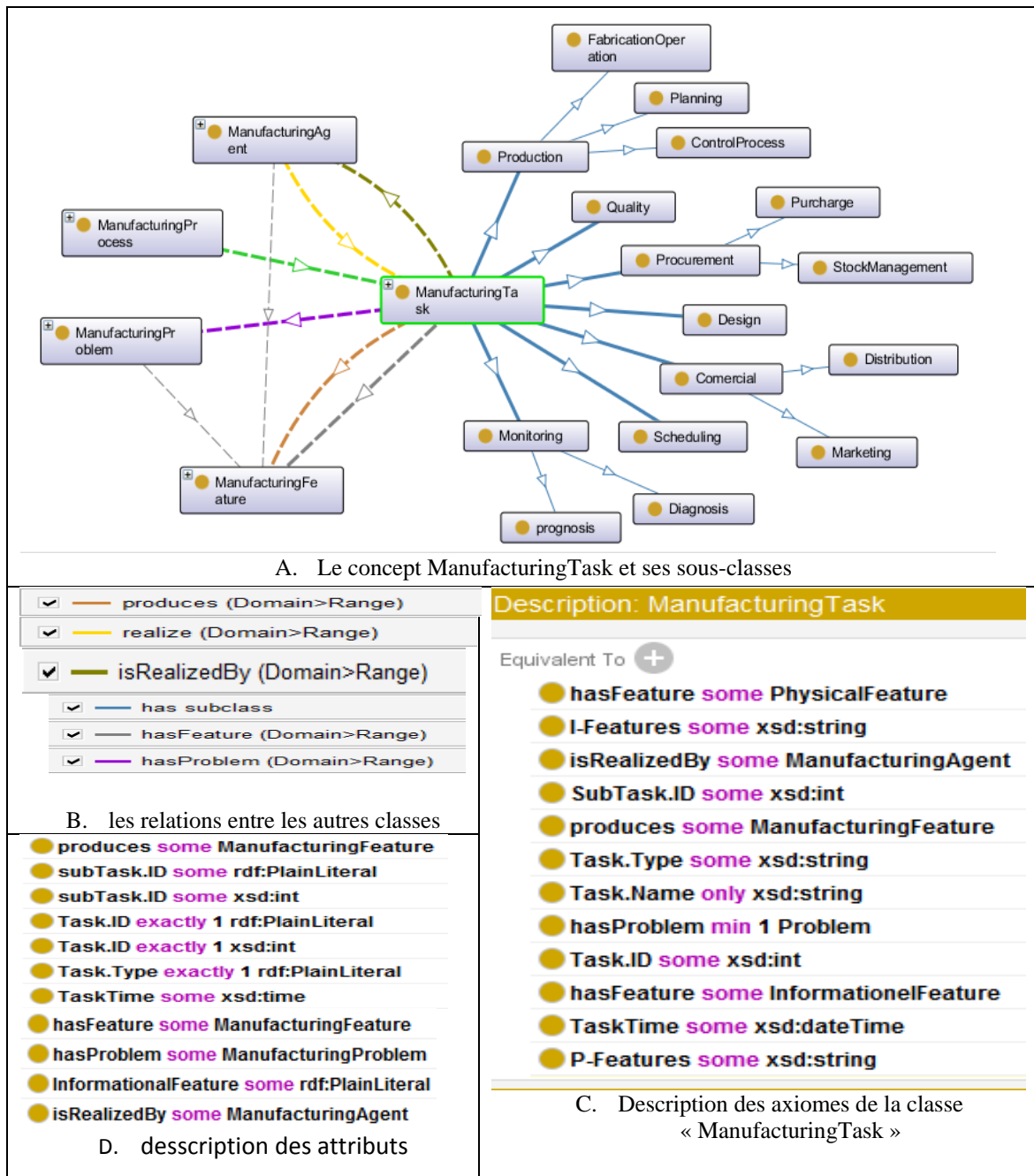


Figure 3.16 Description de concept « ManufacturingTask »

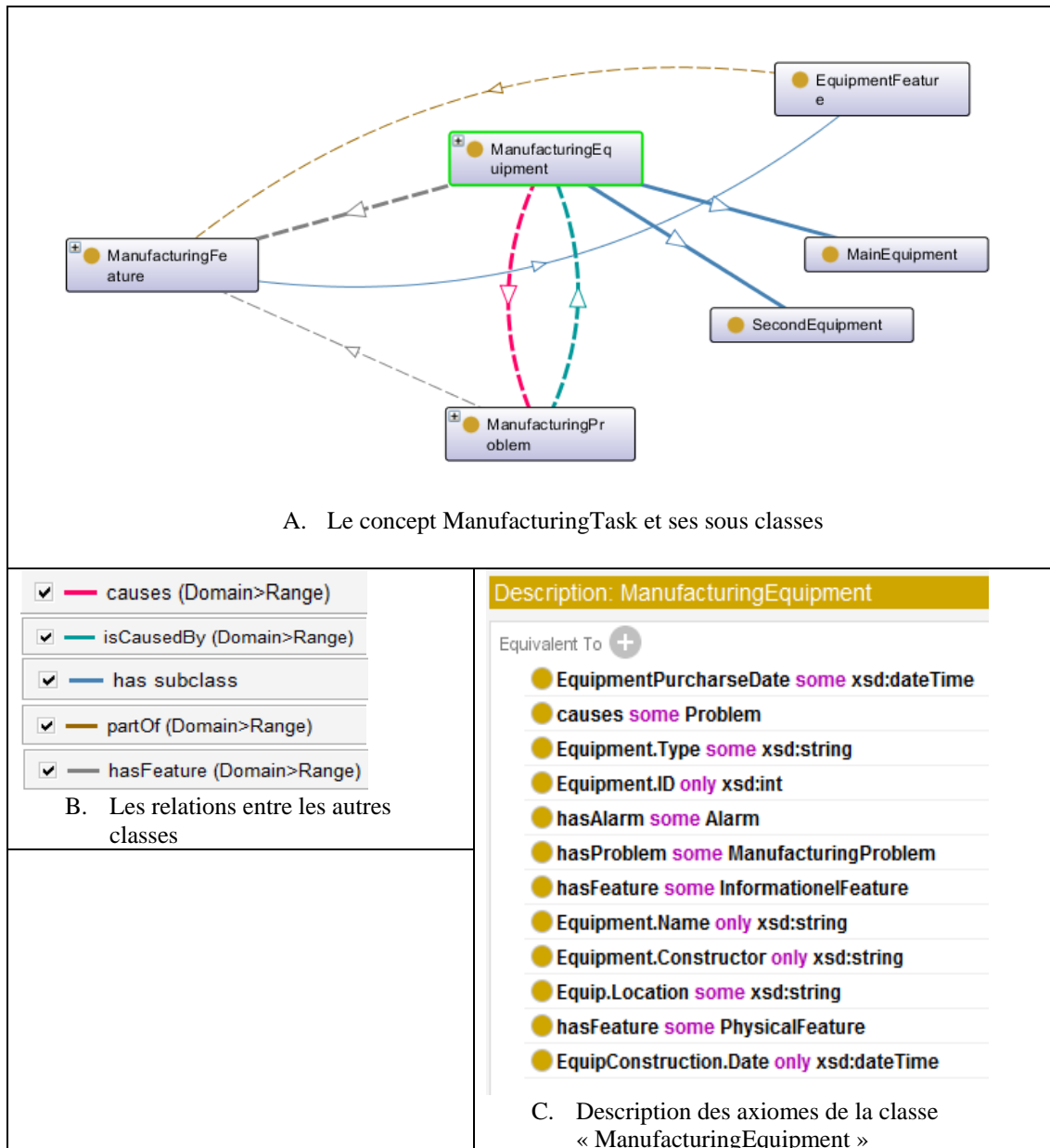


Figure 3.17 Description de concept « ManufacturingEquipment »

### 3.3.3.4 Concept d'agent manufacturier « ManufacturingAgent »

La classe « ManufacturingAgent » est issue du concept « Ressources humaines » de (Lemaingnan et al.2006), le rôle d'agent de fabrication dans le système de fabrication étant la tâche de fabrication, identifiant les problèmes de fabrication et proposant des solutions de fabrication. Dans notre ontologie de tâches de fabrication, les principaux agents de fabrication sont les suivants: responsable, ingénieur, opérateur et expert. La description des agents manufacturiers est représentée dans la Figure 3.18.

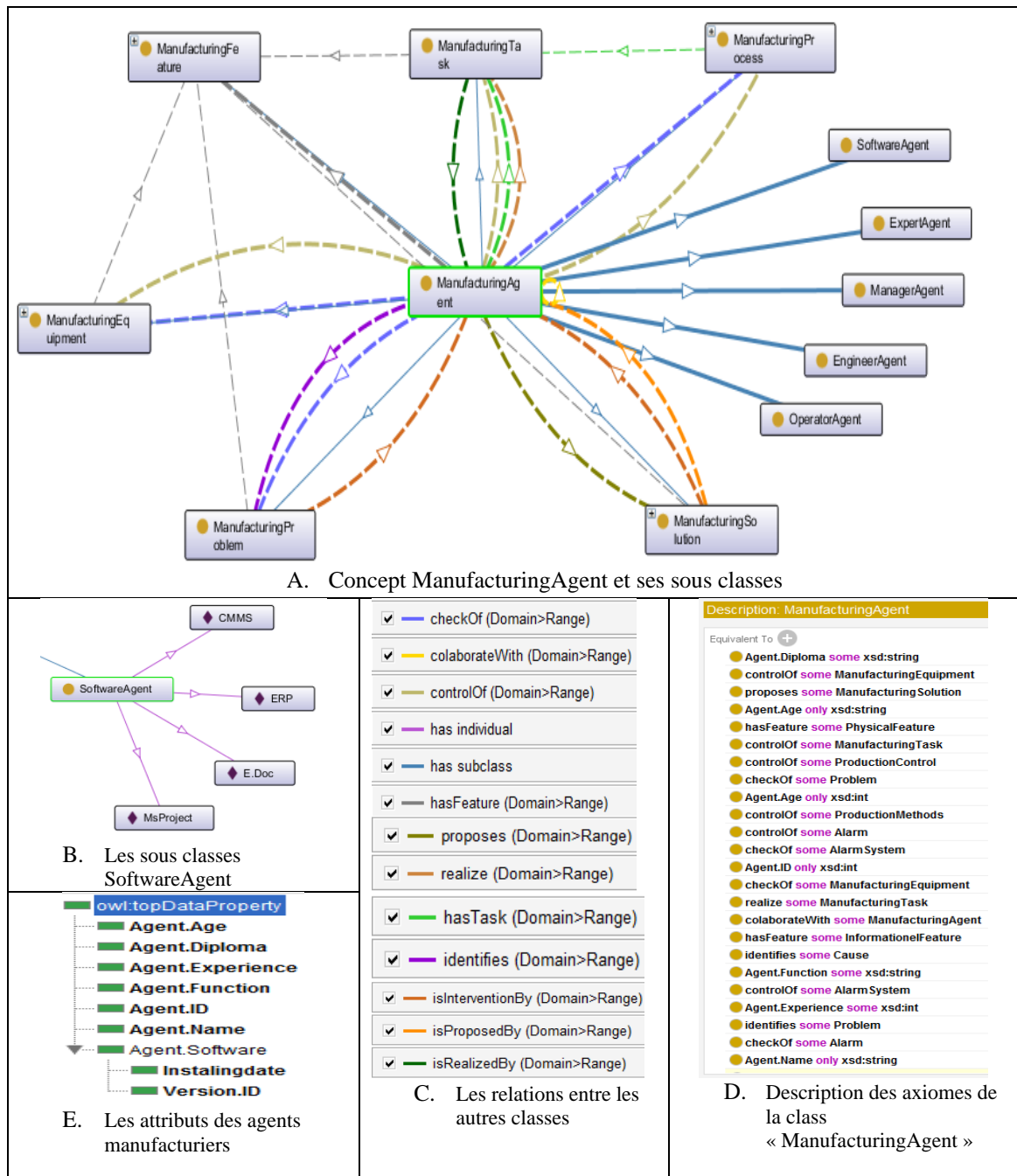


Figure 3.18 Description du concept « ManufacturingAgent »

### 3.3.3.5 Concept de problème manufacturier « ManufacturingProblem/Solution »

Les ontologies de fabrication existantes ne considèrent pas l'aspect résolution de problèmes dans leurs ontologies, nous avons ajouté deux concepts « ManufacturingProblem » et « ManufacturingSolution » afin de faciliter la résolution du problème de fabrication lors de l'exécution de la tâche de fabrication. La tâche de fabrication pose de nombreux problèmes de fabrication. Notre contribution principale est l'ajout du concept « ManufacturingProblem » pour faciliter la description du

problème et collecte automatiquement les informations, une description des agents manufacturiers est représentée dans la Figure 3.19.

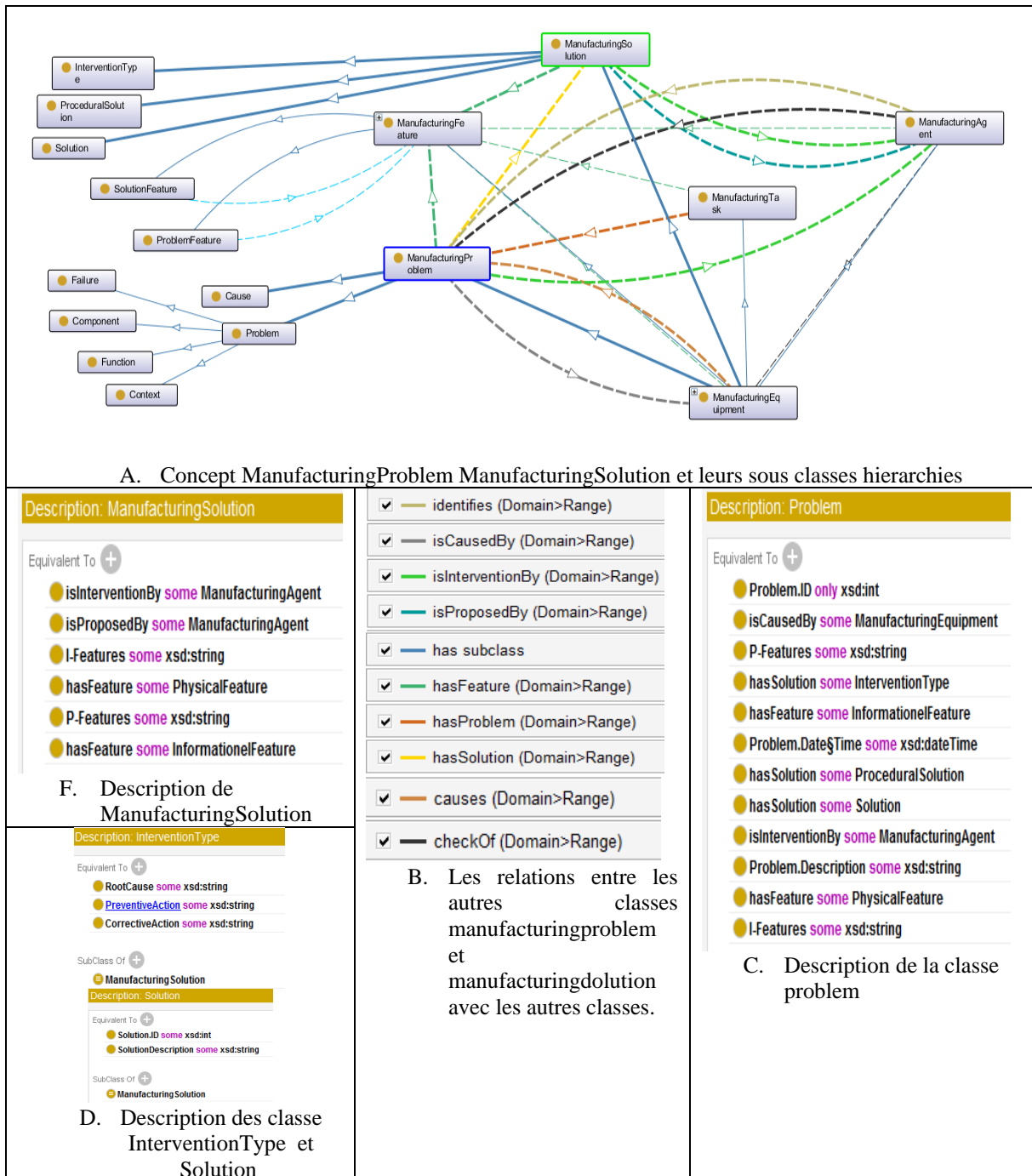


Figure 3.19 Description du concept « ManufacturingProblem/Solution »

### 3.3.3.6 Concept des caractéristiques manufacturières « ManufacturingFeature »

La classe « ManufacturingFeature » a été adoptée à partir de (Lin et al.2011; Young et al.2007; Lemaingnan et al.2006 ; Usman et al. 2013). La fonction de

fabrication décrit tous les concepts d'éléments d'ontologies de fabrication et facilite le partage des connaissances entre les tâches et les agents de fabrication, à l'intérieur et à l'extérieur des entreprises.

La classe « ManufacturingFeature » a été adoptée à partir de (Lin et al.2011; Young et al.2007; Lemaingnan et al.2006 ; Usman et al. 2013). La fonction de fabrication décrit tous les concepts d'éléments d'ontologies de fabrication et facilite le partage des connaissances entre les tâches et les agents de fabrication, à l'intérieur et à l'extérieur des entreprises. Le concept Feature au niveau générique a été défini de manière informelle comme; La caractéristique est tout ce qui a un attribut d'intérêt particulier. Selon Sanfilippo et Borgo (2016), il existe deux types des caractéristiques, tels que les caractéristiques informationnelles (I.Features) et les caractéristiques Physiques (P-Features). Les deux notions sont différentes

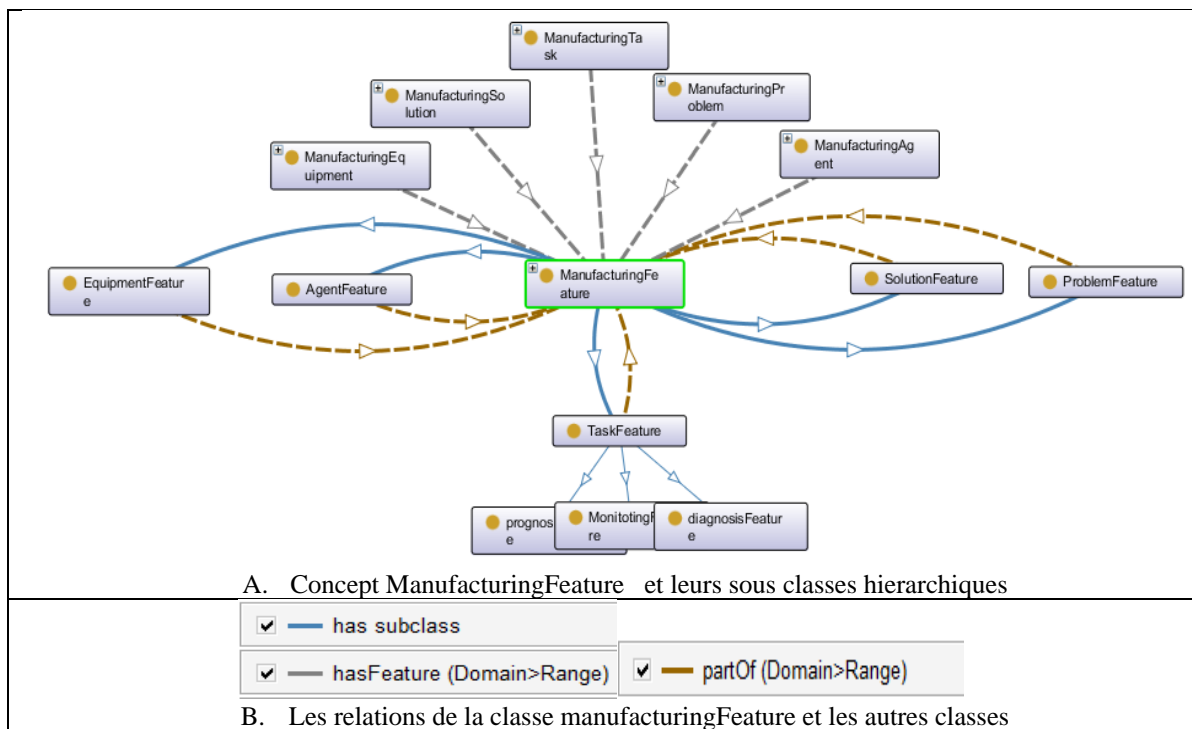


Figure 3.20 Description du concept « ManufacturingFeature »

Les caractéristiques informationnelles (I.Features) indiquent le sens d'une entité d'information, et un élément du modèle ou une unité d'information décrivant une agrégation des propriétés d'un modèle de produit qui sont pertinentes dans le cadre d'une vue spécifique sur le produit. Les caractéristiques Physiques (P-Features) sont des caractéristiques, au sens d'une entité physique, et d'un élément lié à un objet physique, généralement le produit lui-même. Par exemple, les caractéristiques d'un

produit qui résultent de la conception ou un constituant physique d'une pièce. En ce sens, une qualité (par exemple couleur, forme), un composant (physique, chimique). Une description des agents manufacturiers est représentée dans la Figure 3.20. Le concept « ManufacturingFeature » a été entièrement formalisé pour expliquer l'adéquation de MATO en tant qu'ontologie de fabrication des tâches. La formalisation des concepts des caractéristiques et la façon dont ils sont utilisés pour expliquer les nouveaux aspects de la recherche et l'efficacité du MATO sont détaillés au chapitre 4. Nous présentons aussi les classes principales de l'ontologie MATO et leurs relations dans la figure.3.21.

### 3.3.4 Formalisation de l'ontologie MATO

#### 3.3.4.1 Formalisation de la connaissance du domaine

Le langage d'ontologie Web (OWL) est conçu pour représenter des connaissances variées sur les objets et leurs relations. Il est largement utilisé pour exprimer des modèles complexes et des informations d'adresse. Tels que les systèmes de fabrication complexes. Avec l'aide d'OWL, les agents de fabrication sont en mesure de capitaliser les connaissances en matière de fabrication et sont faciles à réutiliser et à partager pour résoudre des problèmes de fabrication. Faciliter également la communication entre experts et agents industriels ayant des terminologies communes, un exemple des connaissances du domaine en langage OWL (voir tableau 3.3)

Tableau 3.3 Exemple de formalisation des connaissances MATO en OWL

---

```
<rdfs:comment rdf:datatype=http://www.w3.org/2001/XMLSchema#string > MATO  
(MANufacturing Tasks Ontology) : an off line knowledge sharing  
process</rdfs:comment>  
<owl:versionInfo rdf:datatype=http://www.w3.org/2001/XMLSchema#string  
>MATO.1.0 MANufacturing Task Ontology Version 1.0: Laboratory of Automation  
and manufacturing, University of batna 2</owl:versionInfo>
```

---

</owl:Class><owl:Classrdf:ID="ManufacturingProcess"/>	<owl:Class rdf:ID="ManagerAgent">
<owl:Classrdf:about="#Comercial"><rdfs:subClassOf>	<rdfs:subClassOfrdf:resource="#ManufacturingAgent"/></owl:Class>
<owl:Classrdf:about="#ManufacturingTask"/>	<owl:Class
</rdfs:subClassOf></owl:Class>	rdf:ID="MainEquipment"><rdfs:subClassOfrdf:resource="#ManufacturingEquipment"/></owl:Class>
<owl:Classrdf:ID="MainEquipmentFeature">	

---



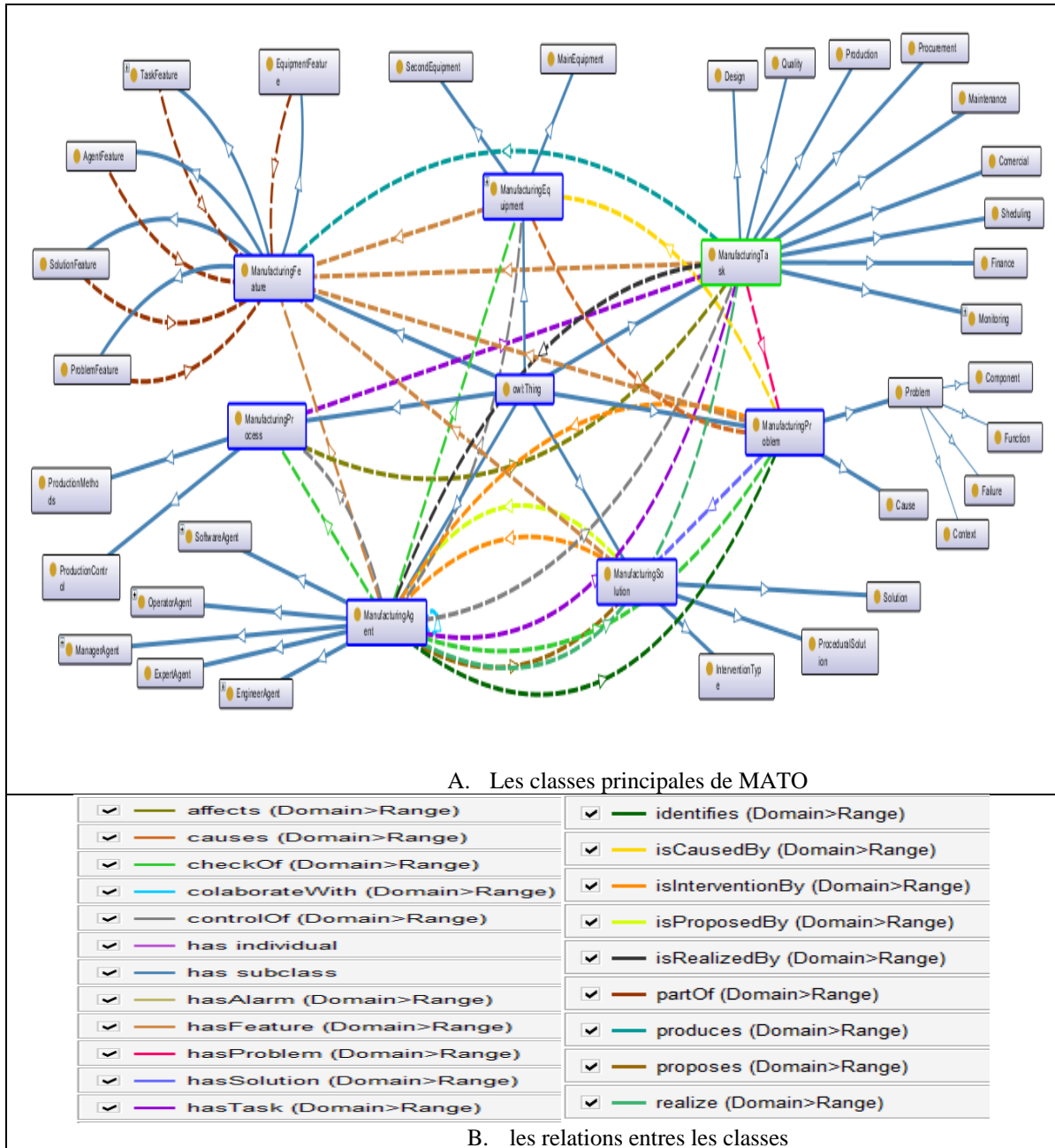


Figure 3.21 Les classes principales de l'ontologie MATO et leurs relations

### 3.3.4.2 Formalisation de la connaissance d'inférence

Le langage de règles du web sémantique (SWRL). SWRL comprend deux parties principales: l'antécédent et le conséquent. Un symbole implicite '→' est utilisé pour relier ces deux parties. De plus, le symbole de conjonction '∧' est utilisé pour connecter différents atomes. Les variables en atomes sont représentées par l'identifiant d'interrogation «?». Les Formalisation de la connaissance d'inférence par SWRL présentés au tableau 3.4.

- Identification de problème, le processus et la tâche lié au problème

**Règle.1** la règle 1 est utilisée pour la configuration initiale du problème de fabrication à partir des instances du processus de fabrication, la tâche de fabrication et le problème lié à la tâche, utilise la propriété *hasProblem*.

**Règle.Gén.1:**  $ManufacturingProcess(?P) \wedge ManufacturingTask(?T) \wedge ManufacturingProblem(?Pr) \wedge Task(?P, ?T) \Rightarrow hasProblem(?T, ?Pr)$

- Identification des caractéristiques du problème

**Règle.2** La règle 2 identifie les caractéristiques du problème, utilise la propriété *hasFeature* associé à chaque instance de la définition du problème et agrège ces instances au sein des instances de la classe caractéristique de la fabrication.

**Règle.Gén.2:**  $ManufacturingProblem(?Pr) \wedge ManufacturingFeature(?F) \Rightarrow hasFeature(?Pr, ?F)$ .

- Identification les caractéristiques de problème, de l'équipement et de l'agent

**Règle.3** la règle 3 est utilisée pour identifier les caractéristiques, les problèmes des équipements et des agents intervenants à ce problème, à partir des instances du problème de fabrication, des équipements et des agents, utilise la propriété *hasFeature* pour l'identification des caractéristiques des problèmes, des équipements et des agents.

**Règle.Gén.3:**  $ManufacturingProblem(?Pr) \wedge ManufacturingFeature(?F) \wedge ManufacturingEquipment(?E) \wedge hasProblem(?E, ?Pr) \wedge ManufacturingAgent(?A) \Rightarrow hasFeature(?Pr, ?F) \wedge hasFeature(?E, ?F) \wedge hasFeature(?A, ?F)$ .

- Identification les causes possibles de problèmes

**Règle.4** Cette règle de configuration détermine l'identification de la cause de problèmes associés aux instances des équipements, sur la base des exigences d'instances de problèmes et des équipements.

**Règle.Gén.4:**  $ManufacturingProblem(?Pr) \wedge ManufacturingEquipment(?E) \wedge hasProblem(?E, ?Pr) \Rightarrow causes(?E, ?Pr)$ .

On peut aussi appliquer la règle 4 par l'utilisation de la relation inverse *isCausedby*.

**Règle.Gén.4+:**  $ManufacturingProblem(?Pr) \wedge ManufacturingEquipment(?E) \wedge hasProblem(?E, ?Pr) \Rightarrow isCausedby(?Pr, ?E)$ .

- Identifier les agents intervenant aux problèmes

**Règle.5** cette règle est appliquée pour gérer la configuration des agents de fabrication qui sont capables d'intervenir aux problèmes de fabrication. Ceci est réalisé par l'identification des problèmes puis par l'identification des instances des agents.

**Règle.Gén.5:**  $ManufacturingProblem(?Pr) \wedge ManufacturingAgent(?A) \Rightarrow identifies(?A, ?Pr)$ .

- Proposer la solution liée au problème



### Règle.6

**Règle.Gén.6:**  $Problem(?Pr) \wedge ManufacturingEquipment(?E) \wedge Solution(?Sol) \wedge hasProblem(?E, ?Pr) \wedge causes(?E, ?Pr) \rightarrow hasSolution(?Pr, ?Sol)$

- Identifier la procédure de la solution

### Règle.7

**Règle.Gén.7:**  $Problem(?Pr) \wedge causes(?E, ?Pr) \wedge Solution(?Sol) \wedge hasSolution(?Pr, ?Sol) \wedge InterventionType(?IT) \rightarrow hasInterventionType(?Sol, ?IT)$

- Identifier la procédure de la solution

**Règle.8** La règle 8, les procédures de la solution à partir de la solution proposée au problème.

**Règle.Gén.8:**  $Problem(?Pr) \wedge causes(?E, ?Pr) \wedge Solution(?Sol) \wedge hasSolution(?Pr, ?Sol) \wedge ProceduralSolution(?PS) \rightarrow hasProceduralSolution(?Sol, ?PS)$

- Identification des agents intervenants à la solution

**Règle.9** La règle 9 propose les solutions des problèmes de fabrication par les agents manufacturiers.

**Règle.Gén.9:**  $ManufacturingProblem(?Pr) \wedge ManufacturingAgent(?A) \wedge ManufacturingSolution(?Sol) \Rightarrow proposes(?A, ?Sol)$

**Règle.10** On peut aussi appliquer la règle inverse de la règle 10 par l'utilisation de la relation inverse *isProposedBy*.

**Règle.Gén.10:**  $ManufacturingProblem(?Pr) \wedge ManufacturingAgent(?A) \wedge ManufacturingSolution(?Sol) \Rightarrow isProposedBy(?Sol, ?A)$

- Configuration de la tâche, caractéristique, processus et Affectation, des agents des problèmes et solutions et équipements

**Règle.11** Après l'application séquentielle des règles de configuration et des règles d'affectation dans SWRL, cette règle identifie les problèmes liés à la tâche, aussi définit les caractéristiques des problèmes de fabrication, des équipements de fabrication et des agents manufacturiers, identification des causes des problèmes liées aux équipements, affecter les agents capables d'intervenir à ces problèmes et les agents réalisateurs de ces tâches, les contrôleurs des équipements, les identificateurs des problèmes et les solutions proposées par ces agents.

**Règle.Gén.11:**  $ManufacturingProcess(?P) \wedge ManufacturingTask(?T) \wedge ManufacturingProblem(?Pr) \wedge ManufacturingFeature(?F) \wedge ManufacturingEquipment(?E) \wedge ManufacturingAgent(?A) \wedge ManufacturingSolution(?Sol) \wedge hasTask(?P, ?T) \wedge hasProblem(?E, ?Pr) \wedge collaborateWith(?A, ?A) \Rightarrow hasProblem(?T, ?Pr) \wedge hasFeature(?Pr, ?F) \wedge hasFeature(?E, ?F) \wedge hasFeature(?A, ?F) \wedge causes(?E, ?Pr) \wedge isCausedby(?Pr, ?E) \wedge identifies(?A, ?Pr) \wedge$

### Chapitre 3. Processus de capitalisation et de partage des connaissances proposées

*proposes(?A, ?Sol) ^ realizes(?A, ?T) ^ isRealizedBy(?T, ?A) ^ controlOf(?A, ?E) ^ isInterventionBy(?Pr, ?A) ^ identifies(?A, ?Pr) ^ proposes(?A, ?Sol).*

**Tableau 3.4** les règles générique SWRL de MATO.

Régle.Gén.1	<u>Identification de problème, le processus et la tâche lié au problème</u> <i>ManufacturingProcess(?P) ^ ManufacturingTask(?T) ^ ManufacturingProblem(?Pr) ^ hasTask(?P, ?T) -&gt; hasProblem(?T, ?Pr)</i>
Régle.Gén.2	<u>Identification les caractéristiques du problème</u> <i>ManufacturingProblem(?Pr) ^ ManufacturingFeature(?F) -&gt; hasFeature(?Pr, ?F)</i>
Régle.Gén.3	<u>Identification les caractéristiques de problème, de l'équipement et de l'agent</u> <i>ManufacturingProblem(?Pr) ^ ManufacturingFeature(?F) ^ ManufacturingEquipment(?E) ^ hasProblem(?E, ?Pr) ^ ManufacturingAgent(?A) -&gt; hasFeature(?Pr, ?F) ^ hasFeature(?E, ?F) ^ hasFeature(?A, ?F)</i>
Régle.Gén.4	<u>Identification les causes possibles de problème</u> <i>ManufacturingProblem(?Pr) ^ ManufacturingEquipment(?E) ^ hasProblem(?E, ?Pr) -&gt; causes(?E, ?Pr)</i>
Régle.Gén.5	<u>identifier les agents intervenant au problème</u> <i>ManufacturingProblem(?Pr) ^ ManufacturingAgent(?A) -&gt; identifies(?A, ?Pr)</i>
Régle.Gén.6	<u>Proposer la solution liée au problème</u> <i>Problem(?Pr) ^ ManufacturingEquipment(?E) ^ Solution(?Sol) ^ hasProblem(?E, ?Pr) ^ causes(?E, ?Pr) -&gt; hasSolution(?Pr, ?Sol)</i>
Régle.Gén.7	<u>Identifier le type d'intervention</u> <i>Problem(?Pr) ^ causes(?E, ?Pr) ^ Solution(?Sol) ^ hasSolution(?Pr, ?Sol) ^ InterventionType(?IT) -&gt; hasInterventionType(?Sol, ?IT)</i>
Régle.Gén.8	<u>Identifier la procédure de la solution</u> <i>Problem(?Pr) ^ causes(?E, ?Pr) ^ Solution(?Sol) ^ hasSolution(?Pr, ?Sol) ^ ProceduralSolution(?PS) -&gt; hasProceduralSolution(?Sol, ?PS)</i>
Régle.Gén.9	<u>Identification les agents intervenant à la solution</u> <i>Problem(?Pr) ^ ManufacturingAgent(?A) ^ autogen0: Solution(?Sol) -&gt; proposes(?A, ?Sol)</i>
Régle.Gén.10	<u>La règle inverse de la Régle.Gén.9</u> <i>Problem(?Pr) ^ ManufacturingAgent(?A) ^ autogen0: Solution(?Sol) -&gt; isProposedBy(?Sol, ?A)</i>
Régle.Gén.11	<u>Configuration de la tâche, caractéristique, processus et Affectation, des agents des problèmes et solution et équipements</u> <i>ManufacturingProcess(?P) ^ ManufacturingTask(?T) ^ ManufacturingProblem(?Pr) ^ ManufacturingFeature(?F) ^ ManufacturingEquipment(?E) ^ ManufacturingAgent(?A) ^ ManufacturingSolution(?Sol) ^ hasTask(?P, ?T) ^ hasProblem(?E, ?Pr) ^ collaborateWith(?A, ?A) -&gt; hasProblem(?T, ?Pr) ^ hasFeature(?Pr, ?F) ^ hasFeature(?E, ?F) ^ hasFeature(?A, ?F) ^ causes(?E, ?Pr) ^ isCausedBy(?Pr, ?E) ^ identifies(?A, ?Pr) ^ proposes(?A, ?Sol) ^ realize(?A, ?T) ^ isRealizedBy(?T, ?A) ^ controlOf(?A, ?E) ^ isInterventionBy(?Pr, ?A)</i>

**Règle.12** Cette règle d'affectation détermine l'identification des agents manufacturiers pour effectuer leurs tâches.

**Régle.Gén.12:** *ManufacturingTask(?T) ^ ManufacturingAgent(?A) ⇒ realizes(?A, ?T).*

**Règle.13** Cette règle d'affectation est l'application inverse de la règle 12.

**Régle.Gén.13:** *ManufacturingTask(?T) ^ ManufacturingAgent(?A) ⇒ isRealizedBy(?T, ?A).*

**Règle.14** Cette règle de configuration précise les identifications individuelles des agents manufacturiers pour contrôler les équipements, basées sur la spécification des agents.

**Régle.Gén.14:**  $ManufacturingAgent(?A) \wedge ManufacturingEquipment(?E) \Rightarrow controlOf(?A, ?E)$ .

**Règle.15** Cette règle d'affectation affecte à chaque problème de fabrication les exigences des agents manufacturiers pour chaque intervention aux problèmes manufacturiers.

**Régle.Gén.15:**  $ManufacturingProblem(?P) \wedge ManufacturingAgent(?A) \Rightarrow isInterventionBy(?Pr, ?A)$ .

**Règle.16** cette règle est une combinaison entre la règle 5 et la règle 9.

**Régle.Gén.16:**  $ManufacturingProblem(?P) \wedge ManufacturingAgent(?A) \wedge ManufacturingSolution(?Sol) \wedge collaborateWith(?A, ?A) \Rightarrow identifies(?A, ?Pr) \wedge proposes(?A, ?Sol)$ .

### 3.3.5 Interrogation de l'ontologie de MATO par les requêtes SQWRL

L'interrogation de l'ontologie MATO est basée sur la sélection de connaissances qui permet de chercher des règles SWRL en se basant sur leur contenu, sur les classes et propriétés utilisées, utilisant le langage des requêtes SQWRL. Cette sélection joue un rôle important dans le partage des connaissances durant l'exécution d'une tâche industrielle, il peut également être intéressant de sélectionner des connaissances liées à la tâche, aux équipements, aux agents manufacturiers, aux problèmes et leurs solutions.

Ces requêtes sont génériques et applicables pour toutes les tâches manufacturières, telles que le diagnostic, l'approvisionnement, la logistique, la planification ...etc.

Sélection et réutilisation des connaissances sur le problème manufacturier :

La requête *sqwrl.Pr* permet de sélectionner les connaissances liée aux problèmes manufacturiers, les connaissances sur le problème, l'identification du problème, le type de problème et sa fréquence. Ces connaissances utilisent le vocabulaire de ce domaine, peuvent contenir des informations intéressantes permettant aux agents manufacturiers de raisonner avec les connaissances de ce domaine et faciliter la tâche d'identification du problème.

**sqwrl.Pr :**  $Problem(?Pr) \wedge ProblemID(?Pr, ?ID) \wedge ProblemType(?Pr, ?Tp) \wedge ProblemDescription(?Pr, ?Ds) \wedge ProblemFrequency(?Pr, ?Fr) \rightarrow sqwrl.select(?Pr, ?ID, ?Ds, ?Tp, ?Fr)$

*La sélection et réutilisation des connaissances sur les caractéristiques de problème manufacturier :*

D'après la sélection des connaissances sur le problème par la requête *sqwrl.Pr*, pour ce faire, il faut interroger aussi l'ontologie par la requête *Sqwrl.PrF* qui permet

de sélectionner des connaissances sur les caractéristiques physiques et informationnelles du problème manufacturier, la partie antécédente de cette requête utilise les classes *Problem(?Pr)* , *ProblemFeature(?PF)*, et les relations *hasFeature(?Pr,?PF)* , *I-Features(?PF,?If)*, *P-Features(?PF,?Pf)*.

**Sqwrl.PrF :** *Problem(?Pr) ^ ProblemFeature(?PF) ^ hasFeature(?Pr,?PF) ^ I-Features(?PF,?If)^P-Features(?PF,?Pf)->sqwrl:select(?Pr,?PF,?If,?Pf)*

Toutes les requêtes proposées pour interroger notre ontologie sont illustrées dans le tableau 3.5.

**Tableau 3.5** les règles génériques SQWRL pour MATO

sqwrl.Pr	<u>Sélection de la connaissance sur le problème</u> <i>Problem(?Pr) ^ ProblemID(?Pr, ?ID) ^ ProblemType(?Pr, ?Tp) ^ ProblemDescription(?Pr, ?Ds) ^ ProblemFrequency(?Pr, ?Fr) -&gt; sqwrl:select(?Pr, ?ID, ?Ds, ?Tp, ?Fr)</i>
Sqwrl.PrF	<u>Sélection de la connaissance sur les caractéristique de problème</u> <i>Problem(?Pr)^ProblemFeature(?PF)^hasFeature(?Pr,?PF)^I-Features(?PF,?If)^P-Features(?PF,?Pf)-&gt;sqwrl:select(?Pr,?PF,?If,?Pf)</i>
Sqwrl.CsPr	<u>Sélection de la connaissance sur les causes de problème</u> <i>Problem(?Pr) ^ Cause(?Cs) ^ hasCause(?Pr, ?Cs) ^ Component(?Cp) ^ Context(?Ct) ^ Failure(?FL) ^ Function(?Ft) ^ ComponentID(?Cp, ?CpID) ^ ComponentType(?Cp, ?Cpt) ^ ContextType(?Ct, ?Ctp) ^ ContextID(?Ct, ?CtID) ^ FailureID(?FL, ?FID) ^ FulfilmentRate(?FL,?Rate)^FunctionID(?Ft, ?FtID) ^ FunctionType(?Ft, ?FtT) -&gt; sqwrl:select(?Pr, ?Cs, ?Cp, ?CpID, ?Cpt, ?Ct, ?CtID, ?Ctp, ?FL, ?FID, ?Rate, ?Ft, ?FtID, ?FtT)^sqwrl:orderBy(?CpID)</i>
sqwrl.Ts	<u>Sélection la connaissance sur la tâche</u> <i>ManufacturingTask(?T) ^ TaskID(?T, ?ID) ^ TaskName(?T, ?N) ^ SubTaskID(?T, ?SID) ^ TaskType(?T, ?Tp) ^ TaskTime(?T, ?Tm) -&gt; sqwrl:select(?T, ?ID, ?N, ?SID, ?Tp, ?Tm)</i>
sqwrl.Sol	<u>Sélection de la connaissance sur la solution</u> <i>Solution(?Sol) ^ RootCause(?sol, ?RC) ^ CorrectiveAction(?Sol, ?CA) ^ PreventiveAction(?Sol, ?PA) ^ SolutionID(?Sol, ?ID) ^ SolutionDEscription(?Sol, ?DS) ^ InterventionType(?IT) ^ ProceduralSolution(?PS) ^ partOf(?IT, ?Sol) ^ partOf(?PS, ?Sol) -&gt; sqwrl:select(?Sol, ?ID, ?RC, ?CA, ?PA, ?DS, ?PS, ?IT)</i>
Sqwrl.SolF	<u>Sélection de la connaissance sur les caractéristique de Solution</u> <i>Solution(?Sol) ^ SolutionFeature(?PF) ^ hasFeature(?Sol, ?SF) ^ P-Features(?SF, ?Pf) -&gt; sqwrl:select(?Sol, ?SF, ?Pf)</i>
Sqwrl.Equip	<u>Sélection de la connaissance sur les équipements</u> <i>ManufacturingEquipment(?E)^EquipmentID(?E,?ID)^EquipmentConstructor(?E,?EC)^EquipConstructionDate(?E,?CD)^EquipLocation(?E,?EL)^EquipmentType(?E,?ET)-&gt; sqwrl:select(?E, ?ID, ?EC,?CD,?EL,?ET)</i>
Sqwrl.EquipF	<u>Sélection la connaissance sur les caractéristique des équipements sélectionner</u> <i>ManufacturingEquipment(?E) ^ EquipmentFeature(?EF) ^ hasFeature(?E, ?EF) ^ P-Features(?EF, ?Pf) -&gt; sqwrl:select(?E, ?EF, ?Pf)</i>
Sqwrl.Ag	<u>Sélection de la connaissance sur l'agent</u> <i>ManufacturingAgent(?A)^AgentAge(?A,?Ag)^AgentDiploma(?A,?AD)^AgentExperience(?A,?Exp)^AgentFunction(?A,?F)^AgentID(?A,?ID)^AgentName(?A,?N)</i>
Sqwrl.Sof	<u>Sélection l'information sur l'agent software</u> <i>SoftwareAgent(?SA)^InstalingDate(?SA,?InD)^VersionID(?SA,?V)^AgentFunction(?SA,?AF)^AgentID(?A,?ID)^AgentName(?SA,?N)</i>

SqwrL. AgF	<u>Sélection de la connaissance sur les caractéristique sur l'agent</u> $ManufacturingAgent(?A) \wedge AgentFeature(?AF) \wedge hasFeature(?A, ?AF) \wedge I-Features(?AF, ?If) \wedge P-Features(?AF, ?Pf) \rightarrow sqwrl:select(?A, ?AF, ?If, ?Pf)$
---------------	---

### **3.3.6 Synthèse de modèle proposé**

Le processus de partage des connaissances comprend quatre composants clés:

- (1) La base de connaissances : C'est l'élément le plus important du processus, la connaissance du domaine par l'ontologie MATO et une base des règles SWRL.
- (2) Le système de gestion de l'ontologie : Dans le cas de la présente étude, Protégé a été utilisé, qui aide les éditeurs à établir et à modifier l'ontologie.
- (3) Le moteur d'inférence : c'est le module de raisonnement du système, un moteur de règles, qui lit les faits existants et les règles créées par les ingénieurs du savoir et infère de nouveaux faits dans ce système; par exemple, le moteur JESS dans cette ontologie, un moteur de raisonnement (Pellet), qui peut vérifier la cohérence de l'ontologie développée pour éliminer les erreurs, est également un élément important.
- (4) L'interface de la requête, qui est souvent utilisée pour interagir avec le système de gestion des connaissances.

La collaboration de ces quatre composants permet à l'ensemble du système d'ontologie de fonctionner de manière transparente et efficace.

Un flux du travail de MATO (Figure.3.22) est fourni comme suit:

Premièrement, les ingénieurs de connaissances traduisent les connaissances capitalisées (voire section 3.2). Deuxièmement, l'ontologie et les règles SWRL définies via Protégé et SWRLTab sont stockées dans une base de connaissances. Ensuite, le moteur de règles exécute les règles SWRL et génère de nouveaux faits dans la gestion de système de l'ontologie. Enfin, le décideur peut obtenir des informations utiles en définissant plusieurs contraintes à partir de l'interface de requête SQWRL (Semantic Query-Enhanced Web Rule Language). Plus de détails sont illustrés dans le tableau 3.6.

### **3.4 Conclusion**

Dans ce chapitre, nous avons proposé un modèle de gestion des connaissances aux entreprises industrielles, ce modèle contient deux processus de gestion, le premier est la capitalisation des connaissances, basé sur une tâche industrielle, dont l'objectif est d'améliorer la performance du système de la production et de capitaliser le savoir-faire de l'entreprise. Le deuxième processus sur le partage des connaissances basé sur l'Ontologie des Tâches Manufacturières.



**Tableau 3.6** Les composants inclus Processus de partage des connaissances par l'ontologie MATO

<b>•Éditeur d'ontologie</b>	Protégé a été adopté pour l'ontologie car il s'agit d'un logiciel open source. Il permet à l'ontologie d'être créée et mise à jour par les utilisateurs et il est compatible avec la plupart des validateurs de syntaxe OWL. De plus, divers plugins peuvent être utilisés en combinaison avec Protégé.
<b>Langage de development</b>	OWL : pour modéliser les connaissances de domaine et de la tâche. SWRL: pour modéliser les connaissances d'inférence et aussi les requêtes SQWRL.
<b>Moteur de règles</b>	Le moteur Jess est un outil essentiel pour le développement de règles qui peuvent être appliquées à plusieurs reprises à un ensemble de faits ou exécutées pour créer de nouveaux faits.
<b>Raisonnement d'ontologie</b>	Pellet est un raisonneur vital dans l'ontologie. Il peut fournir des normes importantes et des services avancés de raisonnement.
<b>Plug-ins</b>	SWRL Tab: pour aider à écrire des règles SWRL. SWRL Editor: pour aider à modifier et à enregistrer diverses règles SWRL. SWRL Jess Bridge: pour faciliter la communication entre l'ontologie et les moteurs de règles. SQWRLQueryTab: pour offrir une interface graphique à travers laquelle les utilisateurs peuvent interagir avec les requêtes SQWRL. SWRLJessTab: pour fournir une interface graphique pour fonctionner avec SWRLJessBridge.

Dans le prochain chapitre on présente la mise en application de notre système de gestion des connaissances proposé.

# ***Chapitre 4 : Application du processus de capitalisation et de partage de connaissances au niveau de la centrale thermique***

Sommaire du chapitre 4:

4.1	Introduction.....	107
4.2	Description du domaine d'application et de la tâche industrielle.....	108
4.3	Application du KCMMT dans une tâche industrielle.....	109
4.3.1	Phase d'externalisation des connaissances.....	109
4.3.2	Phase de modélisation.....	109
4.4	Application de l'ontologie MATO dans une tâche industrielle.....	123
4.4.1	Exemple d'application : Diagnostic du système de refroidissement par hydrogène du turboalternateur.....	124
4.4.2	Phase de la modélisation du système.....	128
4.4.2	Phase du raisonnement du système.....	131
4.4.3	Phase d'implémentation.....	137
4.4.4	Sélection des connaissances de l'ontologie MATO par les requêtes SQWRL pour Diagnostic du système d'alarme.....	140
4.5	Transfert des règles SWRL génériques en OWL vers langage RDF.....	143
4.6	Conception des requêtes SPARQL à partir des requêtes SQWRL.....	144
4.7	Implémentation de l'ontologie MATO pour Diagnostic du système d'alarme.....	148
4.8	Architecture de l'application développée.....	149
4.9	Evaluation de l'ontologie MATO .....	153
4.9.1	Evaluation par la validation automatique.....	153
4.9.2	Validation basée sur les métriques.....	157
4.9.3	Validation basée sur les connaissances d'expert.....	159
4.10	Conclusion.....	161



## **4.1 Introduction**

Ce chapitre est une application de notre contribution proposée dans le chapitre 3 dans une centrale thermique, cette application est composée de deux modèles, le premier est la capitalisation des connaissances KCMMT et le deuxième est le partage des connaissances par l'ontologie MATO dans une tâche industrielle. L'application est sur la tâche de diagnostic de système d'alarme. Ce système affiche des messages d'alarmes par des panneaux de signalisation (système de surveillance en ligne).

L'intervention dans des problèmes liés à des systèmes industriels complexes nécessite un système intelligent contenant un niveau élevé de connaissances, tacites et explicites, dans le but de minimiser le temps d'intervention et d'assurer le fonctionnement d'un système de production lors de l'exécution des tâches.

Dans la littérature, il existe de nombreuses approches en matière de systèmes électriques de diagnostics présentés par (Ferreira et al, 2016), une enquête sur les techniques intelligentes appliquées à l'analyse et au diagnostic des systèmes électriques. Elles ont classé la quatrième approche pour le système de connaissances basé sur les erreurs, logique floue, axés sur les données, techniques d'optimisation et systèmes hybrides. De plus, nous utilisons une technologie sémantique pour les défauts de diagnostic appliqués dans la production de centrales. Pour la suite des travaux de (Titah et al, 2014), il est proposé un système expert d'aide au diagnostic dans un système d'alarme. Pour l'acquisition de connaissances, la méthodologie CommonKADS a été utilisée et le modèle de connaissances en création afin d'assurer la bonne continuité d'un système de production. Ils représentaient la connaissance par une approche logique et procédurale et étaient implémentés dans un générateur de systèmes experts G2. Mais, nous avons un problème de capitalisation des connaissances, de protection du capital intellectuel de l'expert, de représentation des connaissances en termes de valeur, ce sont des règles «si... alors...». dans ce cas, les connaissances restent statiques et il existe un problème de partage et de réutilisation des connaissances. Pour améliorer la capitalisation des connaissances en acquisition de termes et modélisation des connaissances (Titah et al 2017), a proposé un modèle de connaissances pour la tâche de fabrication reposant sur une représentation hybride des connaissances, logique par "expert de la base de connaissances" et "représentation ontologique" sémantique. Le système était conçu

pour la réutilisation des connaissances tacites pour le diagnostic dans une centrale thermique. Mais nous avons des problèmes de partage de connaissances à l'intérieur et à l'extérieur des entreprises.

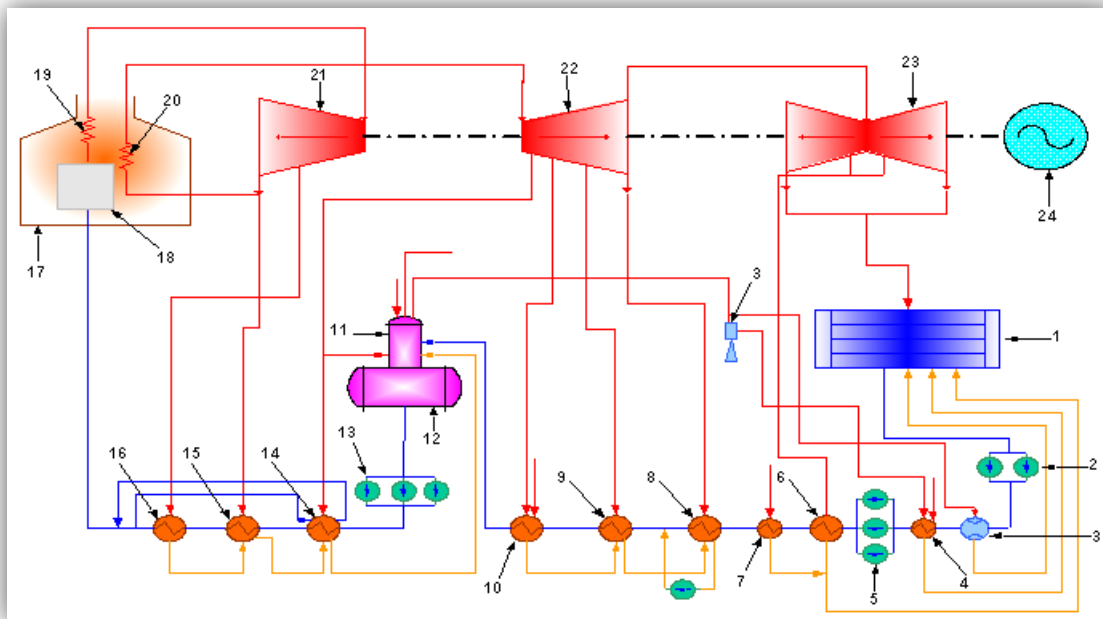
- Absence de concept commun aux problèmes de fabrication
- Difficulté à choisir l'expert spécialisé dans le domaine pour résoudre le problème.
- Manque d'équipement spécialisé.
- Problème de partage des connaissances entre experts dû à l'absence de terminologie commune aux agents industriels.

Le reste de ce chapitre est organisé comme suit : d'abord, nous allons définir le domaine d'application et de la tâche industrielle choisie, l'application du modèle de capitalisation, l'application de modèle de partage. Ensuite, l'interrogation de l'ontologie par les requêtes SQWRL et les requêtes SPARQL. Aussi une application web pour le diagnostic du système d'alarme, Enfin, l'évaluation de notre ontologie.

## **4.2 Description du domaine d'application**

La centrale thermique de Jijel est l'une des plus importantes centrales de l'Algérie, elle participe à 16% de la production globale. Elle est de type thermique, située au bord de la mer et Elle comprend trois groupes du type turbine à vapeur, dont la puissance unitaire est de 210 MW. La centrale thermique de Jijel a été construite par une entreprise russe TPE (Techno Prom Export). Les travaux ont été entamés en 1984, le premier groupe a été couplé au réseau en date du 06 juin 1992 par groupe de travail russe et algérien (experts, ingénieurs, techniciens). Les dates de couplage des groupes 2 et 3 sont respectivement le 1er juin 1993 et le 07 février 1994. Pour commander et contrôler le fonctionnement des équipements relatifs aux trois groupes, la Centrale dispose d'une salle de contrôle située dans le bâtiment de commande. Un schéma simplifié du processus de la centrale thermique est présenté dans la figure 4.1.

Les principales données relatives au fonctionnement de la tranche sont centralisées dans la salle de commande. Les opérateurs veillent au bon fonctionnement de l'installation et des équipements. Toutes les anomalies sont affichées par des alarmes dans les panneaux de signalisation (système de surveillance en ligne), à la salle de commande. Un diagramme d'activité du service quart de production présente la méthode de surveillance et élimination des alarmes, voire figure 4.2.



(1) : Condenseur	(9) : Réchauffeur basse pression 3(RBP3)	(18) : Economiseur
(2) : Pompe d'extraction 1 <sup>ier</sup> étage	(10) : Réchauffeur basse pression 4 (RBP4)	(19) : Surchauffeur
(3) : Ejecteur	(11) : Dégazeur	(20) : Resurchauffeur
(4) : Condenseur de bouillie avec éjecteur	(12) : Bûche alimentaire	(21) : Corps haute pression
(5) : Pompe d'extraction 2 <sup>ème</sup> étage	(14) : Réchauffeur haute pression 1 (RHP1)	(22) : Corps basse pression
(6) : Réchauffeur basse pression 1(RBP1)	(15) : Réchauffeur haute pression 2 (RHP2)	(23) : Corps moyenne pression
(7) : Condenseur de bouillie sans éjecteur	(16) : Réchauffeur haute pression 3 (RHP3)	
(8) : Réchauffeur basse pression 2 (RBP2)	(17) : Générateur de vapeur	

Figure 4.1. Schéma simplifié de processus de production de la centrale

### 4.3 Application du KCMMT dans une tâche industrielle

#### 4.3.1 Phase d'externalisation des connaissances

*Étape 1 : Identification des problèmes par la méthodologie CommonKADS*

*Étape 1.1: Modèle d'organisation*

Ce modèle est construit pour identifier les problèmes, le contexte organisationnel et les possibilités de solutions sont basés sur les systèmes de connaissances. Les problèmes et les solutions proposés sont présentés dans le (tableau 4.1). Au cours de la prochaine décennie, les experts prendront leur retraite, emportant avec eux les savoirs, les savoir-faire (connaissance tacites) et autre expertises indispensables à la primauté de notre entreprise, le détail du modèle d'organisation est construit à l'aide des trois tableaux: MO-2, MO-3, et MO-4 dans les tableaux de l'annexe A.

**Chapitre 4 : Application du processus de capitalisation et de partage de connaissances au niveau de la centrale thermique**

Tableau 4.1. Modèle d'organisation de CommonKADS (Titah et al., 2017)

<b>Problèmes et opportunités</b>	<ul style="list-style-type: none"> <li>• <b>Problème de partage des connaissances entre les experts et les nouveaux ingénieurs.</b></li> <li>• <b>Perte d'expertise et de capital intellectuel</b></li> <li>• <b>Le délai d'intervention en cas d'une anomalie est très lent, qui engendre un déclenchement de système. .</b></li> </ul>
<b>Contexte organisationnel</b>	<ul style="list-style-type: none"> <li>• Assure la bonne continuité de production et la disponibilité des équipements de production.</li> <li>• Assurer le suivi des paramètres économiques et améliorer les performances des équipements.</li> </ul>
<b>Solutions</b>	<ul style="list-style-type: none"> <li>• Externalisation des connaissances tacites en connaissances explicites.</li> <li>• Assurer l'acquisition des connaissances, la capitalisation des connaissances et le partage des connaissances dans l'entreprise.</li> <li>• Création d'un système à base des connaissances</li> </ul>

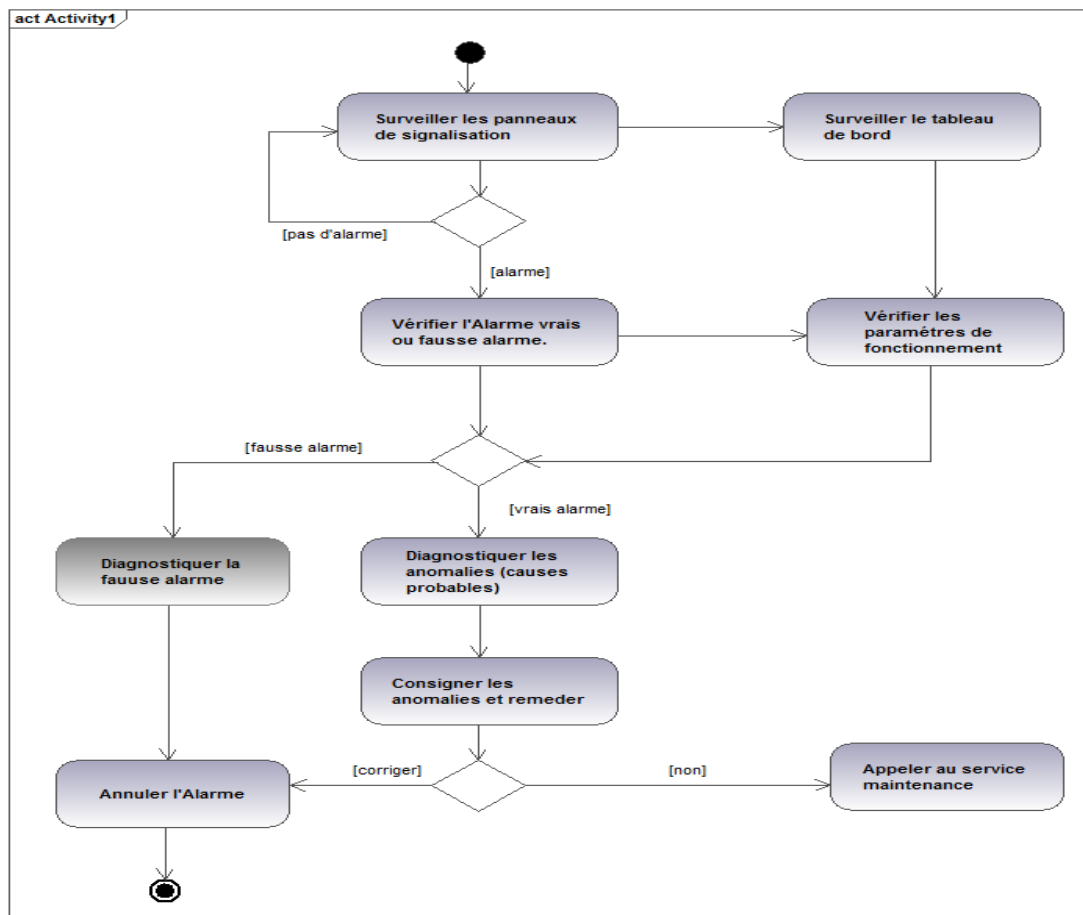


Figure 4.2. Diagramme d'activité du service quart de production

Étape 1.2: Modèle de tâche

Le modèle de tâches permet d'étudier les sous-processus d'affaires. Il analyse l'organisation globale des tâches, les entrées et les sorties, les pré-conditions et les critères de performance, ainsi que les ressources et compétences nécessaires.

**Chapitre 4 : Application du processus de capitalisation et de partage de connaissances au niveau de la centrale thermique**

Ce raffinement est effectué à l'aide du tableau Modèle de Tâches-1 (MT-1). Le détail de chaque élément de connaissance associé à une tâche est obtenu au tableau MT-2 (tableaux de tâche de l'annexe A).

Tableau 4.2. Modèle de tâche de CommonKADS

Modèle de tâche	Feuille de travail d'analyse du modèle de tâche TM-1	
Tâche	Voir MO-3	Diagnostiquer les alarmes de signalisation
Organisation	Voir MO-2	Division exploitation : Service quart, la salle de commande par les cadres d'exploitations
But et valeur		Aide aux personnels d'exploitation pour leur permettre de réaliser le diagnostic d'état de leurs matériels. Il disposera pour cela d'outils performants et attrayants le déchargeant en outre du travail routinier.
Dépendance et flux	Les tâches précédentes  Les tâches suivantes	Vérification des paramètres de fonctionnement.  Consigne les anomalies de fonctionnement et les mesures prises pour y remédier
Objets traités	Objets en entrée  Objets en sortie  Objets internes	Objets d'entrée : alarme de signalisation.  Objets de sortie : identification de la cause probable par l'expert.
Temps et contrôle	Fréquence, Durée  Contrôle  Contraintes & Conditions	La tâche la plus importante est l'identification de la cause de signalisation, le temps d'intervention avant le déclenchement ne dépasse plus les 5 minutes.
Agents	MO-2: Gens, Systèmes  Ressources;  MO-3: effectuée par	Les personnels qui sont responsables de ces tâches :  Les cadres d'exploitation (anciens exploitants) et les ingénieurs de production.
Connaissances et compétences	Voir MO-4	Compétences nécessaires pour bien effectuer la tâche. Pour les éléments de connaissances dans le tableau  MT-2
Qualité et Performance	Mesures	Le but de la tâche est d'éliminer la cause de problème le plus vite possible avant le déclenchement du système.

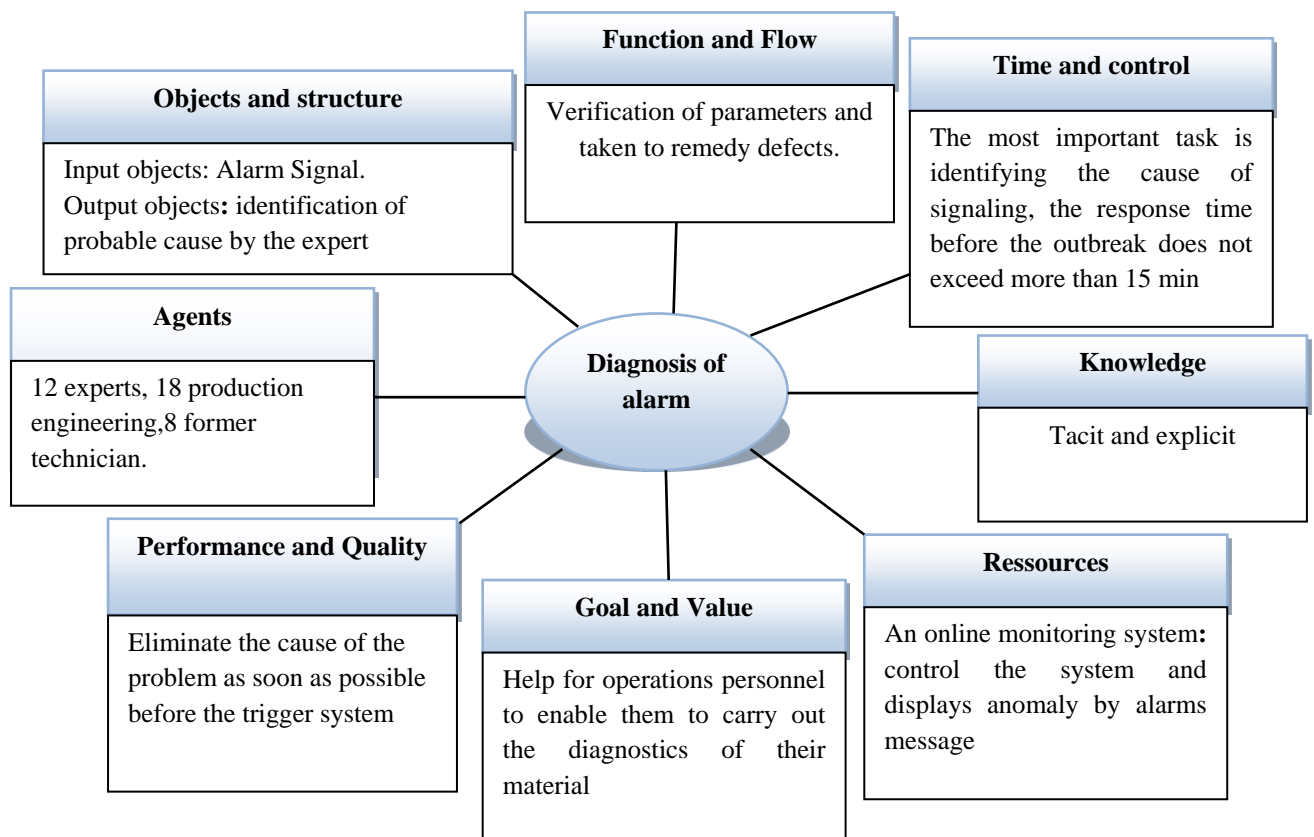


Figure 4.3. Modèle de tâche (Titah et al., 2017)

Étape 1.3: Modèle d'Agent

Le modèle d'agents décrit les caractéristiques des agents qui peuvent être soit des humains, soit des systèmes d'information, soit d'autres entités capables d'effectuer une tâche. La figure 4.3 présente les rôles des agents manufacturiers par un diagramme de cas d'utilisation de service quart de production. Le modèle décrit leurs compétences, leur autorité et leurs contraintes ainsi que les liens de communication. Ce modèle est construit à l'aide du tableau Modèle de l'agent-1 (MA-1) (Tableau 4.1).

Étape 2 : Analyse des problèmes par diagramme de Pareto

Le modèle d'organisation est d'identifier des problèmes d'une façon générale. Il nécessite aussi une connaissance détaillée sur le processus, nous avons présenté un modèle qui permet de le connaître, de le comprendre et de l'interpréter, de détailler l'analyse d'un processus afin de parvenir à une compréhension efficace du problème. Pour analyser les problèmes dans une situation réelle toute défaillance du processus industriel, qui contient 23 composants, affiché sous forme de message d'alarme, le

diagramme de Pareto (Figure 4.11), il représente 80% des alarmes sur les composants critiques du système, ce qui nécessite une surveillance du système et 20% des éléments ne sont pas critiques mais nécessitent une expertise et un temps d'intervention. Notre objectif est présenté par le modèle d'organisation (tableau 4.4).

Tableau 4.3. Modèle d'Agent de CommonKADS

Modèle d'agent	Feuille de travail de modèle d'agentAM-1
Nom	Cadre d'exploitation
Organisation	Division exploitation, service Quart, salle de commande.
Impliqué dans	<ul style="list-style-type: none"> <li>• diagnostique les anomalies de fonctionnement.</li> <li>• contrôler les équipements de production</li> <li>• superviser les opérations de démarrage et d'arrêt des installations.</li> <li>• Consigne les anomalies de fonctionnement et le remède.</li> </ul>
Communique avec	<ul style="list-style-type: none"> <li>• Le chef de quart, les ingénieurs de productions, les TPCC, TCA.</li> <li>• Le logiciel GMAO : service maintenance.</li> </ul>
Connaissance	<ul style="list-style-type: none"> <li>• Les connaissances tacites et explicites pour la surveillance de fonctionnement des installations et maintenir la production demandée sur la base des paramètres affichés (les panneaux de signalisation).</li> </ul>
Autres compétences	<ul style="list-style-type: none"> <li>• Participer lors des arrêts ou dans le cadre de la rotation de quart à des travaux d'entretien sur le matériel.</li> <li>• Etudier les rapports d'exploitation quotidiens et analyse les dysfonctionnement.</li> <li>• Proposer des consignes d'exploitation de fonctionnement.</li> </ul>
Responsabilités et contraintes	<ul style="list-style-type: none"> <li>• Gérer les équipements de mesure de rendement.</li> <li>• Responsable d'arrêt de groupe en cas d'un problème.</li> </ul>

Étape 3 : diagnostiquer les problèmes analysés

Étape 3.1: Brainstorming

D'après la construction du modèle d'organisation, de tâche et d'agent, une séance de brainstorming avec les experts défini par le modèle d'agent, un outil qui aide à l'identification des causes des problèmes, en tenant compte de la connaissance des experts sur les problèmes cités dans le modèle d'organisation. Il vise à générer des idées / suggestions avec un groupe d'agents industriels exécuteurs des tâches

(ingénieur, techniciens, opérateurs....) susceptibles de fournir des avancées significatives dans la résolution de problèmes.

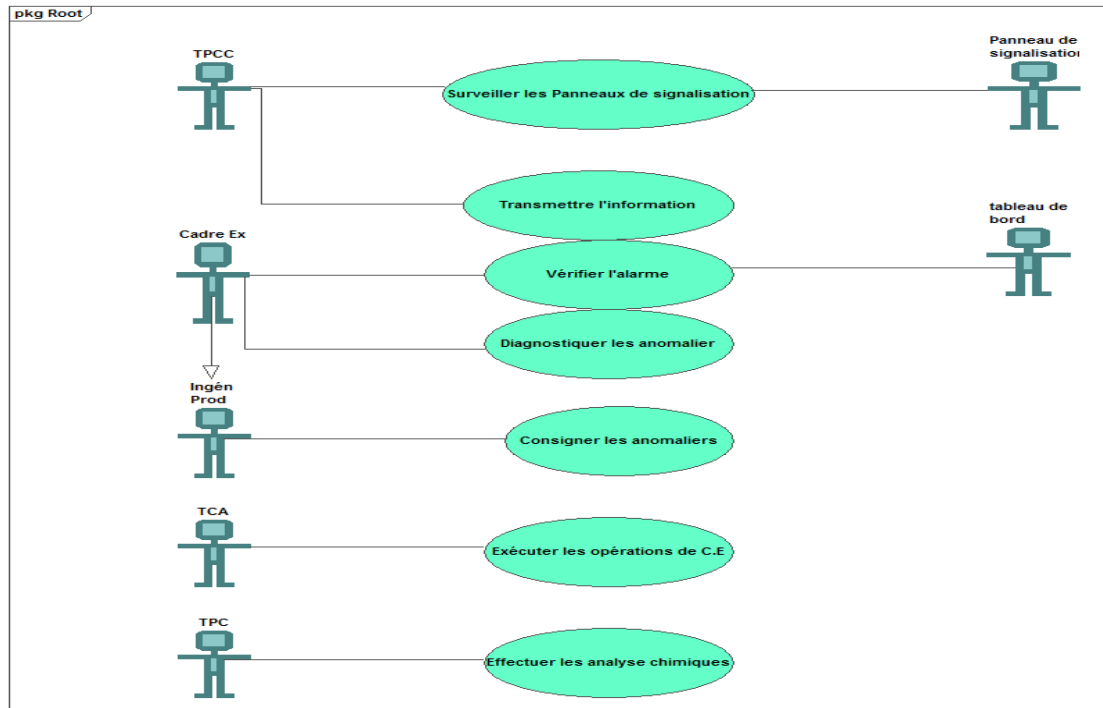


Figure 4.4. Diagramme d'activité du service quart de production

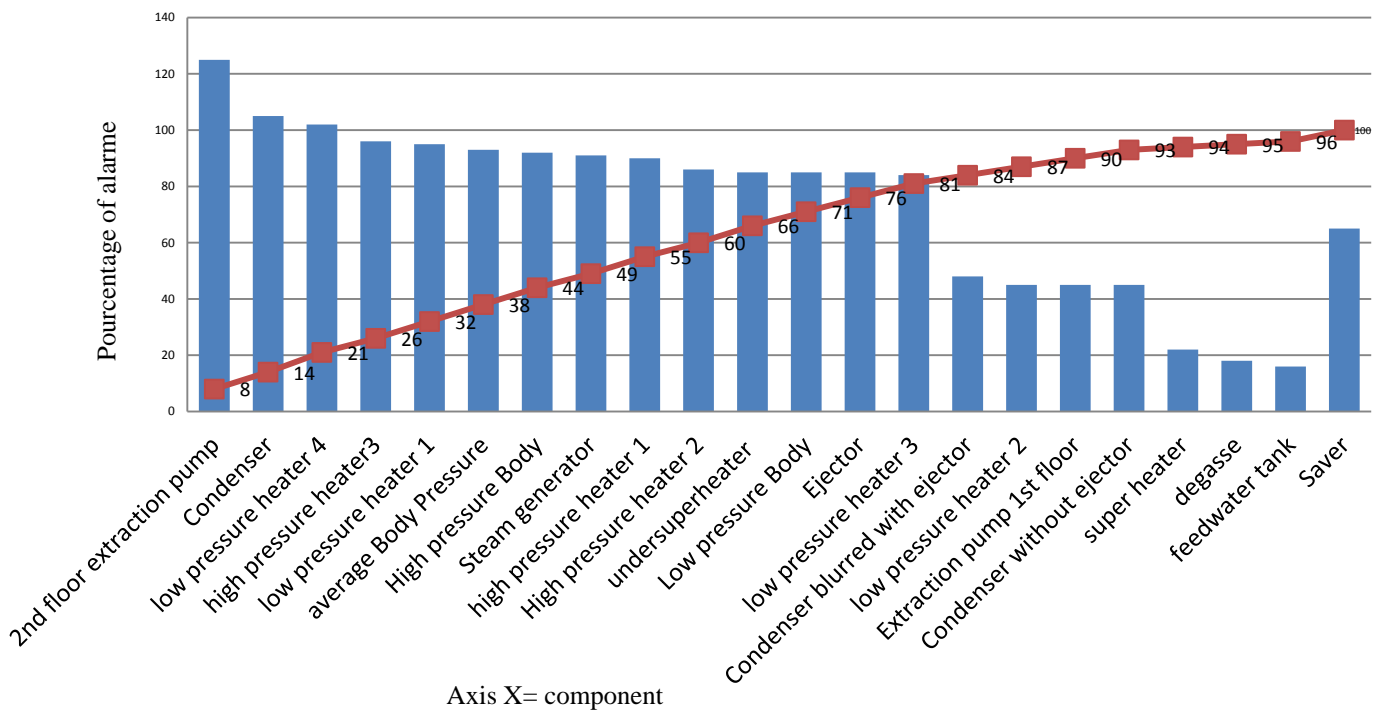


Figure 4.5. Diagramme de Pareto-pourcentage d'alarme de chaque composant (Titah et al.,2017)



Étape 3.2: l'interview individuelle non- structurée

Avant cette étape, nous avons présenté le modèle de tâche et le modèle d'agent (Figure 4.2 et Figure 4.3) pour faciliter la rencontre avec les exécuteurs experts collectifs et individuels des tâches. Un entretien individuel non structuré après une séance de brainstorming a été tenue. En raison des limites de la technique de l'entretien non structuré, nous avons discuté du problème et des opportunités de l'organisation et du problème de capitalisation des connaissances, le modèle d'organisation a présenté une solution basée sur la connaissance. Cette réunion a présenté la manière de cartographier et de faire des techniques d'élicitation.

Tableau .4.4 Temps pour l'acquisition des connaissances lors des entretiens avec les experts. ( Titah et al.,2017).

Experts	Années de travail	étape 2	étape 3	étape 4	Total (h)
<b>A</b>	26	3h 25min	5h 12min	10h 20min	18h 57min
<b>B</b>	25	4h 50min	7h 10min	11h30min	23h 30min
<b>Total</b>		8h 15min	12h 22min	21h 50 min	32h 27 min

Étape 4 : appliquer les techniques de MACTAK

Étape 4.1: Application de la technique de classification des concepts

Nous avons utilisés la méthode de classification des concepts pour identifier et organiser les concepts et leurs relations dans la centrale thermique, comme la classification des problèmes, des taches , des équipements , des ressources humains et aussi les relations entre ces concepts selon le point de vue de l'expert.

Ensuite nous avons présentés des cartes avec le nom des concepts à l'expert, qui est chargé de trier ces cartes qui sont similaires ou du même type.

L'expert doit les regrouper à nouveau et supprimer certains concepts synonymes ou non pertinents.

Cette technique est répété plusieurs fois et les résultats de la classification des concepts est un groupe de composants qui partagent des attributs communs. Afin de faciliter la tâche de communication avec les experts du domaine.

Étape 4.2: Application la technique d'information limitée

Nous avons présenté les problèmes du système d'alarme à l'expert, en s'attendant à ce qu'il propose méthode pour identifier les causes.

L'expert révèle ainsi le niveau minimum d'informations dont il a besoin pour éliminer l'alarme. Nous présentons plusieurs photos liées à des problèmes avec lesquels l'expert a l'habitude de travailler, mais avec une restriction de temps beaucoup plus importante.

Après cela, il lui est demandé d'extérioriser tout ce dont il se souvient de ces images et de fournir leur interprétation. Les experts ont été amenés à répondre à chacune des raisons lors de l'entretien sur les causes des alarmes en utilisant les concepts proposés par les experts du domaine. Des photos de défauts ont également été utilisées à cette époque, la visualisation des images était limitée à 12 s chacune.

Étape 4.3: Application la technique d'analyse de protocole

Nous avons utilisé une technique d'analyse de protocole pour déterminer la structure ou la relation de l'effet de causes multiples en observant systématiquement le processus. Après les interviews ayant été menées individuellement avec les experts, Une analyse du contenu dans une manière systématique et objective.

L'analyse consistait à vérifier l'existence d'une similarité ou d'un schéma de réponses entre les propositions des experts, nous avons classifié les phrases ou leurs parties selon le type de connaissances qu'elles témoignent: concepts, règles, procédures.

Il est difficile de résoudre des problèmes complexes sans considérer une telle structure, qui consiste en une chaîne de causes et d'effets, donc le choix du diagramme d'Ichikawa est une méthode simple et facile pour la représentation du phénomène concernant les causes possibles des alarmes.

Étape 5 : Modèle explicite sous formes de règles de production

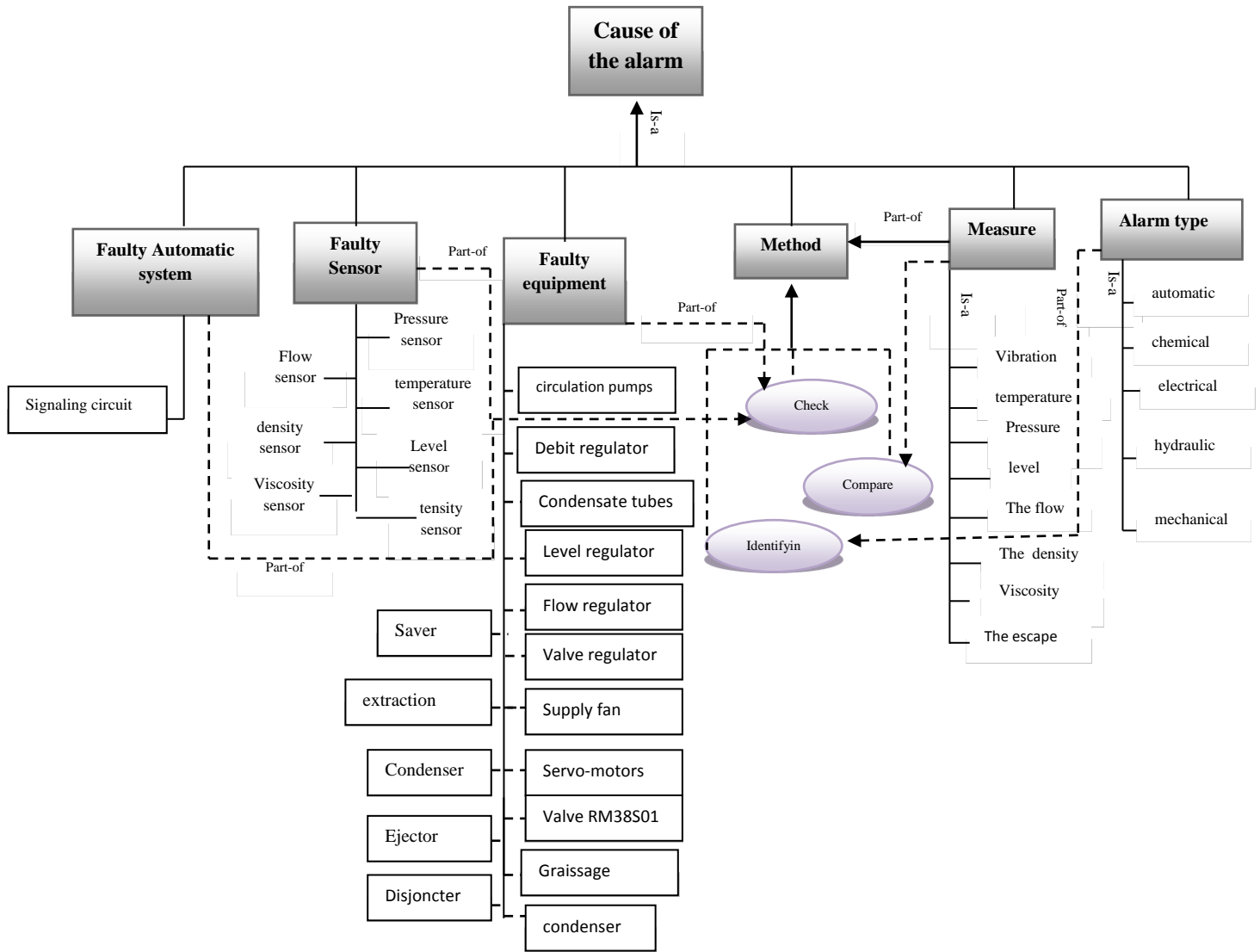


Figure 4.6 Réseaux sémantique pour les causes possible d’alarme (Titah et al.,2017)

Étape 5.1: Préparation du réseau sémantique par diagramme d’Ishikawa

D’après l’analyse du protocole nous avons élaboré les connaissances par des réseaux sémantiques représentant la cause du message d’alarme présenté (Figure 4.5). Les entretiens individuels avec l’expert et le temps d’acquisition des connaissances à l’étape 3 et à l’étape 4 sont présentés dans le tableau 4.4. Après l’aide des experts,

nous nous sommes concentrés sur: le type d'alarme, la mesure, la méthode, l'équipement défectueux, le capteur défectueux et le système automatique défectueux.

Cette représentation des connaissances facilite les règles de production et l'externalisation des connaissances. Dans cette étape, le réseau sémantique a été utilisé comme moyen de représenter les connaissances explicites d'une manière systématique pour faciliter la construction des règles de production.

#### Étape 5.2: Créer des règles de production

La création de règles de production est l'étape importante du processus d'externalisation, Après le développement des réseaux sémantiques, la représentation des connaissances, à travers des règles de production, est très facile pour encoder les connaissances. L'avantage des règles de production extraites par les réseaux sémantiques (Figure 4.5.), développées à l'étape 5.1, permettent de diagnostiquer le système de fabrication et concernant toutes les différentes causes possibles d'alarme. Nous avons introduit un système de refroidissement du turbo-alternateur par l'hydrogène, un système très compliqué qui nécessite beaucoup de connaissances et un temps d'intervention très important.

Toutes les pannes et perturbations affichées sous forme de message d'alarme, telles que: basse pression et pureté dans le corps du turbo-alternateur, apparition de liquide ou d'eau dans le corps de l'alternateur. Après les réseaux sémantiques, les problèmes causés par un équipement défectueux, un capteur défectueux et un système automatique défectueux. Les mesures des paramètres sont: la pression d'hydrogène, la pureté de l'hydrogène, la concentration d'hydrogène, le débit d'eau de refroidissement. Les méthodes utilisées par les experts sont: vérifier l'alarme (vraie ou fausse), diagnostiquer la fausse alarme, identifier la véritable cause de l'alarme et enfin donner une solution au problème. Les règles de production sont illustrées à la Figure.4.6.

#### Étape 5.3 : Valider les règles par la méthode d'observation

Les techniques d'observation a était requis pour évaluées les règles de production (voir section 4.10) Grâce à cette étape, l'expert propose la solution du problème pour observe le processus. L'expert donc plus aise dans son environnement de travail afin d'effectuer les tâches quotidiennes.

L'observation est utilisée pour identifier et vérifier les connaissances externalisées sous forme de concepts, de procédures, d'hypothèses ou de modèles proposés par les experts du domaine pour résoudre un problème automatique.

Low pressure and purity of hydrogen in the body of turbo-alternator		The appearance of liquid in the alternator body	
<p><b>VERIFICATION: true or false alarm</b></p> <p>IF Pressure of Hydrogen = 0.2 bar.</p> <p>AND the purity of hydrogen = 97%</p> <p>THEN alarm is false</p> <p>Caused-by: automatic fault system</p> <p>OR Faulty sensor of purity and pressure.</p> <p>ELSE</p> <p>of Pressure Hydrogen &lt; 0.2 bar</p> <p>AND the purity of hydrogen &lt; 97%.</p> <p>THEN the alarm is true.</p>	<p><b>IDENTIFICATION alarm cause</b></p> <p>IF alarm is true.</p> <p>AND high hydrogen flow for backup power.</p> <p>AND the hydrogen content in the bearing housings &lt; 0.9.</p> <p>THEN perturbation of gas-tightness in turbo-alternator .</p> <p>OR High oil flow to the interior of the turbo-alternator.</p>	<p><b>VERIFICATION: true or false alarm</b></p> <p>IF Pressure of Hydrogen = 0.2 bar.</p> <p>AND humidity of Hydrogen =10g/m<sup>3</sup>.</p> <p>AND the purity of hydrogen=97%</p> <p>THEN alarm is false</p> <p>Caused-by: automatic fault system</p> <p>OR Faulty sensor of humidity , purity and pressure.</p> <p>ELSE</p> <p>Pressure of Hydrogen &gt;= 0.2 bar</p> <p>AND humidity of Hydrogen =10g/m<sup>3</sup>.</p>	<p><b>IDENTIFICATION alarm cause</b></p> <p>IF alarm is true.</p> <p>AND amount of water to increase discharged from the vaporizer of the refrigerating machine &gt;05 kg.</p> <p>AND flow of water through the gas cooler &gt;250 m<sup>3</sup>/h</p> <p>AND pressure of water &gt; pressure of hydrogen.</p> <p>THEN Water leaking gas</p>
<p><b>Procedural Solution</b></p> <ul style="list-style-type: none"> <li>• Check valves closing seal on the gas panel. in drainages And hydraulic lock.</li> <li>• Checking the operation of the pressure regulator. Check for leaks in the return line in sealing oil.</li> </ul>		<p><b>Procedural Solution</b></p> <p>Disconnect the alternator detect and liquidate the default. Define type of alarm and Call the maintenance department.</p>	

Figure.4.7.Règles de production pour le système de refroidissement turbo-alternateur (Titah et al.,2017)

### 4.3.2 Phase de modélisation

Suit à viens la phase d'externalisation des connaissance tacites en connaissances explicites, la modélisation des connaissances dans cette phase, il existe trois catégories dans le modèle de connaissance de CommonKADS: la connaissance des tâches, la connaissance du domaine et la connaissance de l'inférence. deux types de modélisation des connaissances différents on était utilisé: 1) la modélisation par représentation d'ontologie pour la connaissance des tâches et la connaissance du domaine. 2) la modélisation à base de connaissances d'experts pour la connaissance de l'inférence.

### 4.3.2.1. Connaissances du domaine

La modélisation des connaissances de domaine par ontologie, utilisant le Concept-instances-Attributes-relations est similaire dans le langage OWL aux classes-individus-propriétés de données-propriétés d'objet, comme le montre le tableau 4.5.

**Concept:** Un groupe d'objets avec des caractéristiques de partage dans OWL-Ontology défini par des classes, les principales classes des concepts: «Equipment», «AlarmSystem», «agent», «cause», «Solution», illustré dans la figure 4.7.

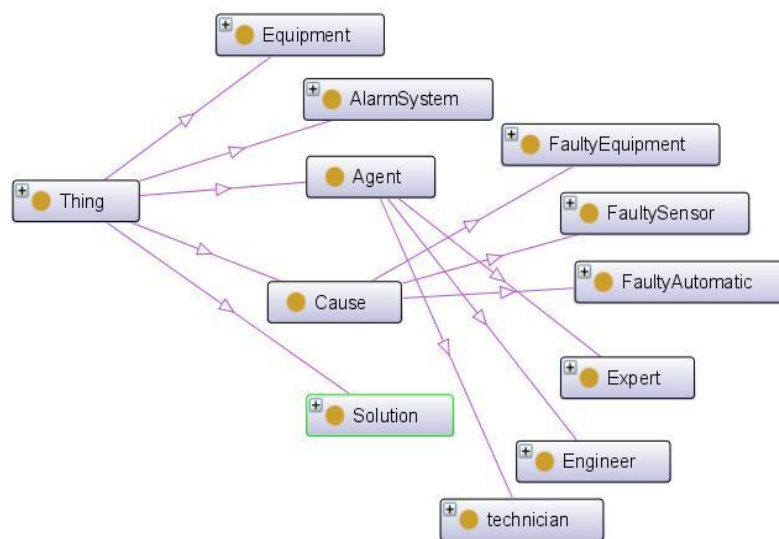


Figure. 4.8. Les concepts principaux des classes (Titah et al., 2017)

**L'instance :** définit l'individu de type concept, définit les instances des concepts 'mainEquipment' sont: 'Alternator', 'condenser', 'Ejector', 'degasser', 'saver', ..., alarm individual: al1, al2, al3, al4..., comme montré dans la Figure.4.8.

**Relations:** les relations dans la connaissance du domaine présentées par des propriétés d'objet, ces relations relient des classes ou un lien entre des individus, dans ce modèle nous avons proposé: 'caused-by', 'check-of', 'Control-of', 'has-individual', 'has-subclass', 'has-a', 'Identify-of', 'Propose-a', montré dans la Figure.4.9.

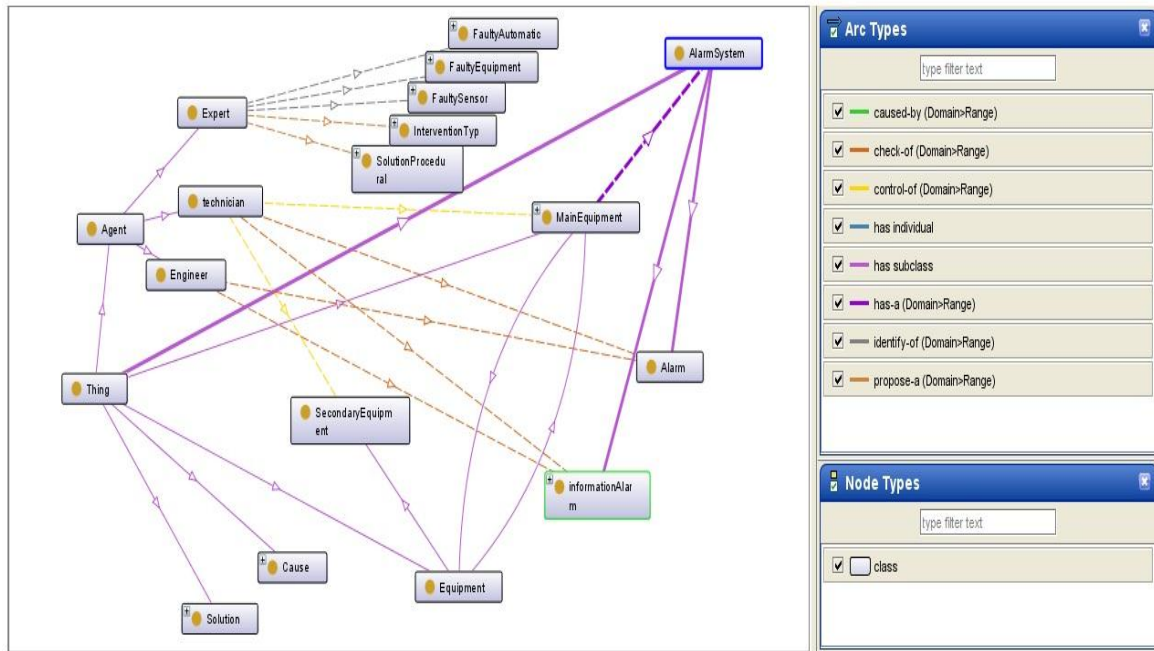


Figure. 4.9. Les instances de la class « MainEquipment » (Titah et al.,2017)

La relation 'has-subclass' lien entre la classe principale et la sous-classe comme exemple (la classe 'Agent' a la sous-classe 'expert', 'Engineer', 'technician'), 'has-individual' link class to individual ('mainEquipment' has- individus 'alernateur', 'dégazeur',...), 'Contrôle-de' lien sous-classe 'Technicien' à 'alarmSystem', 'provoqué par' lien 'alarmEquipment' à 'FaultyEquipment'.

'Identifier-de' lien 'Expert' à 'provoquer', 'Contrôle-de' lien 'Ingénieur' à 'équipement', 'vérification' de lien entre 'Technicien' et 'équipement principal', 'a-un' lien entre 'mainEquipment' et 'alarmSystem', toutes ces relations montrées dans la Figure 4.10.

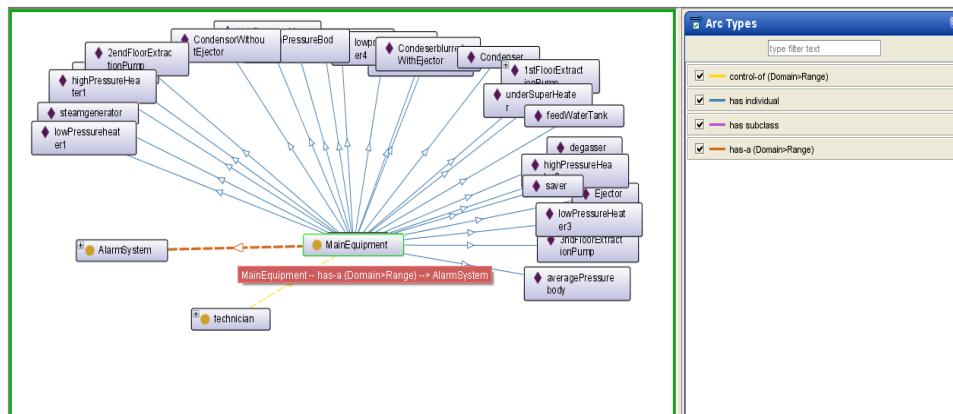


Figure. 4.10. Les relations entre les classes (Titah et al.,2017)

**Attributs:** les propriétés de données relient les individus à leurs valeurs de données, nous avons défini les propriétés de données comme: ‘agentID’, ‘alarmID’, ‘alarmType’, ‘CauseID’, ‘ComponentID’, ‘solutionID’, ‘temperature’, ‘pressure’, ‘pressureHydrogen’, ‘purityHydrogen’, ‘concentrationHydrogen’, ‘humidity’, ‘flow’, ‘level’, ‘density’, ‘vibration’, ‘viscosity’, ‘typeOfIntervention’, ‘espace’.

**4.3.3.2. Connaissances de la tâche**

Dans cette étape, nous avons utilisé la représentation d'ontologie, pour les rôles agents, la description de la tâche et les méthodes utilisées .

les rôles des agents sont définis par les propriétés des objets: ‘check-of’, ‘Control-of’, ‘Identify-of’, ‘Propose-a’.

Le diagnostic d’alarme est une tâche d'analyse, grâce à la décomposition fonctionnelle de la tâche en sous-tâches; nous avons trouvé que l'élément de la tâche de contrôle représente les propriétés des objets dans la connaissance du domaine (Figure 4.11.).

Tableau 4.5.Construction la connaissance du domaine . (Titah et al.,2017 )

Connaissance du domaine	Les composants CommonKADS	OWL Construction
	Concept	Classes
	Relation	Propriétés des objets
	Rules type	Propriétés des données
	instances	individues

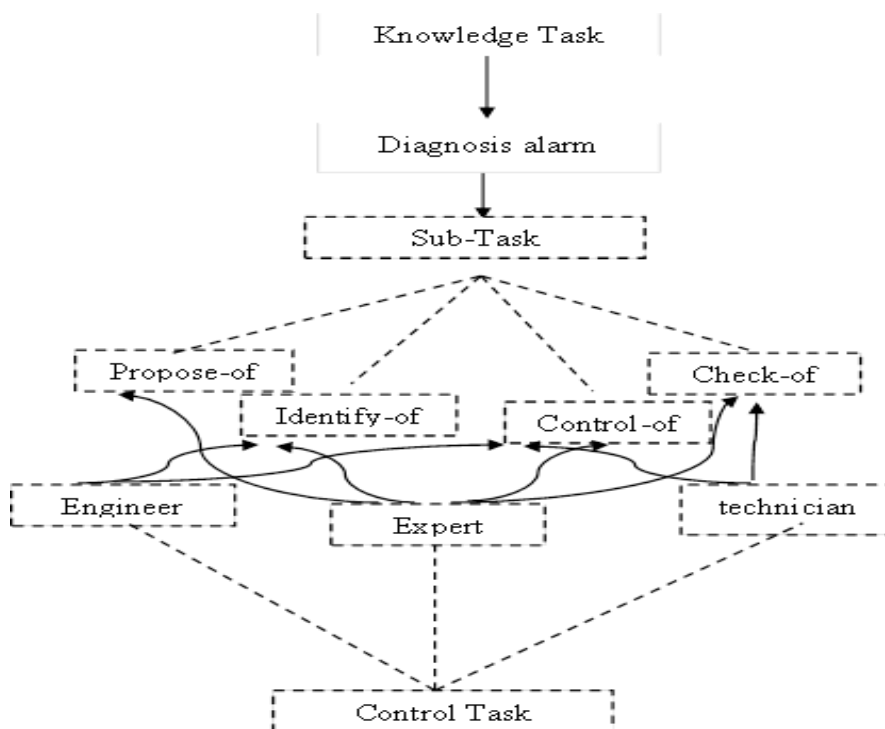


Figure.4.11 Description de la connaissance et contrôle de la tâche (Titah et al.,2017)



### 4.3.3.3 Connaissance d'inférence

La connaissance d'inférences proposées dans ce modèle est l'élément de contrôle de la tâche de connaissance, 'Check-of', 'Control-of', 'Identify-of', 'propose-of', ces éléments sont les propriétés objet du système d'alarme. Dans cette étape, nous avons utilisé un modèle logique défini par (If Then), nous avons proposé une inférence procédurale, illustrée à la figure 4.12.

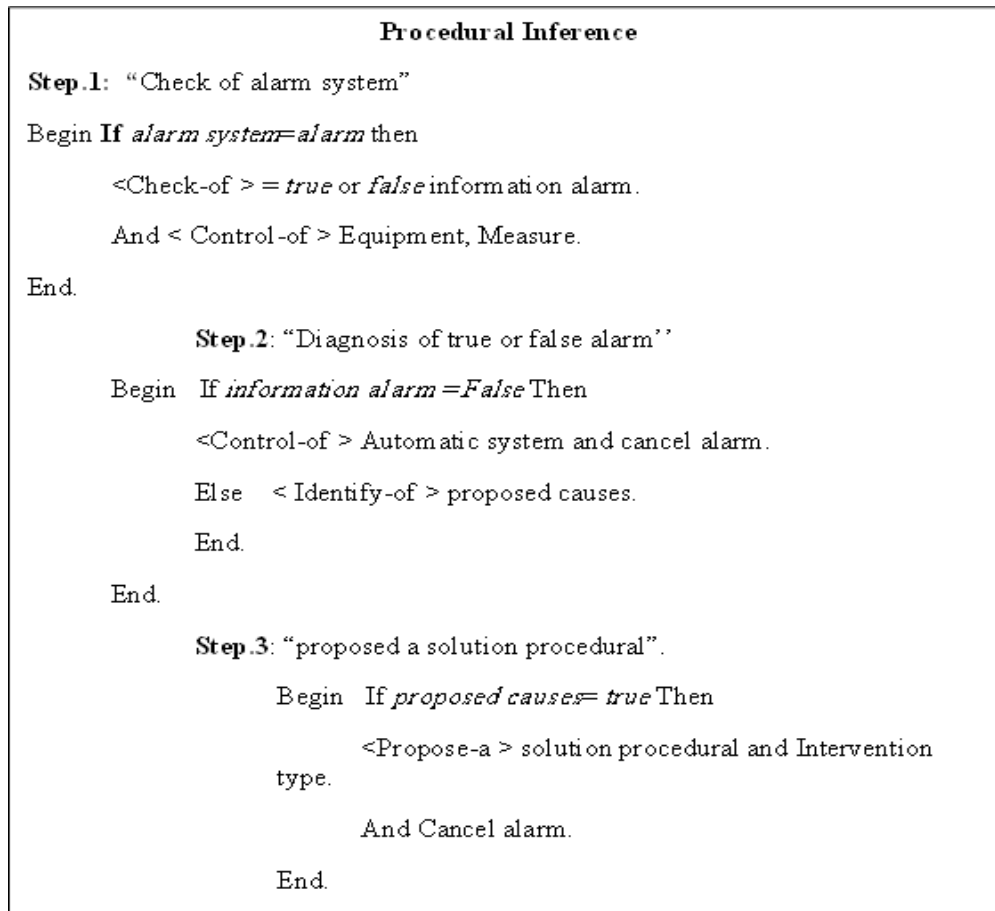


Figure 4.12. La procédure d'inférence (Titah et al., 2017)

## 4.4 Application de l'ontologie MATO dans une tâche industrielle

Dans le but d'améliorer le système d'intervention et la qualité des connaissances à utiliser, nous avons appliqué l'ontologie proposée MATO (Ontologie de la tâche de fabrication) qui compose les concepts principaux: les concepts principaux: le processus industriel, la tâche industrielle, l'agent industriel, le problème industriel, la solution industrielle, l'équipement industriel et les caractéristiques



#### **4.4.1.2 Système de Refroidissements par hydrogène**

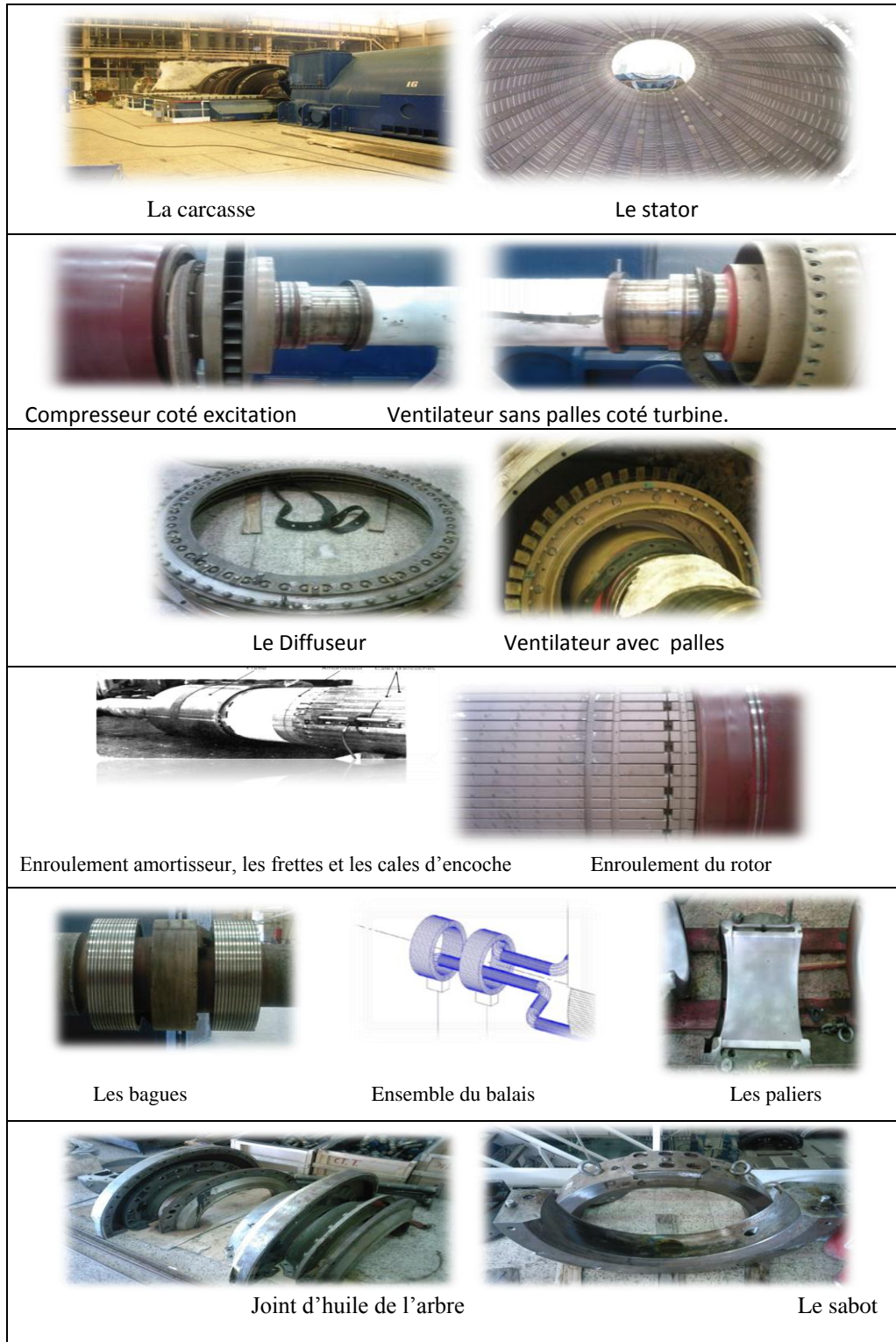
Le refroidissement de l'alternateur est du type mixte. L'enroulement statorique, les barres de liaison et les bornes de sortie sont refroidis à l'eau distillée (distillat), tandis que le noyau du stator et le rotor sont refroidies au gaz, (À l'hydrogène). Ce gaz a été choisi pour sa faible densité, 14 fois plus faible que celle de l'air, ce qui réduit dans le même rapport les pertes par frottement du fluide. Ainsi que pour ses propriétés thermiques, son coefficient de transmission thermique de surface à vitesse et pression égale étant de 50 % supérieur à celui de l'air. Sa conductivité thermique étant 7 fois plus forte.

Le danger de travailler avec l'hydrogène consiste dans l'éventualité de formation, à la suite de perturbation du régime d'utilisation du système de gaz, d'un mélange explosif de l'hydrogène avec de l'air dans le corps de l'alternateur.

Le mélange d'hydrogène et d'air est inflammable lors qu'il contient 4 à 75% d'hydrogène. Le mélange inflammable peut s'enflammer sous l'action d'un feu ouvert, d'un échauffement local à l'écoulement rapide (surtout en présence des produits de corrosion sur les parois des tubes et des appareils).

On a présenté les deux parties constitutives de l'alternateur qui sont le rotor et le stator, ainsi que les organes qu'elles englobent tels que les barres statoriques, la flasque, les paliers et les joints de paliers (Voir la figure 4.14).

Sachant que l'alternateur a besoin d'un refroidissement continu, les russes ont mis au point des systèmes de refroidissement à base d'hydrogène, d'eau distillée ou les deux en même temps comme utilisé par cette centrale thermique de Jijel, on a donc étudié les conditions nécessaires pour éviter le risque de mélange entre l'hydrogène et l'air, et aussi il faut que la pression de l'hydrogène soit supérieur à celle de l'eau distillée pour éviter le court-circuit dans l'alternateur.



**Figure 4.14.** Les deux parties constitutives de l'alternateur , le rotor et le stator, ainsi que les organes qu'elles englobent

#### 4.4.1.3 Condition et paramètre normal dans le système de refroidissement par hydrogène

Lors de fonctionnement normal de l'alternateur la surpression d'hydrogène dans le corps doit être maintenue conforme aux données de base indiquées au tableau en haut. La pression de gaz dans l'alternateur peut être maintenue à l'aide de dispositif d'alimentation d'appoint automatique en hydrogène ou bien, manuellement. Dans les deux cas, la pression dans le corps de l'alternateur ne doit s'écarter de celle nominale de plus +0,2 bar. La pureté de l'hydrogène dans le corps ne doit pas être inférieure à 97%. La teneur en oxygène ne devant excéder 1,2%. La teneur en oxygène maximale de l'hydrogène dans le système de vidange d'huile de la côte hydrogène des joints ne doit dépasser 2%. Le maintien des paramètres d'hydrogène optimaux (pression pureté et humidité) contribue à la diminution de la surchauffe des parties

Actives d'alternateur (enroulements rotorique et fer du rotor et au stator) et à l'amélioration de la transmission de la chaleur à l'hydrogène par la surface des parties actives, augmente la sécurité d'utilisation du système de refroidissement par gaz. Toutefois, au cours de service les fuites d'hydrogène du corps de l'alternateur se produisent car le système n'est pas parfaitement étanche. Une partie d'hydrogène du corps de l'alternateur est apportée avec l'huile d'étanchement.

Lorsqu'un problème surgit dans le système de refroidissement par hydrogène ou lorsque les paramètres dépassent les limites admissibles (tableau 4.6) à cause des perturbations des paramètres du milieu réfrigérant.

**Tableau 4.6** Les conditions normales et transitoires dans le système de refroidissement par Hydrogène

Paramètres à contrôler	Seuil de signalisation	
	A la baisse	A la hausse
Pression d'hydrogène dans le corps	3-0.2 bar	3+0.2 bar
Pureté d'hydrogène	Mini : 97%	plus
Humidité d'hydrogène	10 g/m <sup>3</sup>	-
Température d'hydrogène froid	40 °C	40 °C
Température d'hydrogène Chaud	-	Maxi : 75 °C
Teneur en hydrogène dans les carters de paliers, la gaine blindée	-	Maxi : 1%

L'apparition d'hydrogène dans les carters des paliers, les jeux de barres blindés, l'apparition de l'eau et de l'huile dans l'alternateur, un signal d'alarme sera affiché dans un panneau de signalisation au niveau de la salle de commande et à protéger le turbo-alternateur à ces incidents. Les principales alarmes dans le système de refroidissement qui nécessite une intervention (tableau 4.7).

**Tableau 4.7** Les alarmes principales du système de refroidissement par Hydrogène

Alarme	Description
AI-SRH-01	BASSE PURETE D'HYDROGENE DANS LE CORPS TURBO-ALTERNATEUR
AI-SRH-02	BASSE PRESSION D'HYDROGENE DANS LE CORPS TURBOALTERNATEUR
AI-SRH-03	APPARITION DE L'HYDROGENE DANS LE CARTER DE PALIERS ET LES GAINES DE JEUX DE BARRES
AI-SRH-04	HAUTE TEMPERATURE D'HYDROGENE A LA SORTIE DE REFROIDISSEUR DU BLOC A FREON
AI-SRH-05	AUGMENTATION D'HUMIDITE D'HYDROGENE
AI-SRH-06	HYDROGENE DANS LE SYSTEME DE REFROIDISSEMENT PAR EAU
AI-SRH-07	AUGMENTATION DE CONCENTRATION D'OXYGENE DANS LES VERROUILLAGES HYDRAULIQUES
AI-SRH-08	AUGMENTATION DE PRESSION D'HYDROGENE DANS L'ALTERNATEUR
AI-SRH-09	HAUTE NIVEAU DE'HUILE DANS LE BERROUILLAGE HYDRAULIQUE
AI-SRH-10	AUGMENTATION DE LA QUANTITE D'EAU EVACUEE DU VAPORISATEUR DE LA MACHINE FRIGORIFIQUE

#### 4.4.2 Phase de la modélisation du système

La phase de modélisation représente les connaissances de domaine en langage OWL, Les composants principaux du modèle de connaissances sont les concepts, les propriétés et les instances, nous présentons les principales classes de MATO pour la tâche de diagnostic du système d'alarme sont : la classe *“ManufacturingProcess”*, *“ManufacturingTask”*, *“ManufacturingAgent”*, *“ManufacturingFeature”*, *“ManufacturingEquipment”*, *“ManufacturingProblem”*, *“ManufacturingSolution”*. Un exemple de connaissance du domaine des équipements principales et secondaires de la centrale est illustré à la figure 4.15.



```
<owl:Class>
  <owl:unionOf rdf:parseType="Collection">
    <owl:Class rdf:about="#Alternateur"/>
    <owl:Class rdf:about="#Pompe.Alimentaire"/>
    <owl:Class rdf:about="#Systeme.Graissage"/>
    <owl:Class rdf:about="#rechauffeur.Bas.pression.RBP"/>
    <owl:Class rdf:about="#Condenseur"/>
    <owl:Class rdf:about="#SecondEquipment"/>
    <owl:Class rdf:about="#Station.dessalement"/>
    <owl:Class rdf:about="#Turbine"/>
  </owl:unionOf>
  <owl:Class
rdf:about="#Systeme.Refroidissement.Hydrogene"/>
    <owl:Class rdf:about="#Degazeur"/>
    <owl:Class rdf:about="#Chaudiere"/>
    <owl:Class rdf:about="#Pompe.Circuit.Ferme"/>
    <owl:Class rdf:about="#Rechauffeur.Haute.pression.RHP"/>
    <owl:Class rdf:about="#Pompe.Extraction.1ere.etage"/>
    <owl:Class rdf:about="#Poste-Eau"/>
    <owl:Class rdf:about="#MainEquipment"/>
    <owl:Class rdf:about="#Pompe.Ewtraction.2eme.etage"/>
    <owl:Class rdf:about="#Station.Compression"/>
    <owl:Class rdf:about="#ManufacturingEquipment"/>
  </owl:Class>
```

**Figure 4.15** Exemple des équipements principaux et secondaires de la centrale (MainEquipment) en OWL

- La classe “**ManufacturingProcess**” décrit le processus industriel de la production d’électricité dans la centrale thermique.
- La classe “**ManufacturingTask**” décrit toutes tâches industrielles réalisées pour la production d’électricité.
- La classe “**ManufacturingAgent**” identifie et décrit les agents humains ou informatiques impliqués dans la réalisation des tâches industriels.
- La classe “**ManufacturingEquipment**” identifie et décrit tous les équipements industriels nécessaires utilisés pour la production d’électricité.
- Les équipements principaux sont: «Alternateur», «Chaudière», «Poste-Eau» et «Turbine», les “secondEquipment” composent également “Pompe.Circuit.Ferme”, “Station.Compression”, “Station.dessalement”, “System.Graissage” et “System.refroidissement.Hydrogen”.
- La classe “**ManufacturingProblem**” identifie et décrit tous les problèmes industriels durant la production d’électricité.
- La classe “**ManufacturingSolution**” identifie et propose toutes les solutions industrielles possibles pour résoudre les problèmes industriels durant la production d’électricité.

- La classe “*ManufacturingFeature*” identifie toutes les caractéristiques industrielles liées aux processus, agents, tâches, équipements, problèmes et solutions.

Les propriétés représentent les propriétés d’objets qui sont des relations entre les concepts et les propriétés des données qui sont les attributs, comme les attributs d’alarmes : « AlarmID », « AlarmDescription », « AlarmValue » et « AlarmState ».

Les attributs du système de refroidissement : « TemperatureHydrogène », « PressionHydrogène », un exemple des relations entre « manufacturingTask » et « manufacturingProblem » sont illustrées à la figure.4.16 et les attributs de système de refroidissement dans la figure 4.17.

```
<owl:Class rdf:about="#ManufacturingTask"/>
  <owl:Class rdf:about="#AlarmSystem"/>
  </owl:unionOf>
</owl:Class>
</rdfs:domain>
  <rdfs:range rdf:resource="#ManufacturingProblem"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasSolution">
  <rdfs:range rdf:resource="#ManufacturingSolution"/>
  <rdfs:domain rdf:resource="#ManufacturingProblem"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="identifies">
  <rdfs:range rdf:resource="#ManufacturingProblem"/>
  <rdfs:domain rdf:resource="#ManufacturingAgent"/>
</owl:ObjectProperty>
```

**Figure 4.16** Les relations entre “manufacturingtask”, “manufacturingProblem” et “manufacturingAgent” en OWL

```
AlarmDiagnosis#PressionAir">
  <rdfs:subPropertyOf
rdf:resource="http://www.semanticweb.org/user/ontologies/2020/2/AlarmDia
gnosis#Pression"/>
  </owl:DatatypeProperty>
  <!--
http://www.semanticweb.org/user/ontologies/2020/2/AlarmDiagnosis#Pressio
nCo2 -->
  <owl:DatatypeProperty
rdf:about="http://www.semanticweb.org/user/ontologies/2020/2/AlarmDiagno
sis#PressionCo2">
  <rdfs:subPropertyOf
rdf:resource="http://www.semanticweb.org/user/ontologies/2020/2/AlarmDia
gnosis#Pression"/>
  </owl:DatatypeProperty>
  <!--
http://www.semanticweb.org/user/ontologies/2020/2/AlarmDiagnosis#Pressio
nHydrogène -->
  <owl:DatatypeProperty
rdf:about="http://www.semanticweb.org/user/ontologies/2020/2/AlarmDiagno
sis#PressionHydrogène">
  <rdfs:subPropertyOf
rdf:resource="http://www.semanticweb.org/user/ontologies/2020/2/AlarmDia
gnosis#Pression"/>
  </owl:DatatypeProperty>
```

**Figure 4.17** Exemple des propriétés des données pour le système de refroidissement en OWL



Les individus de notre application sont l'élément de remplissage de la base des connaissances qui représentent les instances des classes, par exemple : l'individu « Al\_SRH\_1 » est un individu de type « Alarm », un exemple est illustré à la figure 4.18.

```
<!-- http://www.semanticweb.org/user/ontologies/2020/2/AlarmDiagnosis#Al_SRH_1
-->
  <owl:NamedIndividual
rdf:about="http://www.semanticweb.org/user/ontologies/2020/2/AlarmDiagnosis#Al_
SRH_1">
  <rdf:type
rdf:resource="http://www.semanticweb.org/user/ontologies/2020/2/AlarmDiagnosis#
Alarm"/>
  <causes
rdf:resource="http://www.semanticweb.org/user/ontologies/2020/2/AlarmDiagnosis#
Système_de_Refroidissement_par_Hydrogene"/>
  <AlarmDescription
rdf:datatype="http://www.w3.org/2001/XMLSchema#string"> BASE PURETE D'HYDROGENE
DANS LE CORPS TURBO-ALTERNATEUR </AlarmDescription>
  <AlarmID rdf:datatype="http://www.w3.org/2001/XMLSchema#NMToken">AL-SRH-
01</AlarmID>
  <AlarmState
rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean">true</AlarmState>
  <AlarmValue
rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean">false</AlarmValue>
  <rdfs:comment> BASE PURETE D'HYDROGENE DANS LE CORPS TURBO-ALTERNATEUR
</rdfs:comment>
  </owl:NamedIndividual>>
```

**Figure 4.18** Exemple de l'individu « Al\_SRH\_1 » de type Alarme en OWL

#### 4.4.3 Phase du raisonnement du système

La connaissance d'inférence dans un modèle de connaissances décrit les inférences, le plus bas niveau de la décomposition fonctionnelle de la tâche (voir section 3.2 Chap3). Ce sont des unités de traitement de l'information. La dernière étape dans la construction d'un modèle de connaissances est la description de chacune des inférences. Cette structure d'inférence est tirée de la méthodologie CommonKADS. Les étapes utilisées pour la réalisation de la tâche sont :

- Etape 1 : Identification de la tâche, le processus et la tâche lié au problème
- Etape.2: Vérification d'alarme
  - Etape.2.1: Vérification des paramètres liés à l'alarme
  - Etape.2.2: Diagnostiquer la fausse alarme
- Etape.3: Identification des causes de problèmes
  - Etape.3.1: Identifier les caractéristiques du problème
  - Etape.3.2: Identifier les caractéristiques du problème, des équipements et d'agents
  - Etape.3.3: Identification des causes possibles du problème
  - Etape.3.4: Identifier les agents intervenants au problème

- Etape.4: Proposition de la solution
  - Etape.4.1: Proposer la solution liée au problème
  - Etape.4.2: Identifier le type d'intervention
  - Etape.4.3: Identifier la procédure de la solution
  - Etape.4.4: Identifier les agents intervenants à la solution

Les Formalisations de la connaissance d'inférence utilisant le langage des règles SWRL, les règles spécifiques pour diagnostiquer les alarmes de système de refroidissement sont extraites à partir des règles génériques de MATO (voir tableau 3.4. chap3), Les règles spécifiques de la connaissance d'inférence par SWRL sont présentées au tableau 4.8. L'exemple suivant présente les étapes nécessaires pour la résolution des problèmes liés au système de refroidissement utilisant les connaissances de domaine en OWL et les connaissances d'inférence en utilisant les règles SWRL.

**AI.SRH.1: " BASSE PURETE D'HYDROGENE DANS LE CORPS TURBO-ALTERNATEUR "**

Ce message d'alarme est signalé lorsqu'il y a une perturbation dans le système de refroidissement de turbo-alternateur par l'hydrogène appliqué par la règle 1 pour configurer la tâche de diagnostic, et identifier le processus, la tâche et ses problèmes associés.

Etape 1 : Identifier la tâche, le processus et le problème lié à la tâche

Avant de diagnostiquer l'alarme, nous avons besoin d'identifier la tâche, le processus et le problème lié à la tâche, dans cette étape, nous avons utilisé la règle générique *Règle.Gén.1* puis la règle spécifique *SWRL.SRH.1*.

**Règle.Gén.1:**  $ManufacturingProcess(?P) \wedge ManufacturingTask(?T) \wedge ManufacturingProblem(?Pr) \wedge hasTask(?P, ?T) \rightarrow hasProblem(?T, ?Pr)$

Si un message d'alarme est signalé, SRH.1, il indique une perturbation dans les paramètres optimaux d'hydrogène alors configurer à la tâche de diagnostic liée au problème de refroidissement. Dans ce cas on applique la règle *SWRL.SRH.1*.

**SWRL.SRH.1:**  $Alarm(Al\_SRH\_1) \wedge Diagnosis(?D) \wedge ManufacturingProblem(?pr) \wedge AlarmState(Al\_SRH\_1, 1) \rightarrow hasProblem(?D, ?pr) \wedge Problem(Pr-Refroidissement-Alternateur-Hydrogène).$

Etape.2: Vérification d'alarme

Etape.2.1: Vérification des paramètres liés à l'alarme

Si l'alarme Al\_SRH\_1 est signalée et la Pureté d'hydrogène au long du volume d'alternateur est supérieure ou égale à 97%, alors l'alarme est fautive et on peut

diagnostiquer la fausse alarme. On applique la règle spécifique *SWRL.SRH.1-1* suivante pour vérifier l'alarme. Si l'alarme est vraie sauter à **l'étape 3**.

**SWRL.SRH.1-1:**  $Alarm(Al\_SRH\_1) \wedge Diagnosis(?D) \wedge AlarmState(Al\_SRH\_1, 1) \wedge MainEquipment(Alternateur) \wedge PureteHydrogène(Alternateur, ?prt) \wedge swrlb:greaterThanOrEqual(?prt, 97) \rightarrow AlarmValue(Al\_SRH\_1, false)$

**Etape.2.2:** diagnostiquer la fausse alarme

Après une vérification, le paramètre de pureté de l'hydrogène dans l'alternateur est contrôlé par l'analyseur de gaz dont l'appareil secondaire est installé dans le tableau de commande de blocs du tableau de signalisation de refroidissement. Dans ce cas, on applique la règle spécifique *SWRL.SRH.1-2*, vérifier les paramètres des capteurs de mesure et identifier les causes de la fausse alarme, Alors éliminer l'alarme et retour à l'étape 1.

**SWRL.SRH.1-2:**  $Alarm(Al\_SRH\_1) \wedge AlarmState(Al\_SRH\_1,1) \wedge AlarmValue(Al\_SRH\_1, false) \wedge Problem(Pr-Refroidissement-Alternateur-Hydrogène) \wedge hasAlarm(Pr-Refroidissement-Alternateur-Hydrogène, Al\_SRH\_1) \rightarrow hasCause(Pr-Refroidissement-Alternateur-Hydrogène, FaultyAutomatic) \wedge hasCause(Pr-Refroidissement-Alternateur-Hydrogène, Circuit-Automatic) \wedge hasCause(Pr-Refroidissement-Alternateur-Hydrogène, FaultyEquipment) \wedge hasCause(Pr-Refroidissement-Alternateur-Hydrogène, Analyseur-Gaz).$

**Etape.3:** Identification des causes de problème

Si l'alarme *Al\_SRH\_1* signalée est vraie et la pureté d'hydrogène au long du volume d'alternateur est inférieure à 97%, alors on identifie les causes possibles. Avant l'identification des causes, on applique les règles génériques : 2, 3 et 4 pour identifier les caractéristiques du problème, les caractéristiques de l'équipement lié au problème et les caractéristiques des agents (les experts du domaine) qui peuvent intervenir à ce problème. Nous avons utilisé la règle générique *Règle.Gén.2* puis la règle spécifique *SWRL.SRH.1-3*.

**Etape.3.1:** Identifier les caractéristiques du problème

L'identification des caractéristiques du problème est une étape importante dans notre tâche, dans ce cas on applique la règle *Règle.Gén.2* puis *SWRL.SRH.1-3*.

**Règle.Gén.2:**  $ManufacturingProblem(?Pr) \wedge ManufacturingFeature(?F) \rightarrow hasFeature(?Pr, ?F)$

**SWRL.SRH.1-3:**  $Problem(Pr-Refroidissement-Alternateur-Hydrogène) \wedge ProblemFeature(?F) \rightarrow hasFeature(Pr-Refroidissement-Alternateur-Hydrogène, ?F)$

**Etape.3.2:** Identifier les caractéristiques du problème, des équipements et d'agents  
Après l'identification des caractéristiques du problème, on définit les caractéristiques des équipements liés au problème et les caractéristiques des agents qui peuvent intervenir au diagnostic, dans ce cas on exécute la règle générique 3 et la règle spécifique SWRL.SRH.1-4.

**Règle.Gén.3:**  $ManufacturingProblem(?Pr) \wedge ManufacturingFeature(?F) \wedge ManufacturingEquipment(?E) \wedge hasProblem(?E, ?Pr) \wedge ManufacturingAgent(?A) \rightarrow hasFeature(?Pr, ?F) \wedge hasFeature(?E, ?F) \wedge hasFeature(?A, ?F)$

**SWRL.SRH.1-4 :**  $Problem(Pr-Refroidissement-Alternateur-Hydrogène) \wedge ProblemFeature(?PF) \wedge MainEquipment(Alternateur) \wedge EquipmentFeature(?EF) \wedge hasProblem(?E, Pr-Refroidissement-Alternateur-Hydrogène) \wedge ManufacturingAgent(?A) \wedge AgentFeature(?AF) \wedge identifies(?Ag, Pr-Refroidissement-Alternateur-Hydrogène) \rightarrow hasFeature(Pr-Refroidissement-Alternateur-Hydrogène, ?PF) \wedge hasFeature(autogen0:Alternateur, ?EF) \wedge hasFeature(?A, ?AF)$

**Etape.3.3:** Identification les causes possibles de problème

Après les caractéristiques d'équipement de problème et d'agent, on exécute la règle générique Règle.Gén.4 puis la règle spécifique SWRL.SRH.1-5 pour identifier toutes les causes possibles de problème sachant que le problème est lié aux équipements. On peut aussi exécuter la règle inverse SWRL.SRH.1-6.

**Règle.Gén.4:**  $ManufacturingProblem(?Pr) \wedge ManufacturingEquipment(?E) \wedge hasProblem(?E, ?Pr) \rightarrow causes(?E, ?Pr)$

**SWRL.SRH.1-5 :**  $Problem(Pr-Refroidissement-Alternateur-Hydrogène) \wedge ManufacturingEquipment(?E) \wedge Cause(FaultyEquipment) \wedge MainEquipment(Alternateur) \wedge hasProblem(Alternateur, Pr-Refroidissement-Alternateur-Hydrogène) \rightarrow causes(?E, Pr-Refroidissement-Alternateur-Hydrogène) \wedge hasCause(Pr-Refroidissement-Alternateur-Hydrogène, FaultyEquipment)$

**SWRL.SRH.1-6 :**  $Problem(Pr-Refroidissement-Alternateur-Hydrogène) \wedge ManufacturingEquipment(?E) \wedge causes(?E, Pr-Refroidissement-Alternateur-Hydrogène) \wedge hasCause(Pr-Refroidissement-Alternateur-Hydrogène, FaultyEquipment) \rightarrow isCausedBy(Pr-Refroidissement-Alternateur-Hydrogène, ?E)$

**Etape.3.4:** Identification des agents intervenant au problème

L'identification des agents intervenant au problème par l'exécution de la règle générique Règle.Gén.5 puis la règle spécifique SWRL.SRH.1-7 pour identifier tous les agents qui peuvent intervenir à l'identification des causes de problème. On peut aussi exécuter la règle inverse SWRL.SRH.1-8.

**Règle.Gén.5:**  $ManufacturingProblem(?Pr) \wedge ManufacturingAgent(?A) \rightarrow identifies(?A, ?Pr)$

**SWRL.SRH.1-7 :**  $Problem(Pr-Refroidissement-Alternateur-Hydrogène) \wedge ManufacturingAgent(?A) \wedge ManufacturingEquipment(?E) \wedge isCausedBy(Pr-Refroidissement-Alternateur-Hydrogène, ?E) \rightarrow identifies(?A, Pr-Refroidissement-Alternateur-Hydrogène)$

**SWRL.SRH.1-8 :**                    *Problem(Pr-Refroidissement-Alternateur-Hydrogène)                    ^  
ManufacturingAgent(?A) -> isInterventionBy(Pr-Refroidissement-Alternateur-Hydrogène,  
?A)*

**Etape.4:** Proposition de la solution

Après l'identification du problème et les causes de problème, cette étape propose la solution liée au problème, le type d'intervention et la procédure de solution et aussi d'identifier les agents intervenant à la solution de problème, on exécute les règles suivantes :

**Etape.4.1:** Proposer la solution liée au problème

**Règle.Gén.6:**    *Problem(?Pr)    ^    ManufacturingEquipment(?E)    ^    Solution    (?Sol)^  
hasProblem(?E, ?Pr) ^causes(?E, ?Pr) ->hasSolution(?Pr,?Sol)*

**SWRL.SRH.1-9 :**                    *Problem(Pr-Refroidissement-Alternateur-Hydrogène)                    ^  
ManufacturingEquipment(?E)    ^    Solution(?Sol)    ^    hasProblem(?E, Pr-Refroidissement-  
Alternateur-Hydrogène)    ^    Cause(FaultyEquipment)    ^    causes(FaultyEquipment, Pr-  
Refroidissement-Alternateur-Hydrogène)    ^    causes(?E, Pr-Refroidissement-Alternateur-  
Hydrogène)    ^    hasCause(Pr-Refroidissement-Alternateur-Hydrogène, FaultyEquipment) ->  
hasSolution(Pr-Refroidissement-Alternateur-Hydrogène, ?Sol)*

**Etape.4.2:** Identification du type d'intervention

**Règle.Gén.7:**                    *Problem(?Pr)^ManufacturingEquipment( ?E)^causes(?E,                    ?Pr)  
^Solution( ?Sol)^                    hasSolution(?Pr,?Sol)^                    InterventionType(?IT)-  
>hasInterventionType( ?Sol, ?IT)*

**SWRL.SRH.1-10 :** *Problem(Pr-Refroidissement-Alternateur-Hydrogène)^causes(?E, Pr-  
Refroidissement-Alternateur-Hydrogène) ^Solution( ?Sol)^ hasSolution(Pr-Refroidissement-  
Alternateur-Hydrogène,?Sol)^ InterventionType(?IT)->hasInterventionType( ?Sol, ?IT)*

**Etape.4.3:** Identification de la procédure de la solution

**Règle.Gén.8:**                    *Problem(?Pr)                    ^                    causes(?E,                    ?Pr)                    ^Solution( ?Sol)^  
hasSolution(?Pr,?Sol)^ProceduralSolution(?PS)->hasProceduralSolution( ?Sol, ?PS)*

**SWRL.SRH.1-11 :** *Problem(Pr-Refroidissement-Alternateur-Hydrogène)    ^    causes(?E, Pr-  
Refroidissement-Alternateur-Hydrogène)    ^Solution( ?Sol)^    hasSolution(Pr-Refroidissement-  
Alternateur-Hydrogène,?Sol)^ProceduralSolution(?PS)->hasProceduralSolution( ?Sol, ?PS)*

**Etape.4.4:** Identification des agents intervenants à la solution

**Règle.Gén.9:**    *Problem(?Pr)    ^    ManufacturingAgent(?A)    ^    Solution(?Sol) -> proposes(?A,  
?Sol)*

**SWRL.SRH.1-12 :**                    *Problem(Pr-Refroidissement-Alternateur-Hydrogène)                    ^  
ManufacturingAgent(?A)    ^    Solution(?Sol) -> proposes(?A, ?Sol)*

**Règle.Gén.10:**    *Problem(?Pr)                    ^                    ManufacturingAgent(?A)                    ^                    Solution(?Sol) ->  
isProposedBy(?Sol, ?A)*

**SWRL.SRH.1-13 :**                    *Problem(Pr-Refroidissement-Alternateur-Hydrogène)                    ^  
ManufacturingAgent(?A)    ^    Solution(?Sol) -> isProposedBy(?Sol, ?A)*

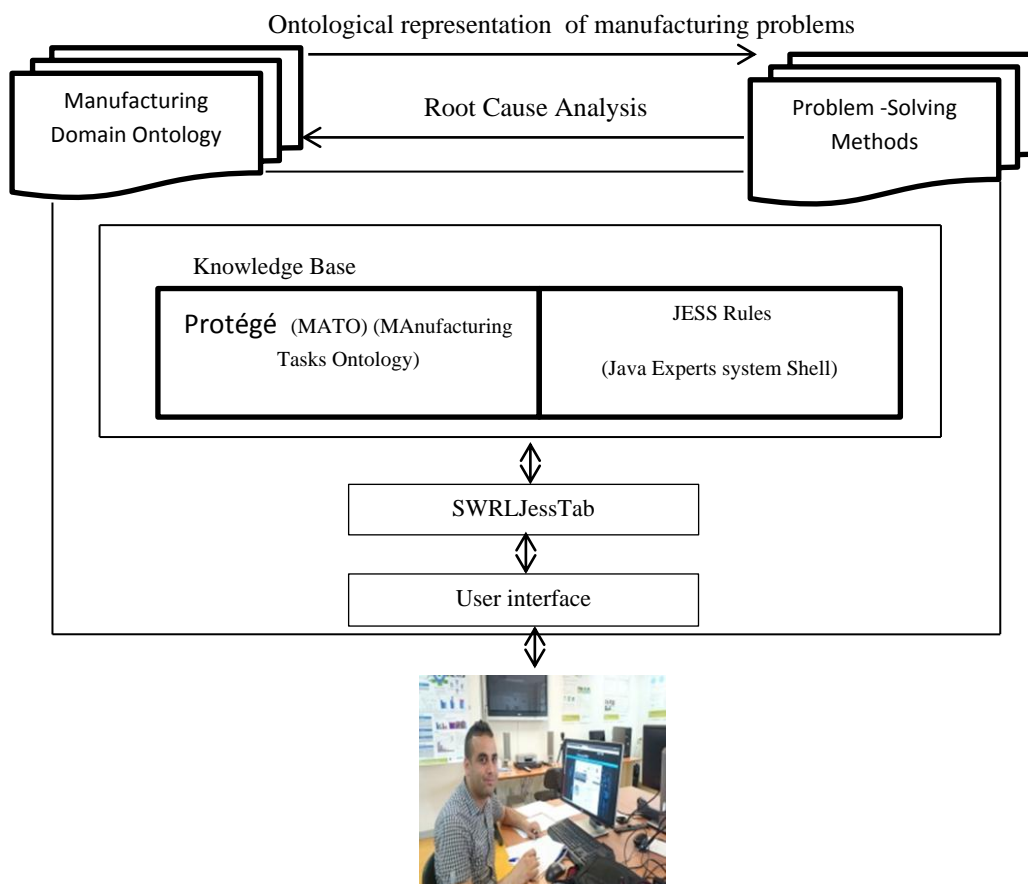
**Tableau 4.8:** les règles SWRL spécifique pour diagnostiquer l’alarme Al\_SRH\_1

SWRL.SRH.1	<u>Identification du problème de refroidissement, l’alarme et la tâche de diagnostique</u> <i>Alarm(Al_SRH_1) ^ Diagnosis(?D) ^ ManufacturingProblem(?pr) ^ AlarmState(Al_SRH_1, 1) -&gt; hasProblem(?D, ?pr) ^ Problem(Pr-Refroidissement-Alternateur-Hydrogène).</i>
SWRL.SRH.1-1	<u>Vérification de l’alarme</u> <i>Alarm(Al_SRH_1) ^ Diagnosis(?D) ^ AlarmState(Al_SRH_1, 1) ^ MainEquipment(Alternateur) ^ PuretéHydrogène(Alternateur, ?prt) ^ swrlb:greaterThanOrEqual(?prt, 97) -&gt; AlarmValue(Al_SRH_1, false)</i>
SWRL.SRH.1-2	<u>Diagnostiquer la fausse alarme</u> <i>Alarm(Al_SRH_1) ^ AlarmState(Al_SRH_1, 1) ^ AlarmValue(Al_SRH_1, false) ^ Problem(Pr-Refroidissement-Alternateur-Hydrogène) ^ hasAlarm(Pr-Refroidissement- Alternateur-Hydrogène, Al_SRH_1) -&gt; hasCause(Pr-Refroidissement-Alternateur-Hydrogène, FaultyAutomatic) ^ hasCause(Pr-Refroidissement-Alternateur-Hydrogène, Circuit-Automatic) ^ hasCause(Pr-Refroidissement-Alternateur-Hydrogène, FaultyEquipment) ^ hasCause(Pr-Refroidissement-Alternateur-Hydrogène, Analyseur-Gaz).</i>
SWRL.SRH.1-3	<u>Identification des caractéristiques du problème</u> <i>Problem(Pr-Refroidissement-Alternateur-Hydrogène) ^ ProblemFeature(?F) -&gt; hasFeature(Pr-Refroidissement-Alternateur-Hydrogène, ?F)</i>
SWRL.SRH.1-4	<u>Identification des caractéristiques du problème, les équipements et les agents</u> <i>Problem(Pr-Refroidissement-Alternateur-Hydrogène) ^ ProblemFeature(?PF) ^ MainEquipment(Alternateur) ^ EquipmentFeature(?EF) ^ hasProblem(?E, Pr-Refroidissement-Alternateur-Hydrogène) ^ ManufacturingAgent(?A) ^ AgentFeature(?AF) ^ identifies(?Ag, Pr-Refroidissement-Alternateur-Hydrogène) -&gt; hasFeature(Pr-Refroidissement-Alternateur-Hydrogène, ?PF) ^ hasFeature(Alternateur, ?EF) ^ hasFeature(?A, ?AF)</i>
SWRL.SRH.1-5	<u>identifier les causes possibles du problème</u> <i>Problem(Pr-Refroidissement-Alternateur-Hydrogène) ^ ManufacturingEquipment(?E) ^ Cause(FaultyEquipment) ^ MainEquipment(Alternateur) ^ hasProblem(Alternateur, Pr-Refroidissement-Alternateur-Hydrogène) -&gt; causes(?E, Pr-Refroidissement-Alternateur-Hydrogène) ^ hasCause(Pr-Refroidissement-Alternateur-Hydrogène, FaultyEquipment)</i>
SWRL.SRH.1-6	<u>La règle inverse de SWRL.SRH.1-5</u> <i>Problem(Pr-Refroidissement-Alternateur-Hydrogène) ^ ManufacturingEquipment(?E) ^ causes(?E, Pr-Refroidissement-Alternateur-Hydrogène) ^ hasCause(Pr-Refroidissement-Alternateur-Hydrogène, FaultyEquipment) -&gt; isCausedBy(Pr-Refroidissement-Alternateur-Hydrogène, ?E)</i>
SWRL.SRH.1-7	<u>Identification des agents intervenant au problème</u> <i>Problem(Pr-Refroidissement-Alternateur-Hydrogène) ^ ManufacturingAgent(?A) ^ ManufacturingEquipment(?E) ^ isCausedBy(Pr-Refroidissement-Alternateur-Hydrogène, ?E) -&gt; identifies(?A, Pr-Refroidissement-Alternateur-Hydrogène)</i>
SWRL.SRH.1-8	<u>La règle inverse de SWRL.SRH.1-7</u> <i>Problem(Pr-Refroidissement-Alternateur-Hydrogène) ^ ManufacturingAgent(?A) -&gt; isInterventionBy(Pr-Refroidissement-Alternateur-Hydrogène, ?A)</i>
SWRL.SRH.1-9	<u>Identification de la solution de problème</u> <i>Problem(Pr-Refroidissement-Alternateur-Hydrogène) ^ ManufacturingEquipment(?E) ^ Solution(?Sol) ^ hasProblem(?E, Pr-Refroidissement-Alternateur-Hydrogène) ^ Cause(FaultyEquipment) ^ causes(FaultyEquipment, Pr-Refroidissement-Alternateur-Hydrogène) ^ causes(?E, Pr-Refroidissement-Alternateur-Hydrogène) ^ hasCause(Pr-Refroidissement-Alternateur-Hydrogène, FaultyEquipment) -&gt; hasSolution(Pr-Refroidissement-Alternateur-Hydrogène, ?Sol)</i>
SWRL.SRH.1-10	<u>Identification du type d’intervention</u> <i>Problem(Pr-Refroidissement-Alternateur-Hydrogène) ^ causes(?E, Pr-Refroidissement-Alternateur-Hydrogène) ^ Solution(?Sol) ^ hasSolution(Pr-Refroidissement-Alternateur-Hydrogène, ?Sol) ^ InterventionType(?IT) -&gt; hasInterventionType(?Sol, ?IT)</i>
SWRL.SRH.1-11	<u>Identification de la procédure de la solution</u> <i>Problem(Pr-Refroidissement-Alternateur-Hydrogène) ^ causes(?E, Pr-Refroidissement-Alternateur-Hydrogène) ^ Solution(?Sol) ^ hasSolution(Pr-Refroidissement-Alternateur-Hydrogène, ?Sol) ^ ProceduralSolution(?PS) -&gt; hasProceduralSolution(?Sol, ?PS)</i>
SWRL.SRH.1-12	<u>Identification des agents intervenant à la solution</u> <i>Problem(Pr-Refroidissement-Alternateur-Hydrogène) ^ ManufacturingAgent(?A) ^ Solution(?Sol) -&gt; proposes(?A, ?Sol)</i>
SWRL.SRH.1-13	<u>La règle inverse de SWRL.SRH.1-12</u> <i>Problem(Pr-Refroidissement-Alternateur-Hydrogène) ^ ManufacturingAgent(?A) ^ Solution(?Sol) -&gt; isProposedBy(?Sol, ?A)</i>



#### 4.4.4 Phase d'implémentation

Dans l'étape d'implémentation, nous avons utilisé Protégé, un outil d'édition le plus ontologique utilisé pour construire l'ontologie. Dans ce travail, nous avons utilisé Protégé 5.5 pour développer la connaissance du domaine, le langage SWRL (Semantic Web Rule Language) pour éditer les règles. Pour combiner les ontologies et les règles: Ontologies par OWL-DL et règles par RuleML (un langage de règles). L'éditeur Protégé OWL prend en charge les règles SWRL, comme les raisonneurs Pellet et Hermit. À l'aide de l'API OWL, SWRL peut être utilisé avec un code Java. Et nous avons utilisé JESS (Java Expert System Shell) pour développer des connaissances sur l'inférence. Nous avons utilisé JessTab développé par (Henrik Eriksson, 2003), est une approche pour intégrer Jess et protégé utilisé dans le plug-in protégé. Il crée une cartographie Protégé et JESS, la combinaison entre la connaissance du domaine (MATO) et l'inférence de domaine (JESS) pour améliorer la résolution de problèmes, la structure d'implémentation des connaissances de domaine et de connaissances d'inférence est illustrée à la figure 4.19.



**Figure 4.19** structure of MATO and PSM (Problem Solving methods)

Dans la construction de la modélisation du diagnostic du système d'alarme, toutes les classes pour le diagnostic de système d'alarme sont présentées par Protégé, montrée dans la figure 4.20 qui présente la connaissance de domaine et nous avons implémenté des règles en utilisant un plug-in SWRLJessTab, le moteur de règles Jess utilisé pour transformer l'ontologie de diagnostic du système d'alarme en OWL en faits JESS et toutes les règles de SWRL en règles JESS. Montrez la figure 4.21 qui présente la connaissance d'inférence.

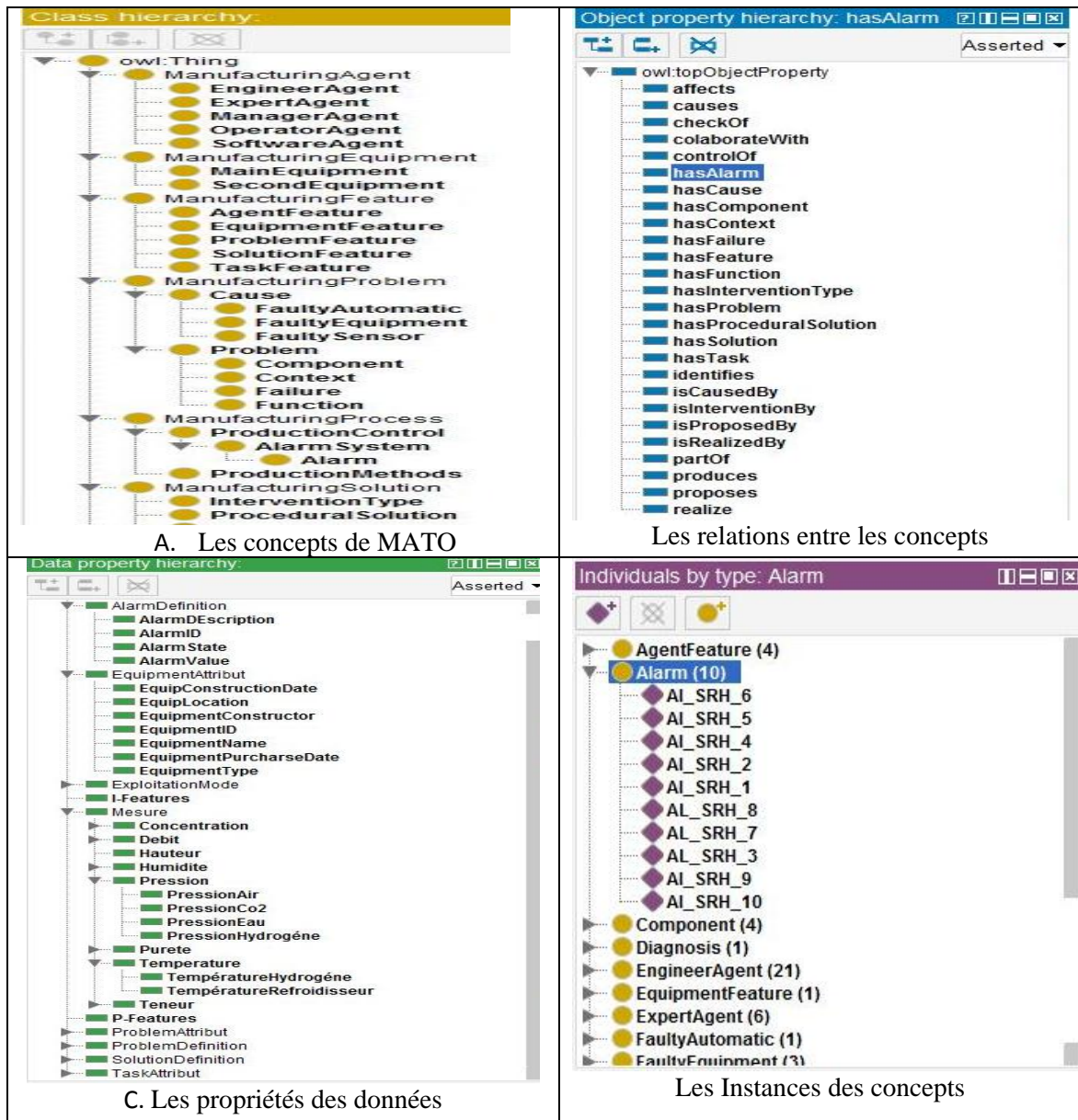


Figure 4.20 Implémentation de la connaissance de domaine dans Protégé.



## Chapitre 4 : Application du processus de capitalisation et de partage de connaissances au niveau de la centrale thermique

Active ontology \* Classes \* Object properties \* Data properties \* Individuals by class \* SWRLTab \* SQWRLTab \*

Name	Rule	Comment
<input type="checkbox"/> SQWRL_SRH.1false.6	AlarmDiagnosis:partOf(?PS, ?Sol) ^ AlarmDiagnosis:CorrectiveAction(?Sol, ?CA) ^ AlarmDiagnosis:Solution...	selection solution and procedural so...
<input checked="" type="checkbox"/> SWRL_R1.Generic	AlarmDiagnosis:ManufacturingProblem(?Pr) ^ AlarmDiagnosis:ManufacturingTask(?T) ^ AlarmDiagnosis:has...	Rule.generic.ManufacturingProcess/...
<input checked="" type="checkbox"/> SWRL_R10.Generic	AlarmDiagnosis:hasProblem(?E, ?Pr) ^ AlarmDiagnosis:ManufacturingSolution(?Sol) ^ AlarmDiagnosis:Man...	Configuration and Affecation rule
<input checked="" type="checkbox"/> SWRL_R2.Generic	AlarmDiagnosis:ManufacturingProblem(?Pr) ^ AlarmDiagnosis:ManufacturingFeature(?F) -> AlarmDiagnosis:...	Define_problem_Feature
<input checked="" type="checkbox"/> SWRL_R3.Generic	AlarmDiagnosis:hasProblem(?E, ?Pr) ^ AlarmDiagnosis:ManufacturingEquipment(?E) ^ AlarmDiagnosis:Ma...	Define_ProblemFeature/Equipmen...
<input checked="" type="checkbox"/> SWRL_R4.Generic	AlarmDiagnosis:hasProblem(?E, ?Pr) ^ AlarmDiagnosis:ManufacturingEquipment(?E) ^ AlarmDiagnosis:Ma...	Define_Causes_Of_Problem
<input checked="" type="checkbox"/> SWRL_R5.Generic	AlarmDiagnosis:ManufacturingAgent(?A) ^ AlarmDiagnosis:ManufacturingProblem(?Pr) -> AlarmDiagnosis:...	DefineAgent for identifies problem

Edit

Name: SWRL\_R1.Generic

Comment: Rule.generic.ManufacturingProcess/ManufacturingTask/ManufacturingProblem.

Status: AlarmDiagnosis:ManufacturingProblem(?Pr) ^ AlarmDiagnosis:ManufacturingTask(?T) ^ AlarmDiagnosis:hasTask(?P, ?T) ^ AlarmDiagnosis:ManufacturingProcess(?P) -> AlarmDiagnosis:hasProblem(?T, ?Pr)

Cancel Ok

New Edit Clone Delete

### A. Implémentation des règles SWRL dans Protégé

Active ontology \* Classes \* Object properties \* Data properties \* Individuals by class \* SWRLTab \* SQWRLTab \*

Name	Rule	Comment
<input type="checkbox"/> SQWRL_SRH.1false.6	AlarmDiagnosis:partOf(?PS, ?Sol) ^ AlarmDiagnosis:CorrectiveAction(?Sol, ?CA) ^ AlarmDiagnosis:Solution...	selection solution and procedural so...
<input checked="" type="checkbox"/> SWRL_R1.Generic	AlarmDiagnosis:ManufacturingProblem(?Pr) ^ AlarmDiagnosis:ManufacturingTask(?T) ^ AlarmDiagnosis:has...	Rule.generic.ManufacturingProcess/...
<input checked="" type="checkbox"/> SWRL_R10.Generic	AlarmDiagnosis:hasProblem(?E, ?Pr) ^ AlarmDiagnosis:ManufacturingSolution(?Sol) ^ AlarmDiagnosis:Man...	Configuration and Affecation rule
<input checked="" type="checkbox"/> SWRL_R2.Generic	AlarmDiagnosis:ManufacturingProblem(?Pr) ^ AlarmDiagnosis:ManufacturingFeature(?F) -> AlarmDiagnosis:...	Define_problem_Feature
<input checked="" type="checkbox"/> SWRL_R3.Generic	AlarmDiagnosis:hasProblem(?E, ?Pr) ^ AlarmDiagnosis:ManufacturingEquipment(?E) ^ AlarmDiagnosis:Ma...	Define_ProblemFeature/Equipmen...
<input checked="" type="checkbox"/> SWRL_R4.Generic	AlarmDiagnosis:hasProblem(?E, ?Pr) ^ AlarmDiagnosis:ManufacturingEquipment(?E) ^ AlarmDiagnosis:Ma...	Define_Causes_Of_Problem
<input checked="" type="checkbox"/> SWRL_R5.Generic	AlarmDiagnosis:ManufacturingAgent(?A) ^ AlarmDiagnosis:ManufacturingProblem(?Pr) -> AlarmDiagnosis:...	DefineAgent for identifies problem
<input checked="" type="checkbox"/> SWRL_R6.Generic	AlarmDiagnosis:ManufacturingSolution(?Sol) ^ AlarmDiagnosis:ManufacturingAgent(?A) ^ AlarmDiagnosis:...	Proposition of Solution by Agent
<input checked="" type="checkbox"/> SWRL_R7.Generic	AlarmDiagnosis:ManufacturingSolution(?Sol) ^ AlarmDiagnosis:ManufacturingAgent(?A) ^ AlarmDiagnosis:...	Identification of Problem and propos...
<input type="checkbox"/> Select4	AlarmDiagnosis:Component(?Cp) ^ AlarmDiagnosis:hasComponent(?fa, ?Cp) ^ AlarmDiagnosis:Compo...	
<input type="checkbox"/> Select1	AlarmDiagnosis:Problem(AlarmDiagnosis:Pr-Refroidissement-Alternateur-Hydrogene) ^ AlarmDiagnosis:Pro...	
<input type="checkbox"/> Sqwrl.1-10	AlarmDiagnosis:Problem(AlarmDiagnosis:Pr-Refroidissement-Alternateur-Hydrogene) ^ AlarmDiagnosis:ha...	selection agent identification of caus...
<input type="checkbox"/> Sqwrl.1-5	AlarmDiagnosis:ManufacturingEquipment(AlarmDiagnosis:Alternateur) ^ AlarmDiagnosis:EquipmentFeatur...	selecting feature equipment
<input type="checkbox"/> Sqwrl.1-6	AlarmDiagnosis:ManufacturingEquipment(AlarmDiagnosis:Alternateur) ^ AlarmDiagnosis:EquipmentID(Alar...	selecting information alternateur

New Edit Clone Delete

Control Rules Asserted Axioms Inferred Axioms OWL 2 RL

OWL axioms successfully transferred to rule engine.  
 Number of SWRL rules exported to rule engine: 23  
 Number of OWL class declarations exported to rule engine: 46  
 Number of OWL individual declarations exported to rule engine: 128  
 Number of OWL object property declarations exported to rule engine: 26  
 Number of OWL data property declarations exported to rule engine: 79  
 Total number of OWL axioms exported to rule engine: 1152  
 The transfer took 1497 millisecond(s).  
 Press the 'Run Drools' button to run the rule engine.

### B. Transférer les connaissances de domaine OWL et règles SWRL en JESS

Active ontology \* Classes \* Object properties \* Data properties \* Individuals by class \* SWRLTab \* SQWRLTab \*

Name	Rule	Comment
<input type="checkbox"/> SQWRL_SRH.1false.6	AlarmDiagnosis:partOf(?PS, ?Sol) ^ AlarmDiagnosis:CorrectiveAction(?Sol, ?CA) ^ AlarmDiagnosis:Solution...	selection solution and procedural so...
<input checked="" type="checkbox"/> SWRL_R1.Generic	AlarmDiagnosis:ManufacturingProblem(?Pr) ^ AlarmDiagnosis:ManufacturingTask(?T) ^ AlarmDiagnosis:has...	Rule.generic.ManufacturingProcess/...

New Edit Clone Delete

Control Rules Asserted Axioms Inferred Axioms OWL 2 RL

A

```

DataProperty: ContextType
CorrectiveAction some xsd:string EquivalentTo RootCause some xsd:string
Class: MainEquipment
TPC5 identifies Pr-Refroidissement-Alternateur-Hydrogene
Problem(Pr-Refroidissement-Alternateur-Hydrogene), AgentFeature(?AF), ExpertAgent(?EA), AgentName(?EA, ?N), identifies(?EA, Pr-Refroidissement-Alternateur-Hydrogene), hasFe...
TaskTime Range: xsd:dateTime
InpProduction6 AgentAge "56"^^xsd:int
InpProduction10 AgentFunction "Assurer la production"^^xsd:string
InstallingDate Range: xsd:dateTime
Temperature SubPropertyOf: Measure
causes Domain ManufacturingEquipment
PrecisionCo2 SubPropertyOf: Precision
hasSolution Range Solution
CMMS Type SoftwareAgent
Expert6 AgentName "Krid Nouredine"^^xsd:string
SubTaskID SubPropertyOf: TaskAttribut
TemperatureHydrogene SubPropertyOf: Temperature
ExpFeature4 Type AgentFeature
InpProduction5 AgentID "122531"^^xsd:int
Expert2 isProposedBy Expert2
DataProperty: ComponentType
ProductionControl SubClassOf ManufacturingProcess
Individual: Sol SRH.1
DataProperty: ProblemDescription
TPC1 AgentExperience "18"^^xsd:int
TPCC1 AgentExperience "7"^^xsd:int
    
```

### C. les axiomes importés

Active ontology \* Classes \* Object properties \* Data properties \* Individuals by class \* SWRLTab \* SQWRLTab \*

Name	Rule	Comment
<input type="checkbox"/> SQWRL_SRH.1false.6	AlarmDiagnosis:partOf(?PS, ?Sol) ^ AlarmDiagnosis:CorrectiveAction(?Sol, ?CA) ^ AlarmDiagnosis:Solution...	selection solution and procedural so...
<input checked="" type="checkbox"/> SWRL_R1.Generic	AlarmDiagnosis:ManufacturingProblem(?Pr) ^ AlarmDiagnosis:ManufacturingTask(?T) ^ AlarmDiagnosis:has...	Rule.generic.ManufacturingProcess/...

New Edit Clone Delete

Control Rules Asserted Axioms Inferred Axioms OWL 2 RL

OWL axioms successfully transferred to rule engine.  
 Number of SWRL rules exported to rule engine: 23  
 Number of OWL class declarations exported to rule engine: 46  
 Number of OWL individual declarations exported to rule engine: 128  
 Number of OWL object property declarations exported to rule engine: 26  
 Number of OWL data property declarations exported to rule engine: 79  
 Total number of OWL axioms exported to rule engine: 1152  
 The transfer took 1497 millisecond(s).  
 Press the 'Run Drools' button to run the rule engine.  
 Successful execution of rule engine.  
 Number of inferred axioms: 6120  
 The process took 3827 millisecond(s).  
 Look at the 'Inferred Axioms' tab to see the inferred axioms.  
 Press the 'Drools->OWL' button to translate the inferred axioms to OWL knowledge.  
 Successfully transferred inferred axioms to OWL model.  
 The process took 1189 millisecond(s).  
 Successfully transferred inferred axioms to OWL model.  
 The process took 467 millisecond(s).

### D. La transformation succès des connaissances en OWL et règles SWRL pour l'inférence

**Figure 4.21** Implémentation de la connaissance d'inférence dans Protégé.

139

#### 4.4.5 Sélection des connaissances de l'ontologie MATO par les requêtes SQWRL pour Diagnostic du système d'alarmes

Les opérateurs principaux utilisent un antécédent des règles SWRL génériques de MATO comme spécification de modèle et remplace la règle résultante par les opérateurs de sélection SQWRL. Les requêtes spécifiques de l'application sont extraites à partir des requêtes génériques SQWRL de l'ontologie MATO (voir tableau 3.9 Chapitre.3.). La sélection des connaissances facilite le partage des connaissances durant l'exécution de la tâche. Les requêtes spécifiques pour la sélection des connaissances par SQWRL sont présentées au tableau 4.9. Les requêtes SQWRL peuvent fonctionner en conjonction avec SWRL et peuvent ainsi être utilisées pour récupérer les connaissances déduites par les règles SWRL. L'exemple suivant présente les étapes nécessaires pour le partage des connaissances durant la résolution des problèmes liés au système de refroidissement utilisant les requêtes spécifiques SQWRL. Les figures 4.22 et 4.23 représentent respectivement la sélection des causes possibles et la procédure de la solution pour la résolution de problèmes liés à l'alarme Al\_SRH\_1.

##### **Etape.3.3:** Sélection des causes possibles de problème lié à l'alarme Al\_SRH\_1

**SOWRL.SRH.1.9:** *Problem(Pr-Refroidissement-Alternateur-Hydrogène) ^ Alarm(Al\_SRH\_1) ^ hasAlarm(Pr-Refroidissement-Alternateur-Hydrogène, Al\_SRH\_1) ^ FaultyEquipment(?FE) ^ hasCause(Pr-Refroidissement-Alternateur-Hydrogène, ?FE) ^ Component(?Cp) ^ hasComponent(?FE, ?Cp) -> sqwrl:select(Pr-Refroidissement-Alternateur-Hydrogène, Al\_SRH\_1, ?FE, ?Cp) ^ sqwrl:columnNames("Problem", "Alarm", "Faulty Equipment", "caused by ")*.

##### **Etape.4.3:** Sélection de procédure de la solution

**SOWRL.SRH.1.13:** *Problem(Pr-Refroidissement-Alternateur-Hydrogène) ^ Alarm(Al\_SRH\_1) ^ hasAlarm(Pr-Refroidissement-Alternateur-Hydrogène, Al\_SRH\_1) ^ FaultyEquipment(?FE) ^ hasCause(Pr-Refroidissement-Alternateur-Hydrogène, ?FE) ^ Component(?Cp) ^ hasComponent(?FE, ?Cp) -> sqwrl:select(Pr-Refroidissement-Alternateur-Hydrogène, Al\_SRH\_1, ?FE, ?Cp) ^ sqwrl:columnNames("Problem", "Alarm", "Faulty Equipment", "caused by ")*.

**Tableau 4.9:** Les règles SOWRL spécifiques pour diagnostiquer l’alarme Al SRH 1

SQWRL.SRH.1	<p><u>Identification de la tâche</u></p> <p><i>Alarm(Al_SRH_1) ^ Diagnosis(?D) ^ AlarmState(Al_SRH_1, 1) ^ TaskID(?D, ?ID) ^ TaskName(?D, ?N) ^ TaskTime(?D, ?T) ^ TaskType(?D, ?Tp) -&gt; sqwrl:select(?D, ?ID, ?N, ?T, ?Tp) ^ sqwrl:orderBy(?ID)</i></p>
SQWRL.SRH.1-1	<p><u>Identification de l’ Alarme signalé</u></p> <p><i>Alarm(Al_SRH_1) ^ AlarmID(Al_SRH_1, ?ID) ^ AlarmDEscription(Al_SRH_1, ?AD) ^ AlarmValue(Al_SRH_1, ?V) ^ AlarmState(Al_SRH_1, ?ST) -&gt; sqwrl:select(Al_SRH_1, ?ID, ?AD, ?V, ?ST) ^ sqwrl:orderBy(?ID)</i></p>
SQWRL.SRH.1-2	<p><u>Localisation du problème</u></p> <p><i>Problem(Pr-Refroidissement-Alternateur-Hydrogène) ^ ProblemID(Pr-Refroidissement-Alternateur-Hydrogène, ?ID) ^ ProblemType(Pr-Refroidissement-Alternateur-Hydrogène, ?Tp) ^ ProblemDescription(Pr-Refroidissement-Alternateur-Hydrogène, ?Ds) ^ hasAlarm(Pr-Refroidissement-Alternateur-Hydrogène, ?Al) ^ ProblemFrequency(Pr-Refroidissement-Alternateur-Hydrogène, ?Fr) -&gt; sqwrl:select(Pr-Refroidissement-Alternateur-Hydrogène, ?ID, ?Ds, ?Tp, ?Fr, ?Al)</i></p>
SQWRL.SRH.1-3	<p><u>Diagnostiquer la fausse alarme</u></p> <p><i>Alarm(Al_SRH_1) ^ AlarmState(Al_SRH_1, 1) ^ AlarmValue(Al_SRH_1, false) ^ Problem(Pr-Refroidissement-Alternateur-Hydrogène) ^ hasAlarm(Pr-Refroidissement-Alternateur-Hydrogène, Al_SRH_1) ^ Cause(FaultyAutomatic) ^ Cause(FaultyEquipment) ^ hasCause(Pr-Refroidissement-Alternateur-Hydrogène, FaultyAutomatic) ^ hasCause(Pr-Refroidissement-Alternateur-Hydrogène, FaultyEquipment) ^ Component(?Cp) ^ Context(?Ct) ^ Failure(?FL) ^ Function(?F) ^ ComponentID(?Cp, ?CpID) ^ ComponentType(?Cp, ?Cpt) ^ ContextType(?Ct, ?Ctp) ^ ContextID(?Ct, ?CtID) ^ FailureID(?FL, ?FID) ^ FulfilmentRate(?FL, ?Rate) ^ FunctionID(?Ft, ?FtID) ^ FunctionType(?Ft, ?FtT) -&gt; sqwrl:select(Pr-Refroidissement-Alternateur-Hydrogène, ?Cp, ?CpID, ?Cpt, ?Ct, ?CtID, ?Ctp, ?FL, ?FID, ?Rate, ?Ft, ?FtID, ?FtT)</i></p>
SQWRL.SRH.1-4	<p><u>Identification les caractéristiques du problème</u></p> <p><i>Problem(Pr-Refroidissement-Alternateur-Hydrogène) ^ ProblemFeature(?PF) ^ hasFeature(Pr-Refroidissement-Alternateur-Hydrogène, ?PF) ^ I-Features(?PF, ?If) ^ P-Features(?PF, ?Pf) -&gt; sqwrl:select(Pr-Refroidissement-Alternateur-Hydrogène, ?PF, ?If, ?Pf) ^ sqwrl:columnNames("Problem ", "ProblemFeature", "InformationelFeature", "PhysicalFeature")</i></p>
SQWRL.SRH.1-5	<p><u>Identification de l’équipement et localisation de problème</u></p> <p><i>ManufacturingEquipment(Alternateur) ^ EquipmentID(Alternateur, ?ID) ^ EquipmentConstructor(Alternateur, ?EC) ^ EquipConstructionDate(Alternateur, ?CD) ^ autogen0:EquipLocation(Alternateur, ?EL) ^ EquipmentType(Alternateur, ?ET) -&gt; sqwrl:select(Alternateur, ?ID, ?EC, ?CD, ?EL, ?ET) ^ sqwrl:columnNames("Alternateur ", "EquipmentID", "EquipmentConstructor", "EquipConstructionDate", "EquipLocation", "EquipmentType")</i></p>
SQWRL.SRH.1-6	<p><u>Identification les caractéristiques des équipements lies au problème</u></p> <p><i>ManufacturingEquipment(Alternateur) ^ EquipmentFeature(?EF) ^ hasFeature(Alternateur, ?EF) ^ P-Features(?EF, ?Pf) -&gt; sqwrl:select(Alternateur, ?EF, ?Pf) ^ sqwrl:columnNames("Equipement", "EquipmentFeature", "Physical Feature")</i></p>
SQWRL.SRH.1-7	<p><u>Identification les agents pour identidies les causes de problème</u></p> <p><i>Problem(Pr-Refroidissement-Alternateur-Hydrogène) ^ ExpertAgent(?EA) ^ identifies(?EA, Pr-Refroidissement-Alternateur-Hydrogène) ^ AgentAge(?EA, ?Ag) ^ AgentID(?EA, ?ID) ^ AgentDiploma(?EA, ?D) ^ AgentExperience(?EA, ?Ex) ^ AgentFunction(?EA, ?Fn) ^ AgentName(?EA, ?N) -&gt; sqwrl:select(Pr-Refroidissement-Alternateur-Hydrogène, ?EA, ?N, ?ID, ?Ex, ?Fn, ?D, ?Ag) ^ sqwrl:columnNames("Problem", "Agent intervenant", "name", "AgentID", "Experience", "Function", "Diploma", "Age")</i></p>
SQWRL.SRH.1-8	<p><u>Identification les caractéristiques des experts pour identidies les causes de problème</u></p> <p><i>Problem(AlarmDiagnosis:Pr-Refroidissement-Alternateur-Hydrogène) ^ AgentFeature(?AF) ^ ExpertAgent(?EA) ^ AgentName(?EA, ?N) ^ identifies(?EA, Pr-Refroidissement-Alternateur-Hydrogène) ^ hasFeature(?EA, ?AF) ^ I-Features(?AF, ?If) ^ P-Features(?AF, ?Pf) -&gt;</i></p>



	<i>sqwrl:select(Pr-Refroidissement-Alternateur-Hydrogène, ?EA, ?N, ?AF, ?Pf, ?If) ^ sqwrl:columnNames("Problem", "Expert intervenant", "Name", "ExpertFeature", "P-Features", "I-Features")</i>
SQWRL.SRH.1-9	<b>Identification les causes possibles de problème</b> <i>Problem(Pr-Refroidissement-Alternateur-Hydrogène) ^ Alarm(Al_SRH_1) ^ hasAlarm(Pr-Refroidissement-Alternateur-Hydrogène, Al_SRH_1) ^ FaultyEquipment(?FE) ^ hasCause(Pr-Refroidissement-Alternateur-Hydrogène, ?FE) ^ Component(?Cp) ^ hasComponent(?FE, ?Cp) -&gt; sqwrl:select(Pr-Refroidissement-Alternateur-Hydrogène, Al_SRH_1, ?FE, ?Cp) ^ sqwrl:columnNames("Problem", "Alarm", "Faulty Equipment", "caused by")</i>
SQWRL.SRH.1.10	<b>Identification les agents intervenant pour Identification de problème</b> <i>Problem(Pr-Refroidissement-Alternateur-Hydrogène) ^ hasAlarm(Pr-Refroidissement-Alternateur-Hydrogène, Al_SRH_1) ^ ManufacturingAgent(?A) ^ AgentName(?A, ?N) ^ AgentFunction(?A, ?F) ^ identifies(?A, Pr-Refroidissement-Alternateur-Hydrogène) -&gt; sqwrl:select(AlarmDiagnosis:Pr-Refroidissement-Alternateur-Hydrogène, Al_SRH_1, ?A, ?N, ?F) ^ sqwrl:columnNames("Problem", "Alarm", "Identification By", "Name", "Function")</i>
SQWRL.SRH.1.11	<b>Proposition la solution liée au problème</b> <i>Component(?Cp) ^ hasAlarm(Pr-Refroidissement-Alternateur-Hydrogène, Al_SRH_1) ^ Solution(?Sol) ^ FaultyEquipment(?FE) ^ hasComponent(?FE, ?Cp) ^ hasCause(Pr-Refroidissement-Alternateur-Hydrogène, ?FE) ^ hasSolution(?Cp, ?Sol) ^ SolutionDescription(?Sol, ?Ds) -&gt; sqwrl:select(AlarmDiagnosis:Pr-Refroidissement-Alternateur-Hydrogène, Al_SRH_1, ?FE, ?Cp, ?Sol, ?Ds) ^ sqwrl:columnNames("Problem", "Alarm", "Cause", "Component Failure", "proposed solution", "Description Solution")</i>
SQWRL.SRH.1.12	<b>Identification le type d'intervention</b> <i>Solution(?Sol) ^ InterventionType(?I) ^ hasInterventionType(?Sol, ?I) ^ Alarm(Al_SRH_1) ^ SolutionDescription(?Sol, ?Ds) ^ interventionType(?I, ?Itp) -&gt; sqwrl:select(AlarmDiagnosis:Al_SRH_1, ?Sol, ?Ds, ?I, ?Itp) ^ sqwrl:columnNames("Alarm", "Solution", "Description Solution", "Intervention Type", "Intervention Description")</i>
SQWRL.SRH.1.13	<b>Identification la procédure de la solution</b> <i>Solution(?Sol) ^ ProceduralSolution(?PS) ^ hasProceduralSolution(?Sol, ?PS) ^ Alarm(Al_SRH_1) ^ SolutionDescription(?Sol, ?Ds) ^ CorrectiveAction(?PS, ?ca) ^ PreventiveAction(?PS, ?pa) -&gt; sqwrl:select(AlarmDiagnosis:Al_SRH_1, ?Sol, ?Ds, ?PS, ?pa, ?ca) ^ sqwrl:columnNames("Alarm", "Solution", "Description Solution", "Procedural Solution", "Preventive Action Description", "Corrective Action Description")</i>
SQWRL.SRH.1.14	<b>Identification les agents intervenant à la solution</b> <i>ManufacturingAgent(?A) ^ AgentName(?A, ?N) ^ AgentFunction(?A, ?F) ^ Solution(?Sol) ^ isProposedBy(?Sol, ?A) -&gt; sqwrl:select(AlarmDiagnosis:?Sol, ?A, ?N, ?F) ^ sqwrl:columnNames("Solution", "Proposed By Agent", "Agent Name", "Agent Function")</i>

Name	Query	Comment
SWRL.R10.Generic	AlarmDiagnosis:hasProblem(?E, ?Pr) ^ AlarmDiagnosis:ManufacturingSolution(?Sol) ^ AlarmDiagnosis:ManufacturingEquipment(?E) ^ AlarmDiagnosis:ManufacturingA...	Configuration and Affection rule
SWRL.R2.Generic	AlarmDiagnosis:ManufacturingProblem(?Pr) ^ AlarmDiagnosis:ManufacturingFeature(?F) -> AlarmDiagnosis:hasFeature(?Pr, ?F)	Define_problem_Feature
SWRL.R3.Generic	AlarmDiagnosis:hasProblem(?E, ?Pr) ^ AlarmDiagnosis:ManufacturingEquipment(?E) ^ AlarmDiagnosis:ManufacturingAgent(?A) ^ AlarmDiagnosis:ManufacturingProbl...	Define_ProblemFeature/EquipmentFeatu...
SWRL.R4.Generic	AlarmDiagnosis:hasProblem(?E, ?Pr) ^ AlarmDiagnosis:ManufacturingEquipment(?E) ^ AlarmDiagnosis:ManufacturingProblem(?Pr) -> AlarmDiagnosis:causes(?E, ?Pr)	Define_Causes_Of_Problem
SWRL.R5.Generic	AlarmDiagnosis:ManufacturingAgent(?A) ^ AlarmDiagnosis:ManufacturingProblem(?Pr) -> AlarmDiagnosis:identifies(?A, ?Pr)	DefineAgentforidentifiesproblem
SWRL.R6.Generic	AlarmDiagnosis:ManufacturingSolution(?Sol) ^ AlarmDiagnosis:ManufacturingAgent(?A) ^ AlarmDiagnosis:ManufacturingProblem(?Pr) -> AlarmDiagnosis:proposes(?A, ...	Proposition of Solution by Agent
SWRL.R7.Generic	AlarmDiagnosis:ManufacturingSolution(?Sol) ^ AlarmDiagnosis:ManufacturingAgent(?A) ^ AlarmDiagnosis:ManufacturingProblem(?Pr) ^ AlarmDiagnosis:collaborateWit...	Identification of Problem and proposition S...
Select.3	AlarmDiagnosis:FaultyAutomatic(?fa) ^ AlarmDiagnosis:hasComponent(?fa, ?Cp) ^ AlarmDiagnosis:Component(?Cp) -> sqwrl:select(?fa, ?Cp)	
Select.4	AlarmDiagnosis:Component(?Cp) ^ AlarmDiagnosis:ComponentID(?Cp, ?CpID) ^ AlarmDiagnosis:ComponentType(?Cp, ?Cpt) -> sqwrl:select(?Cp, ?CpID, ?Cpt) ^ sq...	
Select1	AlarmDiagnosis:Problem(AlarmDiagnosis:Pr-Refroidissement-Alternateur-Hydrogène) ^ AlarmDiagnosis:ProblemID(AlarmDiagnosis:Pr-Refroidissement-Alternateur-H...	
Sqwrl.1-5	AlarmDiagnosis:ManufacturingEquipment(AlarmDiagnosis:Alternateur) ^ AlarmDiagnosis:EquipmentFeature(?EF) ^ AlarmDiagnosis:hasFeature(AlarmDiagnosis:Altern...	selecting feature equipment
Sqwrl.1-6	AlarmDiagnosis:ManufacturingEquipment(AlarmDiagnosis:Alternateur) ^ AlarmDiagnosis:EquipmentID(AlarmDiagnosis:Alternateur, ?ID) ^ AlarmDiagnosis:Equipment...	selecting information alternateur
Sqwrl.1-7	AlarmDiagnosis:Problem(AlarmDiagnosis:Pr-Refroidissement-Alternateur-Hydrogène) ^ AlarmDiagnosis:ExpertAgent(?EA) ^ AlarmDiagnosis:identifies(?EA, AlarmDiag...	selecting information of agent expert
Sqwrl.1-4	AlarmDiagnosis:Problem(AlarmDiagnosis:Pr-Refroidissement-Alternateur-Hydrogène) ^ AlarmDiagnosis:ProblemFeature(?PF) ^ AlarmDiagnosis:hasFeature(AlarmDi...	
Sqwrl.1-8	AlarmDiagnosis:Problem(AlarmDiagnosis:Pr-Refroidissement-Alternateur-Hydrogène) ^ AlarmDiagnosis:AgentFeature(?AF) ^ AlarmDiagnosis:ExpertAgent(?EA) ^ Alar...	selecting information of feature expert
sqwrl.1-9	AlarmDiagnosis:Problem(AlarmDiagnosis:Pr-Refroidissement-Alternateur-Hydrogène) ^ AlarmDiagnosis:Alarm(AlarmDiagnosis:Al_SRH_1) ^ AlarmDiagnosis:hasAlar...	selection cause of problem

Problem	Alarm	Faulty Equipment	caused by
AlarmDiagnosis:Pr-Refroidissement-Alternateur-Hydrogène	AlarmDiagnosis:Al_SRH_1	AlarmDiagnosis:FaultyEquip.SRH.1	AlarmDiagnosis:Etancheité-aux-gaz-du-TurboAlternateur
AlarmDiagnosis:Pr-Refroidissement-Alternateur-Hydrogène	AlarmDiagnosis:Al_SRH_1	AlarmDiagnosis:faulityEquip.SRH.2	AlarmDiagnosis:Système-de-Gaz

**Figure 4.22** Sélection des causes possibles du problème lié à l’alarme Al\_SRH\_1 par la requête SQWRL.

Name	Query	Comment
SQWRL.SRH.1.faise.6	AlarmDiagnosis.partOf(?PS, ?Sol) ^ AlarmDiagnosis.CorrectiveAction(?Sol, ?CA) ^ AlarmDiagnosis.SolutionDescription(?Sol, ?DS) ^ AlarmDiagnosis.partOf(?IT, ?Sol) ^	selection solution and procedural solution
SWRL.R1.Generc	AlarmDiagnosis.ManufacturingProblem(?Pr) ^ AlarmDiagnosis.ManufacturingTask(?T) ^ AlarmDiagnosis.hasTask(?P, ?T) ^ AlarmDiagnosis.ManufacturingProcess(?P).	Rule generic ManufacturingProcess/Manu.
SWRL.R10.Generc	AlarmDiagnosis.hasProblem(?E, ?Pr) ^ AlarmDiagnosis.ManufacturingSolution(?Sol) ^ AlarmDiagnosis.ManufacturingEquipment(?E) ^ AlarmDiagnosis.ManufacturingA.	Configuration and affectation rule
SWRL.R2.Generc	AlarmDiagnosis.ManufacturingProblem(?Pr) ^ AlarmDiagnosis.ManufacturingFeature(?F) ^ AlarmDiagnosis.hasFeature(?Pr, ?F)	Define_problem_Feature
SWRL.R3.Generc	AlarmDiagnosis.hasProblem(?E, ?Pr) ^ AlarmDiagnosis.ManufacturingEquipment(?E) ^ AlarmDiagnosis.ManufacturingAgent(?A) ^ AlarmDiagnosis.ManufacturingProbl.	Define_ProblemFeatureEquipmentFeatu.
SWRL.R4.Generc	AlarmDiagnosis.hasProblem(?E, ?Pr) ^ AlarmDiagnosis.ManufacturingEquipment(?E) ^ AlarmDiagnosis.ManufacturingProblem(?Pr) -> AlarmDiagnosis.causes(?E, ?Pr)	Define_Causes_Of_Problem
SWRL.R5.Generc	AlarmDiagnosis.ManufacturingAgent(?A) ^ AlarmDiagnosis.ManufacturingProblem(?Pr) -> AlarmDiagnosis.identifies(?A, ?Pr)	DefineAgent for identifies problem
SWRL.R6.Generc	AlarmDiagnosis.ManufacturingSolution(?Sol) ^ AlarmDiagnosis.ManufacturingAgent(?A) ^ AlarmDiagnosis.ManufacturingProblem(?Pr) -> AlarmDiagnosis.proposes(?A.	Proposition of Solution by Agent
SWRL.R7.Generc	AlarmDiagnosis.ManufacturingSolution(?Sol) ^ AlarmDiagnosis.ManufacturingAgent(?A) ^ AlarmDiagnosis.ManufacturingProblem(?Pr) ^ AlarmDiagnosis.colaborateWith.	Identification of Problem and proposition S.
Select3	AlarmDiagnosis.FaultyAutomatis(?Fa) ^ AlarmDiagnosis.hasComponent(?Ca, ?Ccp) ^ AlarmDiagnosis.Component(?Cp) -> sqwrl:select(?Fa, ?Ccp)	
Select4	AlarmDiagnosis.Component(?Cp) ^ AlarmDiagnosis.ComponentID(?Cid, ?Coid) ^ AlarmDiagnosis.ComponentType(?Ctp, ?Ctpid) -> sqwrl:set(?Cp, ?Cpid, ?Ctpid) sq.	
Select1	AlarmDiagnosis.Problem(AlarmDiagnosis.Pr-Refroidissement-Alternateur-Hydrogene) ^ AlarmDiagnosis.ProblemID(AlarmDiagnosis.Pr-Refroidissement-Alternateur-H...	
Sqwrl.1-10	AlarmDiagnosis.Problem(AlarmDiagnosis.Pr-Refroidissement-Alternateur-Hydrogene) ^ AlarmDiagnosis.hasAlarm(AlarmDiagnosis.Pr-Refroidissement-Alternateur-Hy...	selection agent identification of cause prob.
Sqwrl.1-5	AlarmDiagnosis.ManufacturingEquipment(AlarmDiagnosis.Alternateur) ^ AlarmDiagnosis.EquipmentFeature(?EF) ^ AlarmDiagnosis.hasFeature(AlarmDiagnosis.Altern...	selecting feature equipment
Sqwrl.1-6	AlarmDiagnosis.ManufacturingEquipment(AlarmDiagnosis.Alternateur) ^ AlarmDiagnosis.EquipmentID(AlarmDiagnosis.Alternateur, ?ID) ^ AlarmDiagnosis.Equipment...	selecting information alternateur
Sqwrl.1-7	AlarmDiagnosis.Problem(AlarmDiagnosis.Pr-Refroidissement-Alternateur-Hydrogene) ^ AlarmDiagnosis.ExpertAgent(?EA) ^ AlarmDiagnosis.identifies(?EA, AlarmDiag...	selecting information of agent expert
Sqwrl.SHR.1-11	AlarmDiagnosis.Component(?Cp) ^ AlarmDiagnosis.hasAlarm(AlarmDiagnosis.Pr-Refroidissement-Alternateur-Hydrogene, AlarmDiagnosis.AI_SRH_1) ^ AlarmDiagn...	selection solution of problem
Sqwrl.SHR.1-12	AlarmDiagnosis.Solution(?Sol) ^ AlarmDiagnosis.interventionType(?T) ^ AlarmDiagnosis.hasInterventionType(?Sol, ?T) ^ AlarmDiagnosis.Alarm(AlarmDiagnosis.AI_SRH...	selection intervention type of solution
Sqwrl.SRH.1-13	AlarmDiagnosis.Solution(?Sol) ^ AlarmDiagnosis.ProceduralSolution(?PS) ^ AlarmDiagnosis.hasProceduralSolution(?Sol, ?PS) ^ AlarmDiagnosis.Alarm(AlarmDiagno...	
Sqwrl.1-4	AlarmDiagnosis.Problem(AlarmDiagnosis.Pr-Refroidissement-Alternateur-Hydrogene) ^ AlarmDiagnosis.ProblemFeature(?PF) ^ AlarmDiagnosis.hasFeature(AlarmDI...	
Sqwrl.1-8	AlarmDiagnosis.Problem(AlarmDiagnosis.Pr-Refroidissement-Alternateur-Hydrogene) ^ AlarmDiagnosis.AgentFeature(?AF) ^ AlarmDiagnosis.ExpertAgent(?EA) ^ Alar...	selecting information of feature expert

Alarm	Solution	Description Solution	Procedural Solution	Preventive Action Description	Corrective Action Description
AlarmDiagnosis.AI_SRH_1	AlarmDiagnosis.Sol.SRH.1	Mettre sous Pression le corp de Stator ensem.	AlarmDiagnosis.ProcSol1	Vérifier l'étanchéité de fermeture de robinets sur... lors de l'arrêt suivant	Mettre sous pression le corp de stator en
AlarmDiagnosis.AI_SRH_1	AlarmDiagnosis.Sol.SRH.2	A la différence nominale Huile Hydrogène	AlarmDiagnosis.ProcSol2	Vérifier l'absence de fuite dans la ligne de retour... A la différence nominale Huile_Hydrogène	Effectuer la révision

**Figure 4.23** Sélection de la procédure de la solution par la requête SQWRL.

### 4.5 Transfert des règles SWRL génériques en OWL vers le langage RDF

Dans l'objectif de reformulation des requêtes par le langage d'interrogation SPARQL qui représente le langage standard d'interrogation de graphes RDF, SPARQL est, d'une part, un protocole et un langage de requête permettant l'accès aux données RDF. Il est aussi un protocole d'accès comme un service Web. En ce sens, on peut transférer des règles SWRL génériques en OWL vers le langage RDF, une base de connaissances OWL peut alors être écrite comme une collection de triplets RDF, publiés dans le Web en langage RDF, sous forme de triplets RDF (sujet -> prédicat -> objet). L'exemple ci-dessous représente le transfert d'une règle générique *Règle.Gén.1* de langage OWL/SWRL vers le langage RDF. Par exemple, la formulation de concept « ManufacturingProcess » dans le langage OWL/SWRL est *ManufacturingProcess(?P)*, dans le langage RDF le concept est représenté par le triplet (*?P rdf:type ns:ManufacturingProcess*). La représentation de propriété « hasTask » par une représentation binaire en langage OWL *hasTask(?P, ?T)*, cette relation est représenté en RDF par les trois triplets (*?P rdf:type ns:ManufacturingProcess*), (*?T rdf:type ns:ManufacturingTask*), (*?P ns: hasTask ?T*). Toutes les règles génériques SWRL de MATO en langage RDF sont illustrées dans le tableau 4.10.

**Règle.Gén.1:** *ManufacturingProcess(?P) ^ ManufacturingTask(?T) ^ ManufacturingProblem(?Pr) ^ hasTask(?P, ?T) -> hasProblem(?T, ?Pr)*

**Règle.Gén.RDF.1:** *[rule1: (?P rdf:type ns:ManufacturingProcess) (?Pr rdf:type ns:ManufacturingProblem) (?T rdf:type ns:ManufacturingTask) (?P ns: hasTask ?T) -> (?T ns:hasProblem ?Pr)]*

**Tableau 4.10** les règles générique SWRL de MATO en langage RDF.

Régle.Gén.1	<u>Identification de problème, le processus et la tâche lié au problème</u> [rule1: (?P rdf:type ns:ManufacturingProcess) (?Pr rdf:type ns:ManufacturingProblem) (?T rdf:type ns:ManufacturingTask) (?P ns: hasTask ?T) -> (?T ns:hasProblem ?Pr)]
Régle.Gén.2	<u>Identification les caractéristiques du problème</u> [rule2: (?Pr rdf:type ns:ManufacturingProblem) (?F rdf:type ns:ManufacturingFeature) -> (?Pr ns: hasFeature ?F)]
Régle.Gén.3	<u>Identification les caractéristiques de problème, de l'équipement et de l'agent</u> [rule3: (?Pr rdf:type ns:ManufacturingProblem) (?F rdf:type ns:ManufacturingFeature) (?A rdf:type ns:ManufacturingAgent) (?E rdf:type ns: ManufacturingEquipment) (?E ns:hasProblem ?Pr)] -> (?Pr ns: hasFeature ?F) (?A ns: hasFeature ?F) (?E ns: hasFeature ?F)]
Régle.Gén.4	<u>Identification les causes possibles de problème</u> [rule4: (?Pr rdf:type ns:ManufacturingProblem) (?E rdf:type ns: ManufacturingEquipment) (?E ns:hasProblem ?Pr)] -> (?E ns: causes ?F)]
Régle.Gén.5	<u>identifier les agents intervenant au problème</u> [rule5: (?Pr rdf:type ns:ManufacturingProblem) (?A rdf:type ns: ManufacturingAgent) -> (?A ns: identifies ?Pr)]
Régle.Gén.6	<u>Proposer la solution liée au problème</u> [rule6: (?Pr rdf:type ns:ManufacturingProblem) (?E rdf:type ns: ManufacturingEquipment) (?Sol rdf:type ns: ManufacturingSolution) (?E ns: hasProblem ?Pr) (?E ns: causes ?Pr) -> (?Pr ns: hasSolution ?Sol)]
Régle.Gén.7	<u>Identifier le type d'intervention</u> [rule7: (?Pr rdf:type ns:ManufacturingProblem) (?E rdf:type ns: ManufacturingEquipment) (?Sol rdf:type ns: ManufacturingSolution) (?IT rdf:type ns: InterventionType) (?E ns: causes ?Pr) (?Pr ns: hasSolution ?Sol) -> (?Sol ns: hasInterventionType ?IT)]
Régle.Gén.8	<u>Identifier la procédure de la solution</u> [rule8: (?Pr rdf:type ns:ManufacturingProblem) (?Sol rdf:type ns: ManufacturingSolution) (?PS rdf:type ns: ProceduralSolution) (?E ns: causes ?Pr) (?Pr ns: hasSolution ?Sol) -> (?Sol ns: hasProceduralSolution ?PS)]
Régle.Gén.9	<u>Identification les agents intervenant à la solution</u> [rule9: (?Pr rdf:type ns:ManufacturingProblem) (?A rdf:type ns: ManufacturingAgent) (?Sol rdf:type ns: ManufacturingSolution) (?Pr ns: hasSolution ?Sol) -> (?A ns: proposes ?Sol)]
Régle.Gén.10	<u>La règle inverse de la Régle.Gén.9</u> [rule10: (?Pr rdf:type ns:ManufacturingProblem) (?A rdf:type ns: ManufacturingAgent) (?Sol rdf:type ns: ManufacturingSolution) (?Pr ns: hasSolution ?Sol) -> (?Sol ns: isProposedBy ?A)]

#### 4.6 Conception des requêtes SPARQL à partir des requêtes SQWRL

**SOWRL.SRH.1-1:** Alarm(Al\_SRH\_1) ^ AlarmID(Al\_SRH\_1, ?ID) ^ AlarmDescription(Al\_SRH\_1, ?AD) ^ AlarmValue(Al\_SRH\_1, ?V) ^ AlarmState(Al\_SRH\_1, ?ST) -> sqwrl:select(Al\_SRH\_1, ?ID, ?AD, ?V, ?ST) ^ sqwrl:orderBy(?ID)

**SOWRL.SRH.1-2:** Problem(Pr-Refroidissement-Alternateur-Hydrogène) ^ ProblemID(Pr-Refroidissement-Alternateur-Hydrogène, ?ID) ^ ProblemType(Pr-Refroidissement-Alternateur-Hydrogène, ?Tp) ^ ProblemDescription(Pr-Refroidissement-Alternateur-Hydrogène, ?Ds) ^ hasAlarm(Pr-Refroidissement-Alternateur-Hydrogène, ?Al) ^ ProblemFrequency(Pr-Refroidissement-Alternateur-Hydrogène, ?Fr) -> sqwrl:select(Pr-Refroidissement-Alternateur-Hydrogène, ?ID, ?Ds, ?Tp, ?Fr, ?Al)

**SPARQL (step.1.1)**

```
PREFIX ns: <http://www.semanticweb.org/userr/ontologies/2020/2/AlarmDiagnosis#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT ?Al ?As ?D ?ID ?N ?T ?Tp
WHERE { ?D rdf:type ns:Diagnosis .
        ?Al rdf:type ns:Alarm .
        ?Al ns:AlarmState ?AS .
        ?D ns:TaskID ?ID .
        ?D ns:TaskName ?N .
        ?D ns:TaskTime ?T .
        ?D ns:TaskType ?Tp .
        FILTER regex (?Al, "^Al_SRH_1" ) }
```

**SPARQL (step.1.2)**

```
PREFIX ns: <http://www.semanticweb.org/userr/ontologies/2020/2/AlarmDiagnosis#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

SELECT ?Al ?ID ?AD ?V ?ST
WHERE {
    ?Al rdf:type ns:Alarm .
    ?Al ns:AlarmID ?ID .
    ?Al ns:AlarmDEscription ?AD .
    ?Al ns:AlarmValue ?V .
    ?Al ns:AlarmState ?ST .
    FILTER regex (?Al, "^Al_SRH_1" ) }
```

**Tableau 4.11** Conception des requêtes SPARQL à partir des requêtes SQWRL.

Requêtes SQWRL	Requêtes SPARQL
<p><b>SQWRL.SRH.1-1:</b> Alarm(<i>Al_SRH_1</i>) ^ AlarmID(<i>Al_SRH_1</i>, ?<i>ID</i>) ^ AlarmDEscription(<i>Al_SRH_1</i>, ?<i>AD</i>) ^ AlarmValue(<i>Al_SRH_1</i>, ?<i>V</i>) ^ AlarmState(<i>Al_SRH_1</i>, ?<i>ST</i>) -&gt; sqwrl:select(<i>Al_SRH_1</i>, ?<i>ID</i>, ?<i>AD</i>, ?<i>V</i>, ?<i>ST</i>) ^ sqwrl:orderBy(?<i>ID</i>)</p>	<p><b>SPARQL.SRH.1-1:</b>                      PREFIX ns: &lt;http://www.semanticweb.org/userr/ontologies/2020/2/AlarmDiagnosis#&gt;                      PREFIX rdf: &lt;http://www.w3.org/1999/02/22-rdf-syntax-ns#&gt;                      SELECT ?Al ?ID ?AD ?V ?ST                      WHERE {                          ?Al rdf:type ns:Alarm .                          ?Al ns:AlarmID ?ID .                          ?Al ns:AlarmDEscription ?AD .                          ?Al ns:AlarmValue ?V .                          ?Al ns:AlarmState ?ST .                          FILTER regex (?Al, "^Al_SRH_1" ) }</p>
<p><b>SQWRL.SRH.1-2:</b> Problem(<i>Pr-Refroidissement-Alternateur-Hydrogène</i>) ^ ProblemID(<i>Pr-Refroidissement-Alternateur-Hydrogène</i>, ?<i>ID</i>) ^ ProblemType(<i>Pr-Refroidissement-Alternateur-Hydrogène</i>, ?<i>Tp</i>) ^ ProblemDescription(<i>Pr-Refroidissement-Alternateur-Hydrogène</i>, ?<i>Ds</i>) ^ hasAlarm(<i>Pr-Refroidissement-Alternateur-Hydrogène</i>, ?<i>Al</i>) ^ ProblemFrequency(<i>Pr-Refroidissement-Alternateur-Hydrogène</i>, ?<i>Fr</i>) -&gt; sqwrl:select(<i>Pr-Refroidissement-Alternateur-Hydrogène</i>, ?<i>ID</i>, ?<i>Ds</i>, ?<i>Tp</i>, ?<i>Fr</i>, ?<i>Al</i>)</p>	<p><b>SPARQL.SRH.1-2:</b>                      PREFIX ns: &lt;http://www.semanticweb.org/userr/ontologies/2020/2/AlarmDiagnosis#&gt;                      PREFIX rdf: &lt;http://www.w3.org/1999/02/22-rdf-syntax-ns#&gt;                      SELECT ?Pr ?ID ?Ds ?Tp ?Fr ?Al                      WHERE { ?Pr rdf:type ns:Problem .                          ?Al rdf:type ns:Alarm .                          ?Pr ns: hasAlarm ns:Al .                          ?Pr ns: ProblemID ?ID .                          ?Pr ns: ProblemDescription ?Ds .                          ?Pr ns: ProblemType ?Tp .                          ?Pr ns: ProblemType ?Tp .                          ?Pr ns: ProblemFrequency ?Fr .                          FILTER regex (?Al, "^Al_SRH_1" ) }</p>
<p><b>SQWRL.SRH.1-3:</b> Alarm(<i>Al_SRH_1</i>) ^ AlarmState(<i>Al_SRH_1</i>, 1) ^ AlarmValue(<i>Al_SRH_1</i>, false) ^ Problem(<i>Pr-Refroidissement-Alternateur-Hydrogène</i>) ^ hasAlarm(<i>Pr-Refroidissement-Alternateur-Hydrogène</i>, <i>Al_SRH_1</i>) ^ Cause(<i>FaultyAutomatic</i>) ^ Cause(<i>FaultyEquipment</i>) ^</p>	<p><b>SPARQL.SRH.1-3</b>                      PREFIX ns: &lt;http://www.semanticweb.org/userr/ontologies/2020/2/AlarmDiagnosis#&gt;                      PREFIX rdf: &lt;http://www.w3.org/1999/02/22-rdf-syntax-ns#&gt;                      SELECT ?Al ?ID ?V ?ST ?Pr ?FE ?FA</p>



**Chapitre 4 : Application du processus de capitalisation et de partage de connaissances au niveau de la centrale thermique**

<p><i>hasCause(Pr-Refroidissement-Alternateur-Hydrogène, FaultyAutomatic) ^ hasCause(Pr-Refroidissement-Alternateur-Hydrogène, FaultyEquipment) ^ Component(?Cp) ^ Context(?Ct) ^ Failure(?FL) ^ Function(?F) ^ ComponentID(?Cp, ?CpID) ^ ComponentType(?Cp, ?Cpt) ^ ContextType(?Ct, ?Ctp) ^ ContextID(?Ct, ?CtID) ^ FailureID(?FL, ?FID) ^ FulfilmentRate(?FL, ?Rate) ^ FunctionID(?Ft, ?FtID) ^ FunctionType(?Ft, ?FtT) -&gt; sqwrl:select(Pr-Refroidissement-Alternateur-Hydrogène, ?Cp, ?CpID, ?Cpt, ?Ct, ?CtID, ?Ctp, ?FL, ?FID, ?Rate, ?Ft, ?FtID, ?FtT)</i></p>	<p>WHERE {  ?Al rdf:type ns:Alarm .  ?Al ns:AlarmID ?ID .  ?Al ns:AlarmValue ?V .  ?Al ns:AlarmState ?ST .  ?Pr rdf:type ns:Problem .  ?Pr ns:hasAlarm ?Al .  ?FA rdf:type ns:FaultyAutomatic .  ?FE rdf:type ns:FaultyEquipment .  ?Pr ns:hasCause ?FA .  ?Pr ns:hasCause ?FE .  FILTER regex (?Al, "^AL_SRH_1" ) }</p>
<p><b><u>SOWRL.SRH.1-4:</u></b>  <i>ManufacturingEquipment(Alternateur) ^ EquipmentID(Alternateur, ?ID) ^ EquipmentConstructor(Alternateur, ?EC) ^ EquipConstructionDate(Alternateur, ?CD) ^ EquipLocation(Alternateur, ?EL) ^ EquipmentType(Alternateur, ?ET) -&gt; sqwrl:select(Alternateur, ?ID, ?EC, ?CD, ?EL, ?ET) ^ sqwrl:columnNames("Alternateur ", "EquipmentID", "EquipmentConstructor", "EquipConstructionDate", "EquipLocation", "EquipmentType")</i></p>	<p><b><u>SPARQL.SRH.1-4:</u></b>  PREFIX ns:  &lt;http://www.semanticweb.org/userr/ontologies/2020/2/AlarmDiagnosis#&gt;  PREFIX rdf: &lt;http://www.w3.org/1999/02/22-rdf-syntax-ns#&gt;  SELECT ?MainEquipment ?EquipmentID  ?EquipmentConstructor ?EquipLocation ?EquipmentType  WHERE { ?MainEquipment rdf:type ns:MainEquipment .  ?MainEquipment ns:EquipmentID ?EquipmentID.  ?MainEquipment ns:EquipmentConstructor  ?EquipmentConstructor.  ?MainEquipment ns:EquipConstructionDate  ?EquipConstructionDate.  ?MainEquipment ns:EquipLocation  ?EquipLocation.  ?MainEquipment ns:EquipmentType  ?EquipmentType. }</p>
<p><b><u>SOWRL.SRH.1-5:</u></b> <i>Problem(Pr-Refroidissement-Alternateur-Hydrogène) ^ ExpertAgent(?EA) ^ identifies(?EA, Pr-Refroidissement-Alternateur-Hydrogène) ^ AgentAge(?EA, ?Ag) ^ AgentID(?EA, ?ID) ^ AgentDiploma(?EA, ?D) ^ AgentExperience(?EA, ?Ex) ^ AgentFunction(?EA, ?Fn) ^ AgentName(?EA, ?N) -&gt; sqwrl:select(Pr-Refroidissement-Alternateur-Hydrogène, ?EA, ?N, ?ID, ?Ex, ?Fn, ?D, ?Ag) ^ sqwrl:columnNames("Problem", "Agent intervenant", "name", "AgentID", "Experience", "Function", "Diploma", "Age")</i></p>	<p><b><u>SPARQL.SRH.1-5</u></b>  PREFIX ns:  &lt;http://www.semanticweb.org/userr/ontologies/2020/2/AlarmDiagnosis#&gt;  PREFIX rdf: &lt;http://www.w3.org/1999/02/22-rdf-syntax-ns#&gt;  SELECT ?Pr ?AgentIntervenant ?Name ?AgentID ?Age ?Diploma ?Experience ?Function  WHERE { ?Pr rdf:type ns:Problem .  ?AgentIntervenant rdf:type ns:ExpertAgent.  ?AgentIntervenant ns:identifies ?Pr .  ?AgentIntervenant ns:AgentName ?Name .  ?AgentIntervenant ns:AgentID ?AgentID.  ?AgentIntervenant ns:AgentAge ?Age .  ?AgentIntervenant ns:AgentDiploma ?Diploma .  ?AgentIntervenant ns:AgentExperience ?Experience .  ?AgentIntervenant ns:AgentFunction ?Function . }</p>
<p><b><u>SOWRL.SRH.1-6:</u></b>  <i>Problem(Pr-Refroidissement-Alternateur-Hydrogène) ^ AgentFeature(?AF) ^ ExpertAgent(?EA) ^ AgentName(?EA, ?N) ^ identifies(?EA, Pr-Refroidissement-Alternateur-Hydrogène) ^ hasFeature(?EA, ?AF) ^ I-Features(?AF, ?If) ^ P-Features(?AF, ?Pf) -&gt; sqwrl:select(Pr-Refroidissement-Alternateur-Hydrogène, ?EA, ?N, ?AF, ?Pf, ?If) ^ sqwrl:columnNames("Problem", "Expert intervenant", "Name", "ExpertFeature", "P-Features", "I-Features")</i></p>	<p><b><u>SPARQL.SRH.1-6</u></b>  PREFIX ns:  &lt;http://www.semanticweb.org/userr/ontologies/2020/2/AlarmDiagnosis#&gt;  PREFIX rdf: &lt;http://www.w3.org/1999/02/22-rdf-syntax-ns#&gt;  SELECT ?Pr ?ExpertIntervenant ?ExpertFeature ?Name ?IFeatures ?PFeatures  WHERE { ?Pr rdf:type ns:Problem .  ?AgentIntervenant rdf:type ns:ExpertAgent.  ?ExpertFeature rdf:type ns:AgentFeature .  ?AgentIntervenant ns:identifies ?Pr .  ?AgentIntervenant ns:hasFeature ?ExpertFeature .  ?AgentIntervenant ns:AgentName ?Name .  ?ExpertFeature ns:I-Features ?IFeatures .  ?ExpertFeature ns:P-Features ?PFeatures . }</p>
<p><b><u>SOWRL.SRH.1-7:</u></b> <i>Problem(Pr-Refroidissement-Alternateur-Hydrogène) ^ Alarm(AL_SRH_1) ^ hasAlarm(Pr-Refroidissement-Alternateur-Hydrogène, AL_SRH_1) ^ FaultyEquipment(?FE) ^ hasCause(Pr-Refroidissement-Alternateur-Hydrogène, ?FE) ^ Component(?Cp) ^ hasComponent(?FE, ?Cp) -&gt;</i></p>	<p><b><u>SPARQL.SRH.1-7</u></b>  PREFIX ns:  &lt;http://www.semanticweb.org/userr/ontologies/2020/2/AlarmDiagnosis#&gt;  PREFIX rdf: &lt;http://www.w3.org/1999/02/22-rdf-syntax-ns#&gt;  SELECT ?Problem ?FE ?ComponentFailer</p>



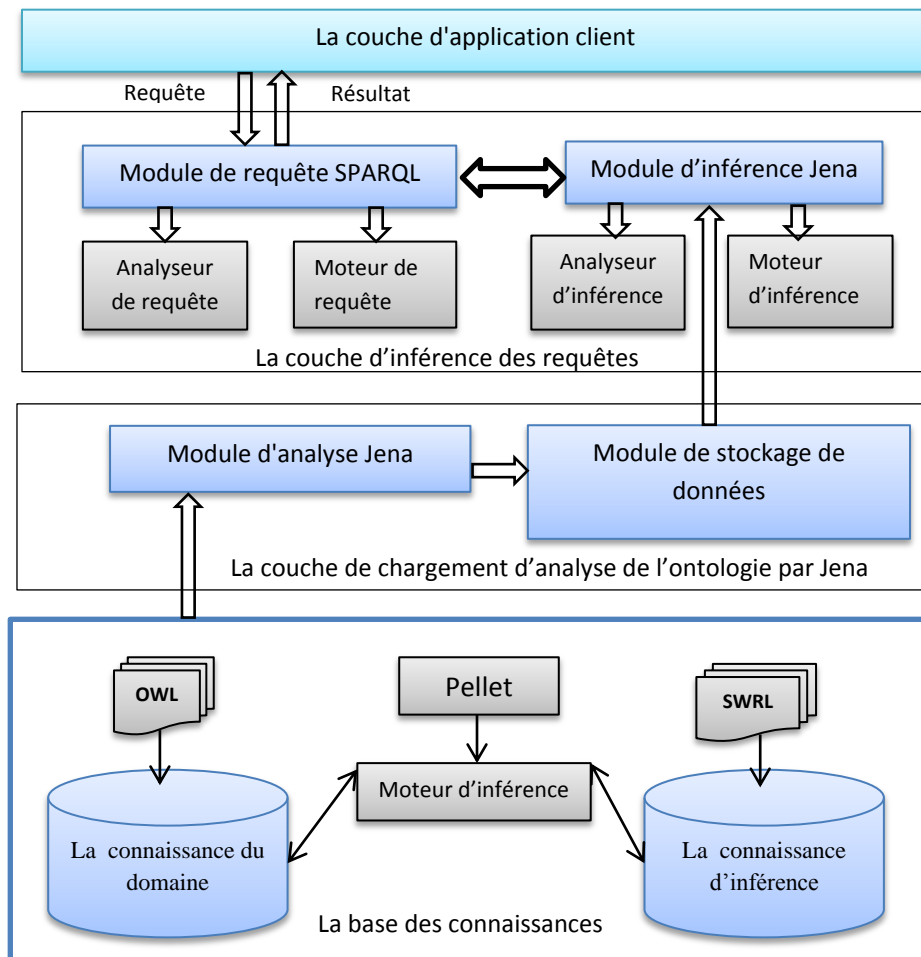
**Chapitre 4 : Application du processus de capitalisation et de partage de connaissances au niveau de la centrale thermique**

<p><i>sqwrl:select(Pr-Refroidissement-Alternateur-Hydrogène, Al_SRH_1, ?FE, ?Cp) ^ sqwrl:columnNames("Problem", "Alarm", "Faulty Equipment", "caused by")</i></p>	<p><i>WHERE {?Problem rdf:type ns:problem . ?FE rdf:type ns:Faultyequipment. ?Problem ns:hasCause ?FE. ?Componentfailer rdf:type ns:Component . ?FE ns:hasComponent ?Componentfailer . }</i></p>
<p><b><u>SOWRL.SRH.1-8 :</u></b> <i>Problem(Pr-Refroidissement-Alternateur-Hydrogène) ^ hasAlarm(Pr-Refroidissement-Alternateur-Hydrogène, Al_SRH_1) ^ ManufacturingAgent(?A) ^ AgentName(?A, ?N) ^ AgentFunction(?A, ?F) ^ identifies(?A, Pr-Refroidissement-Alternateur-Hydrogène) -&gt; sqwrl:select(AlarmDiagnosis:Pr-Refroidissement-Alternateur-Hydrogène, Al_SRH_1, ?A, ?N, ?F) ^ sqwrl:columnNames("Problem", "Alarm", "Identification By", "Name", "Function")</i></p>	<p><b><u>SPARQL.SRH.1-8</u></b>  <b>PREFIX ns:</b>  <i>&lt;http://www.semanticweb.org/userr/ontologies/2020/2/AlarmDiagnosis#&gt;</i>  <b>PREFIX rdf:</b> <i>&lt;http://www.w3.org/1999/02/22-rdf-syntax-ns#&gt;</i>  <b>PREFIX xsd:</b> <i>&lt;http://www.w3.org/2001/XMLSchema#&gt;</i>    <i>SELECT ?Pr ?alarm ?AgentIntervenant ?Name ?Function  WHERE { ?alarm rdf:type ns:Alarm .  ?alarm ns:AlarmID "AL-SRH-01"^^xsd:NMTOKEN .  ?Pr ns:hasAlarm ?alarm .  ?Pr rdf:type ns:Problem .  ?AgentIntervenant rdf:type ns:ExpertAgent .  ?AgentIntervenant ns:identifies ?Pr .  ?AgentIntervenant ns:AgentName ?Name .  ?AgentIntervenant ns:AgentID ?AgentID .  ?AgentIntervenant ns:AgentFunction ?Function .  FILTER regex (?alarm, "AL_SRH_1" ) }</i></p>
<p><b><u>SOWRL.SRH.1-9 :</u></b> <i>Component(?Cp) ^ hasAlarm(Pr-Refroidissement-Alternateur-Hydrogène, Al_SRH_1) ^ Solution(?Sol) ^ FaultyEquipment(?FE) ^ hasComponent(?FE, ?Cp) ^ hasCause(Pr-Refroidissement-Alternateur-Hydrogène, ?FE) ^ hasSolution(?Cp, ?Sol) ^ SolutionDescription(?Sol, ?Ds) -&gt; sqwrl:select(AlarmDiagnosis:Pr-Refroidissement-Alternateur-Hydrogène, Al_SRH_1, ?FE, ?Cp, ?Sol, ?Ds) ^ sqwrl:columnNames("Problem", "Alarm", "Cause", "Component Failure", "proposed solution", "Description Solution")</i></p>	<p><b><u>SPARQL.SRH.1-9</u></b>  <b>PREFIX ns:</b>  <i>&lt;http://www.semanticweb.org/userr/ontologies/2020/2/AlarmDiagnosis#&gt;</i>  <b>PREFIX rdf:</b> <i>&lt;http://www.w3.org/1999/02/22-rdf-syntax-ns#&gt;</i>  <i>SELECT ?Pr ?FE ?componentFailure ?Sol ?DSolution  WHERE { ?Pr rdf:type ns:Problem .  ?FE rdf:type ns:FaultyEquipment .  ?componentFailur rdf:type ns:Component .  ?Sol rdf:type ns:Solution .  ?Pr ns:hasCause ?causes .  ?FE ns:hasComponent ?componentFailur .  ?componentFailur ns:hasSolution ?Sol .  ?Solution ns:SolutionDescription ?DSolution . }</i></p>
<p><b><u>SOWRL.SRH.1-10:</u></b> <i>Solution(?Sol) ^ InterventionType(?I) ^ hasInterventionType(?Sol, ?I) ^ Alarm(Al_SRH_1) ^ SolutionDescription(?Sol, ?Ds) ^ interventionType(?I, ?Itp) -&gt; sqwrl:select(AlarmDiagnosis:Al_SRH_1, ?Sol, ?Ds, ?I, ?Itp) ^ sqwrl:columnNames("Alarm", "Solution", "Description Solution", "Intervention Type", "Intervention Description")</i></p>	<p><b><u>SPARQL.SRH.1-10</u></b>  <b>PREFIX ns:</b>  <i>&lt;http://www.semanticweb.org/userr/ontologies/2020/2/AlarmDiagnosis#&gt;</i>  <b>PREFIX rdf:</b> <i>&lt;http://www.w3.org/1999/02/22-rdf-syntax-ns#&gt;</i>  <i>SELECT ?Sol ?DS ?I ?Itp  WHERE {?Sol rdf:type ns:Solution .  ?I rdf:type ns:InterventionType .  ?Sol ns:SolutionDescription ?DS .  ?Sol ns:hasInterventionType ?I .  ?I ns:interventionType ?Itp .  }</i></p>
<p><b><u>SOWRL.SRH.1-11:</u></b> <i>Solution(?Sol) ^ ProceduralSolution(?PS) ^ hasProceduralSolution(?Sol, ?PS) ^ Alarm(Al_SRH_1) ^ SolutionDescription(?Sol, ?Ds) ^ CorrectiveAction(?PS, ?ca) ^ PreventiveAction(?PS, ?pa) -&gt; sqwrl:select(AlarmDiagnosis:Al_SRH_1, ?Sol, ?Ds, ?PS, ?pa, ?ca) ^ sqwrl:columnNames("Alarm", "Solution", "Description Solution", "Procedural Solution", "Preventive Action Description", "Corrective Action Description")</i></p>	<p><b><u>SPARQL.SRH.1-11</u></b>  <b>PREFIX ns:</b>  <i>&lt;http://www.semanticweb.org/userr/ontologies/2020/2/AlarmDiagnosis#&gt;</i>  <b>PREFIX rdf:</b> <i>&lt;http://www.w3.org/1999/02/22-rdf-syntax-ns#&gt;</i>  <i>SELECT ?Sol ?DS ?PS ?pa ?ca  WHERE {?Sol rdf:type ns:Solution .  ?PS rdf:type ns:ProceduralSolution .  ?Sol ns:SolutionDescription ?DS .  ?Sol ns:hasProceduralSolution ?PS .  ?PS ns:PreventiveAction ?pa .  ?PS ns:PreventiveAction ?ca .  }</i></p>
<p><b><u>SOWRL.SRH.1-12</u></b>  <i>ManufacturingAgent(?A) ^ AgentName(?A, ?N) ^ AgentFunction(?A, ?F) ^ Solution(?Sol) ^</i></p>	<p><b><u>SPARQL.SRH.1-12</u></b>  <b>PREFIX ns:</b>  <i>&lt;http://www.semanticweb.org/userr/ontologies/2020/2/AlarmDiagnosis#&gt;</i></p>

<pre>isProposedBy(?Sol, ?A) -&gt; sqwrl:select(AlarmDiagnosis:?Sol, ?A, ?N, ?F) ^ sqwrl:columnNames("Solution", "Proposed By Agent", "Agent Name", "Agent Function")</pre>	<pre>iagnosis#&gt; PREFIX rdf: &lt;http://www.w3.org/1999/02/22-rdf-syntax-ns#&gt; SELECT ?Sol ?A ?N ?F WHERE {?Sol rdf:type ns:Solution . ?A rdf:type ns:ManufacturingAgent . ?A ns:AgentName ?N . ?A ns:AgentFunction ?F . ?Sol ns:isProposedBy ?A. ?PS ns:PreventiveAction ?pa . ?PS ns:PreventiveAction ?ca . }</pre>
--	---

### 4.7 Implémentation de l'ontologie MATO pour le diagnostic du système d'alarme

Une application web PPASD (Power Plant Alarm System Diagnosis) est réalisée conformément aux exigences réelles du processus de la centrale thermique et l'architecture du système client/serveur conventionnel est adoptée. La structure de PPASD est illustrée à la Figure. 4.24.



**Figure 4.24** les règles génériques SWRL de MATO en langage RDF.

Ce système comprend la base de connaissances de bas niveau et l'application interactive de haut niveau. La base de connaissances de bas niveau est la partie centrale du PPASD, qui est utilisée pour définir et former le modèle d'ontologie MATO de sélection des étapes nécessaires à la tâche de diagnostic ainsi que l'ensemble de règles de sélection des connaissances. Le moteur d'inférence Pellet implémente l'inférence de connaissances et le test de cohérence de la bibliothèque d'ontologies MATO et l'ensemble de règles de sélection de l'équipement. L'application interactive de haut niveau est l'implémentation du PPASD, qui est utilisée pour analyser la base de connaissances de bas niveau et effectuer une requête d'inférence de données analysées.

Les résultats de la requête sont finalement renvoyés à l'interface client. Donc, l'application d'interaction de niveau supérieur comprend la couche de stockage d'analyse, la couche de requête d'inférence et la couche d'application cliente

#### **4.8 Architecture de l'application développée**

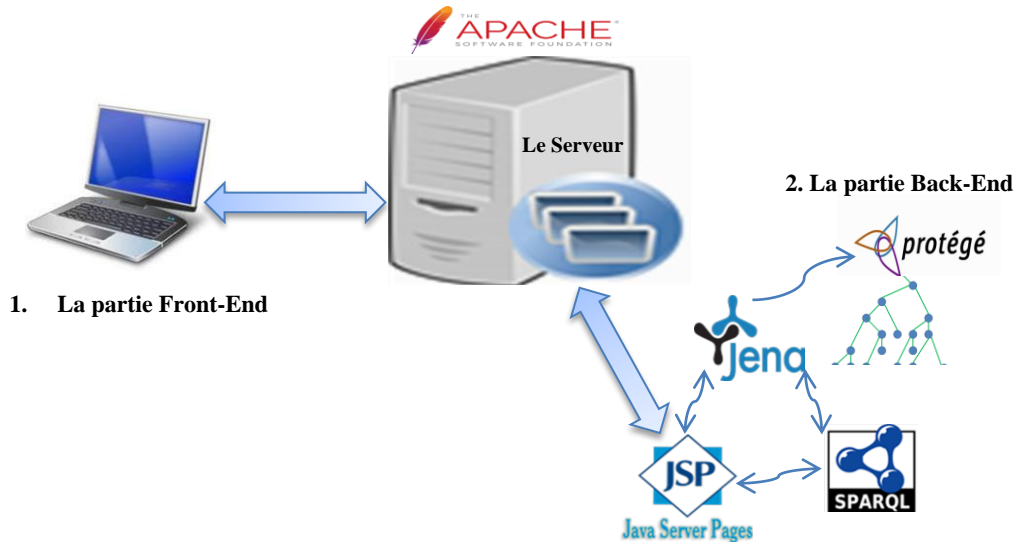
Dans cette section, la figure 4.25 illustre l'architecture globale du système et inclut les composants de base du système et les interactions entre eux. Les composants peuvent être divisés en deux catégories, la partie Front-End (FE) et la partie Back-End (BE), tandis que la communication entre les deux, est établie via le serveur Apache Tomcat.

Les technologies et les outils utilisés dans le système:

- L'éditeur d'ontologie open source **Protégé**<sup>1</sup> ([Horridge et al., 2004](#)), avec le raisonneur Pellet ([Glimm et al., 2010](#)).
- Le langage de requête d'ontologie **SPARQL**<sup>2</sup> ([W3C SPARQL Working Groupe, 2008](#)).
- **JSP**<sup>3</sup> (Java Server Pages), la technologie Java populaire pour créer du contenu Web dynamique.
- **Jena**<sup>4</sup>, un framework Java pour construire des applications Web Sémantique qui sert de lien entre un code OWL et un programme JAVA et que la majorité des applications de Web sémantique sont développées en Jena.

- Serveur **Apache Tomcat**<sup>5</sup> et conteneur de servlets pour l'hébergement de l'application Web.

Un aperçu des technologies impliquées et du rôle qu'elles jouent dans le projet est affiché dans le tableau 2.



**Figure 4.25** L'architecture globale du système et les composants de base.

**Tableau 4.12** Les technologies et les outils utilisés dans le système.

Les outils utilisés dans le système	Les technologies utilisées
L'éditeur d'ontologie	Protégé (V.5.5.0)
Le langage de l'ontologie	OWL
le raisonneur de l'ontologie	Pellet
Le langage de requête d'ontologie	SPARQL
La partie Front-End	JSP(V.2.2)
La librairie	Jena(V.3.14.0)
Le serveur web	Apache Tomcat (V.9.0.3)

La partie FE se compose essentiellement de l'interface utilisateur, c'est la partie cliente du programme, où l'utilisateur saisit l'entrée initiale du système (les entrées : le problème, l'équipement et l'alarme) et récupère les résultats.

1 Protégé : <https://protege.stanford.edu/>

2 SPARQL : <https://www.w3.org/2009/01/sparql-charter>

3 Java Server Pages (JSP): <https://jsp.java.net/>.

4 Jena Semantic Web framework: <http://jena.apache.org/index.html>.

5 Apache Tomcat: <http://tomcat.apache.org/>

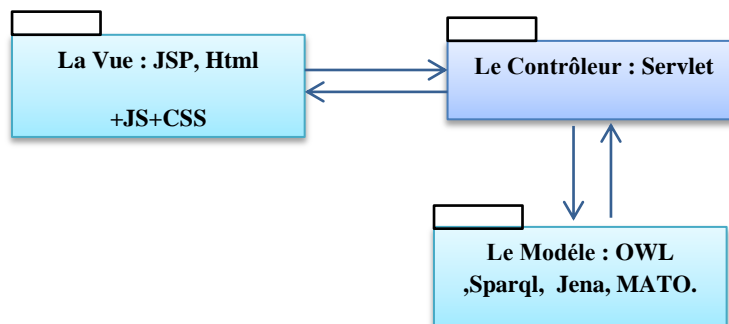
D'autre part, le BE se compose de l'ontologie elle-même, ainsi que des modules Jena. Les fichiers JSP appropriés sont responsables de l'affichage des informations à l'utilisateur et, également, de la communication avec Jena. Ce dernier est utilisé pour charger et manipuler l'ontologie et pour lui soumettre des requêtes SPARQL. Enfin, les résultats extraits permettent de partager les connaissances.

Après avoir généré le code OWL correspondant à notre ontologie on a été très vite confronté au problème de son exploitation dans un programme JAVA, surtout que peu d'articles sont ceux qui parlent de détail d'implémentation chose qui nous a demandé plus d'effort pour trouver une solution.

Le paradigme MVC est un schéma de programmation qui propose de séparer une application en trois parties :

- le modèle, qui contient la logique et l'état de l'application ;
- la vue, qui représente l'interface utilisateur ;
- le contrôleur, qui gère la synchronisation entre la vue et le modèle.

Notre système basé sur JEE, le modèle est assuré par des EJB et/ou des JavaBeans liés au modèle OWL de notre ontologie, le contrôleur est assuré par des servlets et la vue par des JSP, JSF, HTML+JS. Architecture de notre système MVC est expliqué sur La Figure 4.26.



**Figure 4.26** Le paradigme MVC pour notre système basé sur JEE.

L'application Web PPASD contient multi-pages, une page html « About » de description du système, une page principale « home » est disponible pour interroger la base de connaissance, ainsi d'une page JSP pour soumettre des données au contrôleur. La Figure 4.27 présente l'interface graphique du système PPASD en HTML +CSS+JS. La Figure 4.28 est une interface des paramètres d'entrées de système PPASD en JSP, la Figure 4.29 un exemple d'une étape 1.1 de système PPASD



Figure 4.27 L'interface graphique du système PPASD en HTML+CSS+JS

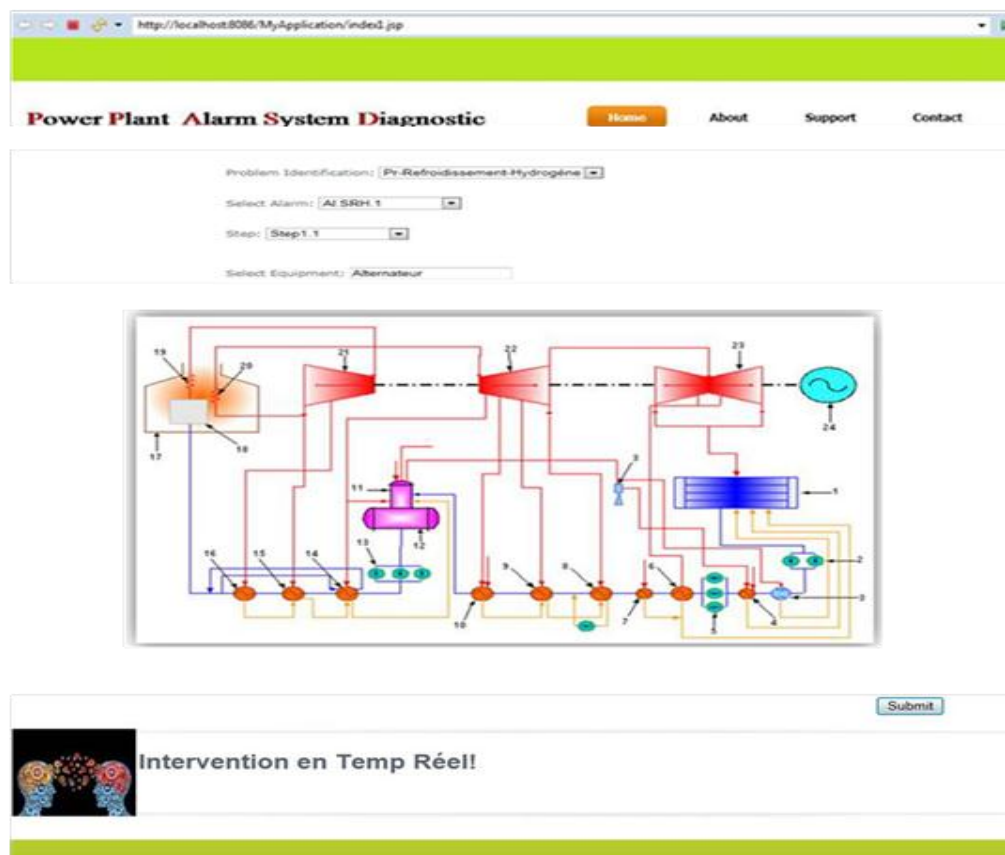


Figure 4.28 Interface des paramètres d'entrées de système PPASD en JSP



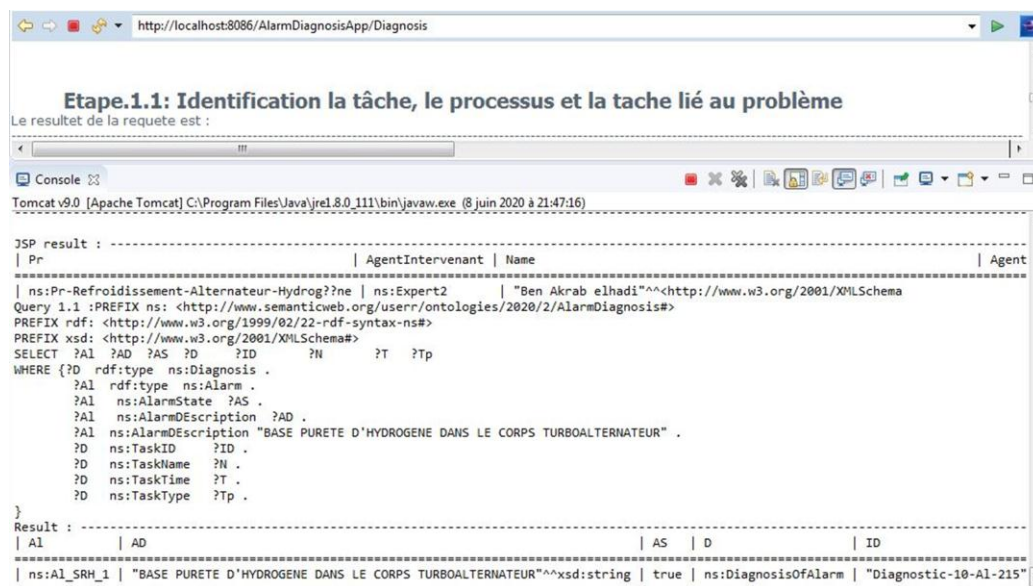


Figure 4.29 Exemple d'une étape 1.1 de système PPASD

## 4.9 Evaluation de l'ontologie MATO

L'évaluation de l'ontologie est la tâche de mesurer la qualité d'une ontologie selon des critères donnés. C'est «un jugement technique du contenu de l'ontologie par rapport à un référentiel à chaque phase et entre les phases de leur cycle de vie» (Gómez-Pérez ., 2004),

La validation de l'ontologie est une partie importante de l'évaluation de la qualité d'une ontologie, et généralement le seul moyen d'assurer la logique des connaissances encodées dans l'ontologie. Il existe un grand nombre d'approches de validation d'ontologie, (Vrandečić., 2009 ; Gómez-Pérez., 2004 ; Tartir et al., 2005 ; Vrandecic., 2010 ; Lozano-Telloand., 2004 ).

Nous avons choisi d'évaluer l'ontologie MATO en suivant différentes approches de validation.

- L'évaluation par la validation automatique
- La validation basée sur les métriques
- la validation basée sur les connaissances d'experts

### 4.9.1 Evaluation par la validation automatique

Nous avons utilisé deux types de validation, une validation offline par la Vérification de la consistance à l'aide de raisonneur Pellet<sup>6</sup>.

Cette méthode vérifie la cohérence interne d'une ontologie (le contenu ne contient pas d'informations contradictoires) et ignore ses connaissances de base. La deuxième est la validation de la structure par l'utilisation d'un évaluateur (scanner) on-line par OOPS !<sup>7</sup>

### a. La Vérification de la consistance à l'aide de raisonneur Pellet

La vérification des axiomes logiques est une tâche essentielle dans la validation de l'ontologie. En effet, cette validation garantit la cohérence des axiomes logiques. Cette satisfaction consiste à (i) vérifier l'encodage de la spécification, (ii) détecter des erreurs telles que des classes mal formées, des axiomes redondants, etc., et (iii) confirmer que l'ontologie MATO a été construite selon certains critères de qualité.

En termes de temps, Pellet l'un des raisonneurs les plus rapides dans la classification des ontologies. À l'aide de son plugin de raisonnement Pellet<sup>6</sup>, Protégé a automatiquement vérifié les classes et relations inférées ainsi que les hiérarchies, les erreurs et les problèmes sont marqués en rouge. De plus, le raisonnement Pellet a été utilisé pour vérifier l'exactitude des règles SWRL éditées dans SWRLTab (comme illustré à la Figure 4.30).

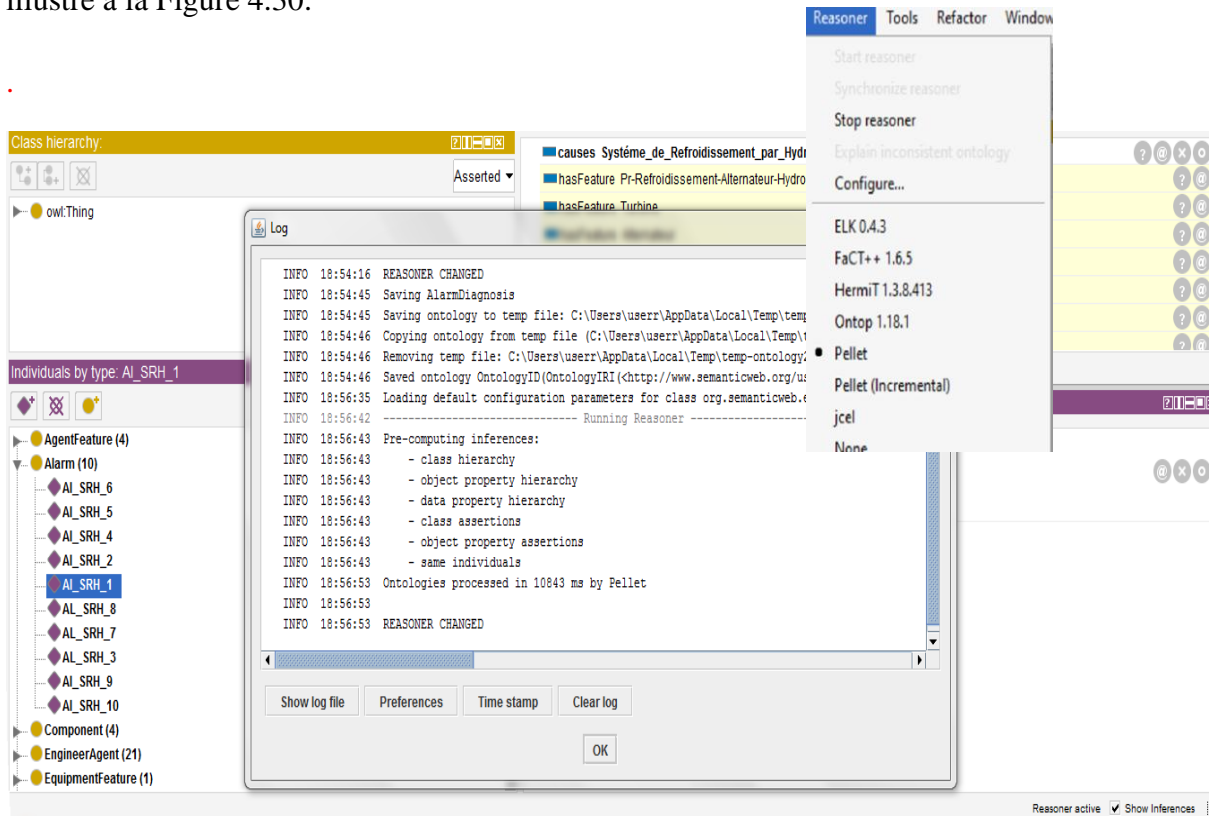


Figure 4.30 Les résultats de vérification de MATO par Pelett<sup>6</sup>

### b. La Validation de la structure avec OOPS !

Pour la validation de la structure de MATO, nous l'avons évaluée en la soumettant à OOPS!<sup>7</sup> (Ontology Pitfall Scanner!) (Poveda-Villal et al., 2014), un évaluateur d'ontologie à jour en ligne, gratuit et indépendant du module de développement de



l'ontologie, qui détecte certains des pièges les plus courants (trouvés dans la littérature) qui apparaissent lors du développement des ontologies.

Le scanner OOPS !<sup>7</sup> détecte la plupart des erreurs dans l'ontologie et suggère des améliorations. Ces pièges sont classés selon trois types: critiques, mineurs et importants.

- Critique: il est important de supprimer cet écueil. Cela peut affecter le raisonnement, la cohérence et l'applicabilité de l'ontologie.
- Important: Bien qu'il ne soit pas critique pour la fonction d'ontologie, il est important de supprimer ce type de piège.
- Mineur: Ce n'est pas vraiment un problème, mais en le corrigeant, cette ontologie deviendrait plus authentifiée.

Ces types sont en outre regroupés sur la base de certains critères tels que l'exactitude, la concision, la clarté, l'exhaustivité et la cohérence. Le résultat est une page Web sur laquelle sont répertoriées les pitfalls (les erreurs identifiées) accompagnées d'une proposition de résolution. La Figure 4.31 présente les résultats de vérification on line de MATO par OOPS !<sup>7</sup>.

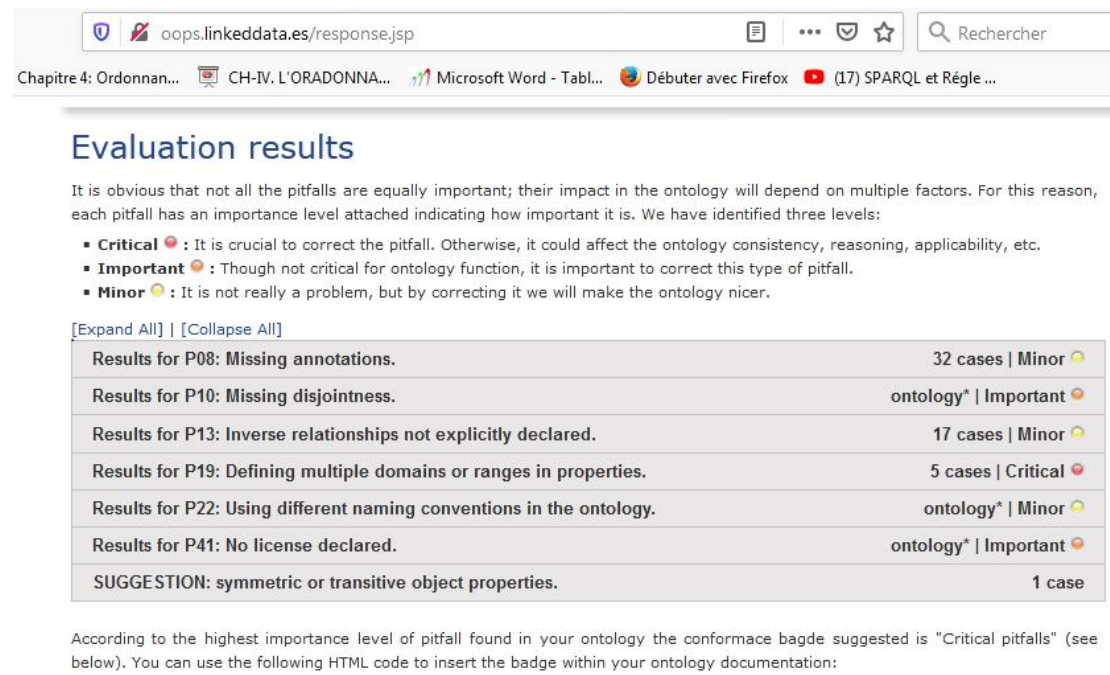


Figure 4.31 Les résultats de vérification on line de MATO par OOPS !<sup>7</sup>

6 Pellet : <https://www.w3.org/2001/sw/wiki/Pellet>

7 OOPS ! : <http://oops.linkeddata.es/>

8 OntoMetric : <https://ontometrics.informatik.uni-rostock.de>

Tableau 4.13 Les pièges déclarés dans MATO

Numéro et nom du Pitfal	Catégorie(s) du pitfal	Description du pitfal	Nombre de cas	Importance
P08 : Annotation manquante	Compréhension humaine	Les termes d'ontologie manquent de propriétés d'annotation qui amélioreraient la compréhension et l'utilisabilité de l'ontologie du point de vue de l'utilisateur	32	Mineure
P10 : Disjonction disparu	Manque de disjonction	L'ontologie manque d'axiomes disjoints entre les classes ou les propriétés qui devraient être définies comme disjointes.	ontologie*	Important
P13 : Relation inverse non-définie	Compréhension humaine Problèmes de modélisation	Lorsqu'une relation n'a pas de relation inverse définie.	17	Mineure
P19 : Permutation d'intersection et d'union	Compréhension humaine Consistance logique Problèmes de modélisation	Le domaine ou la plage (ou les deux) d'une propriété (relations et attributs) est défini en indiquant plusieurs instructions rdfs: domain ou rdfs: range.	5	Critique
P22 : Utilisation de différentes conventions de nommage	Compréhension humaine	Les éléments d'ontologie ne sont pas nommés selon la même convention	ontologie*	Mineure
P41 : Aucune licence déclarée	aucune licence déclarée	Les métadonnées d'ontologie omettent les informations sur la licence qui s'applique à l'ontologie.	ontologie*	Important

Le tableau 4.13 montre qu'il y a trois pièges mineurs, deux pièges importants et un piège critique dans MATO. Les pièges Mineur et les pièges importants ne sont pas des problèmes majeurs, mais le piège critique devrait être supprimé de l'ontologie.

Dans l'évaluation MATO, il y a une critique (P19) qui a été définitivement supprimée de l'ontologie. p19 (Définition de plusieurs domaines et plages dans les propriétés) qui est liée à l'erreur commune qui en résulte lors de la définition de plusieurs domaines.

Dans MATO, les propriétés *proposes* et *proposed-by* se forment dans une relation inverse, mais ont le même domaine et gamme. Nous avons changé le domaine et le rang des deux propriétés et supprimé cet écueil de l'ontologie.

Ils sont trois pièges mineurs (P08, P13 et P22) comme dans le tableau 5.2. Ces pièges mineurs ont été éliminés en appliquant une solution utile telle que l'ajout d'annotations.

#### **4.9.2 Validation basée sur les métriques**

L'outil OntoMetric<sup>8</sup> a été utilisé pour l'évaluation basée sur les métriques. Dans OntoMetric<sup>8</sup>, la validation basée sur les métriques est un outil Web qui valide et affiche des statistiques sur une ontologie donnée par le téléchargement la source d'ontologie enregistrée en tant que fichier \*.RDF ou \*.OWL ou la saisie d'URL de l'ontologie dans la zone de saisie. MATO est évalué sur la base de cinq mesures: les métriques de classe, les métriques de schéma, les métriques de base, les métriques de base de connaissances et les métriques de graphique. La figure 4.32 représente la soumission de l'ontologie par le simulateur online OntoMetric<sup>8</sup> pour l'évaluation de l'ontologie MATO. Toutes ces mesures métriques sont présentées dans le tableau 4.14.

- **Métrique de classe:** il examine les classes et les relations d'ontologie en mesurant l'importance de classe (CI), la connectivité de classe (CC) et le nombre d'instances de classe (CIC) d'une classe spécifique, la Richesse de l'héritage de classe (CIR), le nombre des sous classes (CCC), dans le tableau 4.14, toutes les mesures sont mesurées pour une classe « ManufacturingProblem » de MATO.
- **Métrique du schéma:** la métrique du schéma se concentre sur la conception de l'ontologie pour aborder la richesse des attributs (AR), la richesse d'héritage (IR) et la richesse des relations (RR) ainsi qu'un ratio d'axiomes et de classes (A/CR), l'équivalence (ER) et les relations inverses (IRR) d'une conception de schéma d'ontologie. Ces mesures sont mesurées pour le MATO et discutées dans le tableau 4.14.
- **Métrique de base:** les outils OntoMetric et Protégé sont utilisés pour mesurer la métrique de base. Les deux outils sont appliqués et les résultats sont similaires à ceux présentés dans le tableau 4.14. La métrique de base comprend une métrique simple telle que: nombre de classes, axiomes, objets, propriétés, Des particuliers et bien d'autres.
- **Métriques de base de connaissances:** la métrique de base de connaissances ou métrique d'instance représente une métrique qui exprime la base de connaissances dans son ensemble et définit la manière dont chaque classe est utilisée dans la base de connaissances. Cette métrique est calculée en mesurant la population moyenne (AP)

et la richesse de classe (CR). AP indique le nombre d'individus par rapport au nombre de classes.

La richesse de classe indique comment les instances sont organisées entre les classes.

- **Mesures de graphique:** cette mesure évalue la structure de l'ontologie, comme le nombre de nœuds racine, le nombre de nœuds feuille, le nombre de nœuds frères, la profondeur et la largeur. Le nombre total de nœuds racines est présenté par Absolute Root Cardinality, le nombre de nœuds feuilles est représenté par Absolute Leaf Cardinality et le nombre de nœuds frères est représenté par Absolute Sibling Cardinality. La propriété de profondeur est liée à la cardinalité des chemins tandis que la cardinalité des niveaux est mesurée par la propriété de largeur comme indiqué dans le tableau 4.14.

Upload a file

Parcourir... validation.owl Upload

```

<?xml version="1.0"?>
<Ontology xmlns="http://www.w3.org/2002/07/owl#"
  xml:base="http://www.semanticweb.org/user/ontologies/2020/2/AlarmDiagnosis"
  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/user/ontologies/2020/2/AlarmDiagnosis"
  versionIRI="http://www.semanticweb.org/user/ontologies/2020/2/AlarmDiagnosis"
/1.1">
  <Prefix name="" IRI="http://www.semanticweb.org/user/ontologies/2020/2/AlarmSystemDiagnosis#" />
  <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="ProblemData" IRI="http://www.semanticweb.org/user/ontologies/2020/2/AlarmSystemDiagnosis#ProblemData$" />
  <Prefix name="ProblemData1" IRI="http://www.semanticweb.org/user/ontologies/2020/2/AlarmDiagnosis#ProblemData$" />
  <Prefix name="AlarmDiagnosis" IRI="http://www.semanticweb.org/user/ontologies/2020/2/AlarmDiagnosis#" />

```

Choose the metrics, you would like to calculate.

Base metrics

Schema metrics

Knowledgebase metrics

Class metrics \*

Graph metrics

\* Note: checking class metrics may cause long calculation times depending on the size of the ontology

Calculate metrics

A. La soumission de l'ontologie MATO par OntoMetric<sup>8</sup>

**Results**

**OntologyID: Optional.of(http://www.semanticweb.org/user/ontologies/2020/2/AlarmDiagnosis)**

**Base metrics**    [more details](#)    [Hide](#)

Axioms:	1173
Logical axioms count:	867
Class count:	46
Total classes count:	46
Object property count:	26
Total object properties count:	26
Data property count:	79
Total data properties count:	79
Properties count:	105
Individual count:	126
Total individuals count:	126
DL expressivity:	ALEHI(D)

B. les résultats de l'évaluation de l'ontologie MATO par OntoMetric<sup>8</sup>

Figure 4.32 Les indicateurs d'évaluation de MATO par OntoMetric<sup>8</sup>

Tableau 4.14 Les résultats d'évaluation de MATO par OntoMetric

les Métriques	Les attributs	Les résultats de MATO
les Métriques de classe	Class connectivity:	5
	Class importance: (CI)	0.039683
	Class inheritance richness:(CIR)	11.5
les Métriques de base	Axioms	1173
	Logical axioms count	867
	Total classes count	46
	Total object properties count	26
	Total data properties count	79
	Properties count	105
	Total individuals count	126
Les Métriques de schéma	Attribute richness: (AR)	1.717391
	Inheritance richness: (IR)	0.891304
	Relationship richness: (RR)	0.4875
	Equivalence ratio: (ER)	0.282609
	Axiom/class ratio: (CR)	25.5
	Inverse relations ratio: (IRR)	0.076923
les Métriques de graphique	Class/relation ratio: (CR)	0.575
	Absolute root cardinality: (ARC)	7
	Absolute leaf cardinality(ALC)	34
	Absolute sibling cardinality:	46
	Absolute depth:	101
	Average depth:	2.104167
	Maximal depth:	4
	Absolute breadth:	48
	Average breadth:	3.692308
	Maximal breadth:	11
	Ratio of leaf fan-outness:	0.73913
	Ratio of sibling fan-outness:	1.0
	Tangledness:	0.043478
	Total number of paths:	48
Average number of paths:	12.0	
les Métriques de base de connaissances	Average population:	2.73913
	Class richness:	0.391304

### 4.9.3 Validation basée sur les connaissances d'expert

La validation basée sur les connaissances d'expert au domaine, repose sur une communication avec les experts de domaine d'application (ingénieurs, techniciens, opérateurs et aussi managers) ayant participé à la phase d'externalisation des connaissances tacites de modèle de capitalisation. La méthode interactive de validation que nous avons mise en place avec les acteurs du domaine,

Nous avons communiqué sur notre projet par la présentation de l'ontologie MATO proposée et aussi l'application PPASD dans la centrale thermique, etc. Suite à cela, nous avons mis notre ontologie à disposition des acteurs du domaine d'application.

Chaque acteur disposait de son propre ordinateur et donc d'un accès à l'ontologie MATO mise en ligne sur WEBPROTÉGÉ<sup>9</sup>. Nous leur avons ensuite proposé de se retrouver par petits groupes pour leur permettre de discuter ensemble, de manière interactive, de la modélisation et d'apporter leurs critiques sur les concepts, les termes de règles et de contraintes, de propriétés et d'attributs et résolutions en cas de désaccord et les différentes requêtes SPARQL pour la sélection des connaissances partagées durant l'exécution de la tâche.

Les experts de domaine étaient invités à laisser un commentaire en texte libre sur le site WEBPROTÉGÉ<sup>9</sup>, cela contribue à l'interaction entre les acteurs. Ceux non présents durant la séance de validation, peuvent avoir accès aux discussions et répondre ou y participer. Une fois ces recommandations posées, Ils interagissaient entre eux uniquement pour expliciter des choix de modélisation jugés ambigus par les experts. La figure 4.33 présente la mise en ligne de l'ontologie MATO sur le site on line WEBPROTÉGÉ<sup>9</sup> pour donner l'accès aux acteurs pour évaluer l'ontologie et laisser des commentaires.

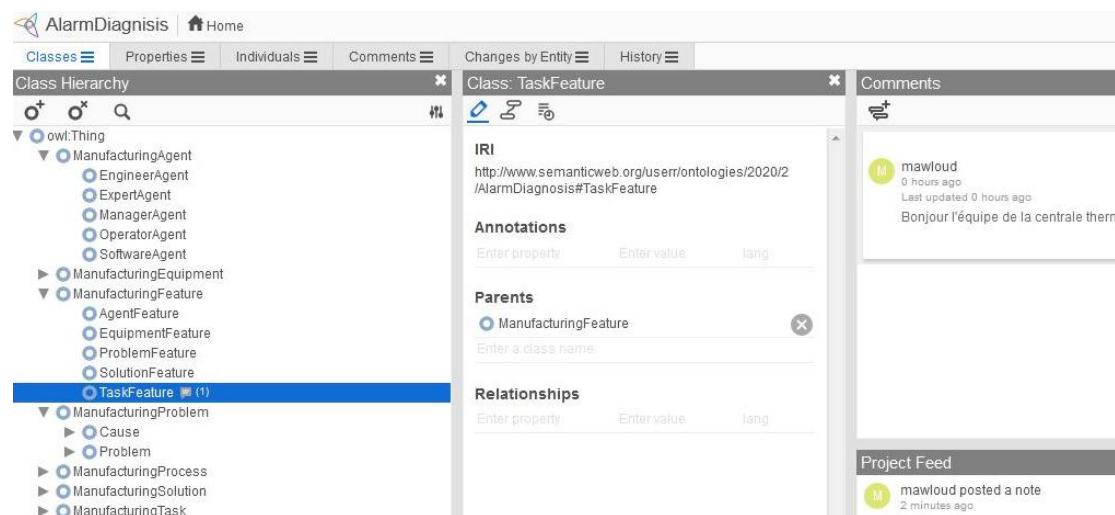


Figure 4.33 La mise en ligne de l'ontologie MATO sur le site on line WEBPROTÉGÉ<sup>9</sup>

<sup>9</sup> WEBPROTÉGÉ: <https://webprotege.stanford.edu/#projects>,

#### **4.10 Conclusion**

Dans ce chapitre, nous avons appliqué notre système de gestion des connaissances proposé au chapitre 3 sur une centrale thermique, ce système est composé de deux modèles, le premier est la capitalisation des connaissances KCMMT et le deuxième est le partage des connaissances par l'ontologie MATO dans une tâche industrielle.

Le premier modèle de cette application est de sauvegarder les connaissances appliquées dans l'industrie manufacturière, en utilisant une nouvelle technique et méthodologie d'externalisation des connaissances. Notre modèle contient deux processus principaux.

L'externalisation des connaissances et modélisation des connaissances. Dans la phase d'externalisation des connaissances : nous avons appliqué les cinq étapes.

L'identification des problèmes par la méthodologie CommonKADS. l'étape 2 : l'analyse des problèmes par diagramme de Pareto, l'étape 3 : diagnostiquer les problèmes analysés, l'étape 4 : appliquer les techniques de MACTAK, l'étape 5 : modèle explicite sous formes de règles de production. La deuxième phase est la phase de modélisation, nous avons utilisé deux types de modélisation, une modélisation à base d'une représentation ontologique pour la connaissance de la tâche et la connaissance du domaine et l'autre à base de connaissances d'experts pour la connaissance d'inférence. Ce modèle facilite la capture des connaissances expertes et transforme les connaissances tacites en explicites avec une maximisation des règles de production.

Dans le deuxième modèle, nous avons appliqué l'ontologie des tâches de fabrication MATO (MANufacturing Task Ontology), pour le diagnostic de système d'alarme et de système de production. Le système utilise tous les concepts importants du domaine de fabrication, du problème de fabrication, de la tâche de fabrication, du processus de fabrication, de l'agent de fabrication, de l'équipement de fabrication et de la fonction de fabrication., un exemple d'application détaillé pour le diagnostic d'alarme de système de refroidissement par hydrogène de turbo alternateur. La phase de modélisation utilisant le langage OWL, la phase de raisonnement par les règles SWRL et la phase d'implémentation des connaissances de domaine et d'inférence par l'éditeur Protégé. L'éditeur Protégé prend en charge les règles SWRL, comme le Pellet raisonneur et l'utilisation de l'API OWL. Enfin, nous avons utilisé JESS (Java



Expert System Shell) pour développer des connaissances d'inférence et nous avons utilisé JessTab, la combinaison entre la connaissance du domaine (MATO) et l'inférence du domaine (JESS).

Pour l'interrogation de notre ontologie, nous avons utilisé les requêtes SQWRL pour la sélection des connaissances de l'ontologie MATO par les requêtes SQWRL pour le diagnostic du système d'alarme.

Pour faciliter l'utilisation de notre modèle de gestion des connaissances au niveau de la centrale, une application accessible à tous les agents manufacturiers pour aider au diagnostic hebdomadaire de leur système d'alarme. Une application web PPSD (Power Plant Alarm System Diagnosis) est établie conformément aux exigences réelles du processus de la centrale thermique. Ce système comprend la base de connaissances de bas niveau (la base des connaissances) et l'application interactive de haut niveau (module de stockage de données, le module d'analyse, la couche d'inférence des requêtes, la couche d'application client). Pour illustrer cette application, nous avons utilisé le modèle MVC (le modèle qui contient la logique et l'état de l'application, la vue qui représente l'interface utilisateur et le contrôleur, qui gère la synchronisation entre la vue et le modèle).

Pour cela, nous avons utilisé le langage des requêtes SPARQL, SPARQL est une recommandation du W3C depuis janvier 2008. Il joue le rôle d'un pont entre les technologies du Web sémantique (dont RDF), et les plateformes Web déjà existantes. Il est une API universelle d'accès aux données.

Notre application web est basée sur Java JEE, le modèle est assuré par des EJB et/ou des JavaBeans liés au modèle OWL de notre ontologie, à cause de cette plateforme, nous avons conçu notre requête par les requêtes SPARQL à partir des requêtes SQWRL. Transférer les règles SWRL génériques en OWL vers le langage RDF.

Enfin, nous avons évalué la qualité de l'ontologie MATO en suivant différentes approches de validation, l'évaluation par la validation automatique, la validation basée sur les métriques et la validation basée sur les connaissances d'experts.



# Conclusion générale et perspectives

1. Bilan.....	163
2. Perspectives.....	165

## **Conclusion générale**

### **1. Bilan**

Dans cette thèse, nous nous sommes intéressés à un modèle de gestion des connaissances dans les entreprises industrielles, ce modèle contient deux processus de gestion, le premier est de capitalisation des connaissances, basé sur une tâche industrielle et le deuxième processus sur le partage des connaissances à base d'une ontologie des tâches manufacturières.

Concernant la capitalisation des connaissances, nous avons étudié deux phases importantes ; l'externalisation et la modélisation des connaissances, ce processus est spécifique à une tâche industrielle. Nous avons utilisé une combinaison de deux méthodologies : une méthodologie d'ingénierie des connaissances CommonKADS et une méthodologie d'externalisation des connaissances MACTAK. Dans la phase de modélisation, nous avons utilisé deux différentes techniques de modélisation, une modélisation basée sur l'expert et la deuxième une représentation ontologique. Pour le deuxième processus, nous avons développé un processus de partage basé sur l'Ontologie des Tâches Manufacturières, nous avons développé une ontologie de tâche manufacturière MATO en identifiant un ensemble des concepts de fabrication et leurs relations, Cet ontologie proposée facilite le partage des connaissances entre les tâches de fabrication et aide à capturer et à réutiliser les connaissances durant l'exécution des tâches.

Une application a été développée représente notre système de gestion des connaissances proposé à une centrale thermique, il est composé de deux modèles, le premier est la capitalisation et le deuxième est le partage des connaissances.

Le premier modèle de cette application est de sauvegarder les connaissances appliquées dans l'industrie manufacturière, en utilisant une nouvelle technique et méthodologie d'externalisation des connaissances. Notre modèle contient deux processus principaux: l'externalisation des connaissances et la modélisation des connaissances. Dans la phase d'externalisation des connaissances : nous avons appliqué les cinq étapes.

L'identification des problèmes par la méthodologie CommonKADS. L'étape 2 : l'analyse des problèmes par diagramme de Pareto, l'étape 3 : diagnostiquer les

problèmes analysés, l'étape 4 : appliquer les techniques de MACTAK, l'étape 5 : modèle explicite sous forme des règles de production. La deuxième phase est la phase de modélisation, nous avons utilisé deux types de modélisation, une modélisation à base d'une représentation ontologique pour la connaissance de la tâche et la connaissance de domaine et une autre à base des connaissances d'expert pour la connaissance d'inférence. Ce modèle facilite la capture des connaissances expertes et transforme les connaissances tacites en explicites avec une maximisation de règles de production.

Dans le deuxième modèle, nous avons appliqué l'ontologie des tâches de fabrication MATO (MANufacturing Task Ontology) pour le diagnostic de système d'alarme de système de production, le système utilise tous les concepts importants du domaine de fabrication, du problème de fabrication, de la tâche de fabrication, du processus de fabrication, de l'agent de fabrication, de l'équipement de fabrication et de la tâche de fabrication. Un exemple d'application détaillé pour le diagnostic d'alarme de système de refroidissement par hydrogène de turbo alternateur. La phase de modélisation utilisant le langage OWL, la phase de raisonnement par les règles SWRL et la phase d'implémentation des connaissances de domaine et d'inférence par l'éditeur Protégé. L'éditeur Protégé prend en charge les règles SWRL, comme le raisonneur Pellet et l'utilisation de l'API OWL.

Enfin, nous avons utilisé JESS (Java Expert System Shell) pour développer des connaissances d'inférence et nous avons utilisé JessTab pour la combinaison entre la connaissance du domaine (MATO) et l'inférence de domaine (JESS). Pour l'interrogation de notre ontologie, nous avons utilisé la requête SQWRL pour la sélection des connaissances de l'ontologie MATO par les requêtes SQWRL pour le diagnostic du système d'alarme.

Pour faciliter l'utilisation de notre processus de partage des connaissances au niveau de la centrale et élargir le champ d'utilisation, une application web PPASD (Power Plant Alarm System Diagnosis) est établie conformément aux exigences réelles du processus de la centrale thermique et accessible à tous les agents manufacturiers pour aider au diagnostic hebdomadaire de leur système d'alarme.

Ce système comprend la base de connaissances de bas niveau (la base des connaissances) et l'application interactive de haut niveau (module de stockage de données, le module d'analyse, la couche d'inférence des requêtes, la couche d'application client). Pour illustrer cette application, nous avons utilisé le modèle

MVC (le modèle qui contient la logique et l'état de l'application, la vue qui représente l'interface utilisateur et le contrôleur, qui gère la synchronisation entre la vue et le modèle).

Pour cela nous avons utilisé le langage des requêtes SPARQL qui joue le rôle d'un pont entre les technologies du Web sémantique (dont RDF), et les plateformes Web déjà existantes. Il est une API universelle d'accès aux données.

Notre application web est basée sur Java JEE, le modèle est assuré par des EJB et/ou des JavaBeans liés au modèle OWL de notre ontologie. Grâce à cette plateforme, nous avons conçu nos requêtes par SPARQL à partir des requêtes SQWRL et transféré les règles SWRL génériques en OWL vers le langage RDF.

Nous avons évalué la qualité de l'ontologie MATO en suivant différentes approches de validation, l'évaluation par la validation automatique, la validation basée sur les métriques et la validation basée sur les connaissances d'experts.

Nous sommes convaincus que ce modèle de gestion des connaissances manufacturier servira à conserver la mémoire de la centrale, la réalisation de système d'aide au diagnostic peut se révéler concrètement utile pour favoriser le partage de connaissances entre les experts et tous les agents d'exploitation et de gérer le savoir-faire opérationnel des experts de domaine.

## 2. Perspectives

Dans l'objectif de généraliser la culture de capitalisation et partage des connaissances dans nos entreprises industrielles, nous pouvons utiliser cette approche de capitalisation et de partage des connaissances pour toutes les tâches manufacturières et aussi dans plusieurs domaines industriels. Ceci permet d'élargir le champ d'utilisation de notre modèle proposé et construire un système de gestion des connaissances plus sophistiqué.

L'Automatisation de processus d'acquisition des connaissances dans une tâche de fabrication spécifique: cette perspective élimine le problème de la capture des connaissances.

Intégration de ce modèle dans un environnement industrie 4.0, avec la connexion de tous les objets manufacturiers par l'application web.

Construction d'un système de fabrication sémantique garantit l'interopérabilité des informations sémantiques au cours de l'évolution de processus de fabrication, ce système nécessite une évaluation dans plusieurs domaines de l'industrie. La réutilisation des ontologies manufacturières dans une bibliothèque intelligente en ligne est possible.

## Bibliographies

### A

- Asterhan, C. S., & Bouton, E. (2017). Teenage peer-to-peer knowledge sharing through social network sites in secondary schools. *Computers & Education*, 110, 16-34.
- Ameri, F., & Dutta, D. (2008). A matchmaking methodology for supply chain deployment in distributed manufacturing environments. *Journal of Computing and Information Science in Engineering*, 8(1).
- Ameri, F., Urbanovsky, C., & McArthur, C. (2012, July). A systematic approach to developing ontologies for manufacturing service modeling. In *Proceedings of the workshop on ontology and semantic web for manufacturing* (1- 14).
- Anjum, N., Harding, J., Young, R., Case, K., Usman, Z., & Changoora, T. (2013). Verification of knowledge shared across design and manufacture using a foundation ontology. *International Journal of Production Research*, 51(22), 6534–6552.
- Azme, N. N., Kassim, N. A., Abdullah, C. Z. H., & Campus, P. P. (2017). Dimensions of Knowledge Management Maturity: Top Management Support and Leadership, People, and Information Technology. *Dimensions*, 7(2), 1-7.
- Aamodt, A. and Plaza, E. (1994) ‘Case-based reasoning: Foundational issues, methodological variations, and system approaches’, *AI Communications*, 7(1), 39–59.
- Agarwal, B., Poria, S., Mittal, N., Gelbukh, A., & Hussain, A. (2015). Concept-level sentiment analysis with dependency-Based semantic parsing: a novel approach. *Cognitive Computation*, 7(4), 487–499.
- Alarcón, R. H., Chueco, J. R., García, J. P., & Idoipe, A. V. (2010). Fixture knowledge model development and implementation based on a functional design approach. *Robotics and Computer-Integrated Manufacturing*, 26(1), 56-66.
- Aamodt, A. and Plaza, E. (1994) ‘Case-based reasoning: Foundational issues, methodological variations, and system approaches’, *AI Communications*, vol. 7, No. 1, pp. 39–59.
- Arias, M., Fernández, J. D., Martínez-Prieto, M. A., & de la Fuente, P. (2011). An empirical study of real-world SPARQL queries. *arXiv preprint arXiv:1103.5043*.

### B

- Benslimane, S. M., Bensaber, D. A., & Malki, M. (2005, January). Towards a multi-representation ontology-based information systems mediation. In *The 3rd ACS/IEEE International Conference on Computer Systems and Applications*, 2005. (44). IEEE.
- Batet, M., Sánchez, D., & Valls, A. (2011). An ontology-based measure to compute semantic similarity in biomedicine. *Journal of biomedical informatics*, 44(1), 118-125.

- Brockmans, S., Colomb, R. M., Haase, P., Kendall, E. F., Wallace, E. K., Welty, C., & Xie, G. T. (2006, November). A model driven approach for building OWL DL and OWL full ontologies. In *International Semantic Web Conference* (pp.187-200). Springer, Berlin, Heidelberg.
- Breuker, J. (2013). A cognitive science perspective on knowledge acquisition. *International Journal of Human-Computer Studies*, 71(2), 177-183.
- Bimba, A. T., Idris, N., Al-Hunaiyyan, A., Mahmud, R. B., Abdelaziz, A., Khan, S., & Chang, V. (2016). Towards knowledge modeling and manipulation technologies: A survey. *International Journal of Information Management*, 36(6), 857-871.
- Boyd, D. M., & Ellison, N. B. (2007). Social network sites: Definition, history, and scholarship. *Journal of computer-mediated Communication*, 13(1), 210-230.
- Bukowitz, W. R., & Williams, R. L. (1999). *The knowledge management field book*. Upper Saddle River, N.J: Financial Times (FT) Press, Prentice Hall.
- Borgo, S., & Leitão, P. (2004, October). The role of foundational ontologies in manufacturing domain applications. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"* (pp. 670-688). Springer, Berlin, Heidelberg.
- Ben Said, A., Shahzad, M. K., Zamai, E., Hubac, S., & Tollenaere, M. (2016). Experts' knowledge renewal and maintenance actions effectiveness in high-mix low-volume industries, using Bayesian approach. *Cognition, Technology & Work*, 18(1), 193-213.
- Bechhofer, S., Van Harmelen, F., Hendler, J., Horrocks, I., McGuinness, D. L., Patel-Schneider, P. F., & Stein, L. A. (2004). OWL web ontology language reference. W3C recommendation, 10(02).
- Brichni, M., Mandran, N., Gzara, L., Dupuy-Chessa, S., & Rozier, D. (2014). Wiki for knowledge sharing, a user-centred evaluation approach: a case study at STMicroelectronics. *Journal of Knowledge Management*.
- Brickley, D. (1998). *Resource Description Framework (RDF) Schema Specification*.

## C

- Castilho, L. V., Lopes, H. S., & Tacla, C. A. (2008, September). Modeling and building an ontology for neuropsychiatric physiotherapy domain. In *2008 Eighth International Conference on Hybrid Intelligent Systems* (pp. 210-215). IEEE.
- Chandrasekaran, B., Josephson, J. R., & Benjamins, V. R. (1999). What are ontologies, and why do we need them?. *IEEE Intelligent systems*, (1), 20-26.
- Choo, A. S., Nag, R., & Xia, Y. (2015). The role of executive problem solving in knowledge accumulation and manufacturing improvements. *Journal of Operations Management*, 36, 63-74.
- Chan, C. W. (2004). From knowledge modeling to ontology construction. *International Journal of Software Engineering and Knowledge Engineering*, 14(06), 603-624.

- Chang, X., Sahin, A., & Terpenney, J. (2008). An ontology-based support for product conceptual design. *Robotics and computer-integrated manufacturing*, 24(6), 755-762.
- Chang, X., Rai, R., & Terpenney, J. (2010). Development and utilization of ontologies in design for manufacturing. *Journal of Mechanical Design*, 132(2), 021009.
- Chungoora, N., & Young, R. I. M. (2011). The configuration of design and manufacture knowledge models from a heavyweight ontological foundation. *International Journal of Production Research*, 49(15), 4701-4725.
- Chungoora, N., & Young, R. I. (2011, September). An Ontological Approach to Consolidate Standards-Based Concepts in Production Engineering. In *2011 IEEE Fifth International Conference on Semantic Computing* (pp. 514-521). IEEE.
- Cairó, O., & Guardati, S. (2012). The KAMET II methodology: Knowledge acquisition, knowledge modeling and knowledge generation. *Expert Systems with Applications*, 39(9), 8108-8114.
- Compton P (2013). Situated cognition and knowledge acquisition research. *International Journal of Human-Computer Studies* . 71(2), 184–90.
- Cooke, N. J. (1994). Varieties of knowledge elicitation techniques. *International Journal of Human-Computer Studies*, 41(6), 801-849.
- Campuzano, F., Garcia-valverde, T., Serrano, E. and Botia, J.A. (2014). Generation of human computational models with knowledge engineering. *Engineering Applications of Artificial Intelligence*, 35, 259-276.
- Chen, Y. J. (2010). Development of a method for ontology-based empirical knowledge representation and reasoning. *Decision Support Systems*, 50(1), 1-20.
- Chen, W. C., Tseng, S. S., & Wang, C. Y. (2005). A novel manufacturing defect detection method using association rule mining techniques. *Expert systems with applications*, 29(4), 807-815.

## D

- Davenport, T. H., & Prusak, L. (1998). *Working knowledge: How organizations manage what they know*. Harvard Business Press.
- Dartigues, Christel, Parisa Ghodous, Michael Gruninger, Denis Pallez and Ram Sriram. 2007. "CAD/CAPP Integration using Feature Ontology " *Concurrent Engineering*:237-249.
- do Rosário, C. R., Kipper, L. M., Frozza, R., & Mariani, B. B. (2015). Modeling of tacit knowledge in industry: Simulations on the variables of industrial processes. *Expert Systems with Applications*, 42(3), 1613-1625.
- Dignum, F., & Vab de Riet, R. (1991). Knowledge base modelling based on linguistics and founded in logic. *Data & Knowledge Engineering*, 7, 1–34.

## E

- El-Diraby, T. E., & Osman, H. (2011). A domain ontology for construction concepts in urban infrastructure products. *Automation in Construction*, 20(8), 1120-1132.



- El-Diraby, T. E., & Osman, H. (2011). A domain ontology for construction concepts in urban infrastructure products. *Automation in Construction*, 20(8), 1120-1132.
- Eid, M. I., & Al-Jabri, I. M. (2016). Social networking, knowledge sharing, and student learning: The case of university students. *Computers & Education*, 99, 14-27.
- Ermine, J.L. (1996) *Les systèmes de connaissances*, éditions Hermès, France.

## F

- Fensel, D., Motta, E., Decker, S., & Zdráhal, Z. (1997, October). Using ontologies for defining tasks, problem-solving methods and their mappings. In *International Conference on Knowledge Engineering and Knowledge Management* (pp. 113-128). Springer, Berlin, Heidelberg.
- Fenves, S. J., Foufou, S., Bock, C., & Sriram, R. D. (2008). CPM2: a core model for product data. *Journal of computing and information science in engineering*, 8(1), 014501.
- Fellbaum, C., Hahn, U., & Smith, B. (2006). Towards new information resources for public health—from Word Net to Medical Word Net. *Journal of Biomedical Informatics*, 39(3), 321–332.

## G

- Guarino, N. (1997). Understanding, building and using ontologies. *International Journal of Human-Computer Studies*, 46(2-3), 293-310.
- Gruber, T. R. (1995). Toward principles for the design of ontologies used for knowledge sharing. *International journal of human-computer studies*, 43(5-6), 907-928.
- Gruber, T. R. (1993). A translation approach to portable ontology specifications. *Knowledge acquisition*, 5(2), 199-220.
- Gruber, T. R. (1995). Toward principles for the design of ontologies used for knowledge sharing. *International journal of human-computer studies*, 43(5-6), 907-928.
- Gruber, T. R. (1993). A translation approach to portable ontology specifications. *Knowledge acquisition*, 5(2), 199-220.
- Gangemi, A., & Borgo, S. (2004, October). Core ontologies in ontology engineering 2004.(Un) Successful cases and best practices for ontology engineering: reusing well-founded ontologies for domain content specification. In *Proceedings of the EKAW\* 04 workshop on core ontologies in ontology engineering, Northamptonshire (UK)* (Vol. 118).
- Gómez-Pérez, A. (2004). Ontology evaluation. In *Handbook on ontologies* (pp. 251-273). Springer, Berlin, Heidelberg.
- Gomez-Perez, A., Fernández-López, M., & Corcho, O. (2006). *Ontological Engineering: with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web*. Springer Science & Business Media.

- Grüniger, M., & Fox, M. S. (1995). Methodology for the design and evaluation of ontologies. in Workshop on basic ontological issues in knowledge sharing (IJCAI-95),Montreal, Canada.
- Gašević, D., Djurić, D., & Devedžić, V. (2007). MDA-based automatic OWL ontology development. *International Journal on Software Tools for Technology Transfer*, 9(2), 103.
- Gierej, S. (2017). The framework of business model in the context of Industrial Internet of Things. *Procedia Engineering*, 182, 206-212.
- Gonnet, S., Vegetti, M., Leone, H., & Henning, G. (2007). SCOntology: a formal approach toward a unified and integrated view of the supply chain. In *Adaptive Technologies and Business Integration: Social, Managerial and Organizational Dimensions* (pp. 137-158). IGI Global.
- Gunendran, A. G., & Young, R. I. M. (2010). Methods for the capture of manufacture best practice in product lifecycle management. *International Journal of Production Research*, 48(20), 5885-5904.
- Gruber, T. R. (2013). Nature, nurture, and knowledge acquisition. *International journal of human-computer studies*, 71(2), 191-194.
- Grundstein M. (1996) "CORPUS," An Approach to Capitalizing Company Knowledge. In: Ein-Dor P. (eds) *Artificial Intelligence in Economics and Management*. Springer, Boston, MA.
- Grundstein M., *Management des connaissances de l'entreprise: problématique, axe de progrès, orientations*, 2000.
- Grundstein, Michel (May 22–24, 1995)..La Capitalisation des Connaissances de l'Entreprise, Système de production des connaissances. Actes du Colloque "L'Entreprise Apprenante et les sciences de la complexité", Université de Provence, Aix-en-Provence, France.
- Grundstein M. (2000). *Management des connaissances de l'entreprise: problématique, axe de progrès, orientations*.
- Günel, A., Meshram, A., Bley, T., Schuetze, A., & Klusch, M. (2013). Statistical and semantic multisensor data evaluation for fluid condition monitoring in wind turbines. In *Proc. 16th Intl. Conf. on Sensors and Measurement Technology, Germany*.
- Girard, J., & Girard, J. (2015). Defining knowledge management: Toward an applied compendium. *Online Journal of Applied Knowledge Management*, 3(1), 1-20.
- Gaines, B. R. (2013). Knowledge acquisition: Past, present and future. *International Journal of Human-Computer Studies*, 71(2), 135-156.
- George S. and Karapistolis, D.(2014) 'Tourist Destination Marketing Supported by Electronic Capitalization of Knowledge', *Procedia - Social and Behavioral Sciences* , 148, pp. 110 – 118 .

## H

- Hamalainen, M., & Karjalainen, J. (2017). Social manufacturing: When the maker movement meets interfirm production networks. *Business Horizons*, 60(6), 795-805.
- Huang, D., & Gao, J. (2009, May). Technology of building domain ontology based on SECI. In 2009 WRI World Congress on Software Engineering (Vol. 4, pp. 519-523). IEEE.
- Haarslev, V., Hidde, K., Möller, R., & Wessel, M. (2012). The RacerPro knowledge representation and reasoning system. *Semantic Web*, 3(3), 267-277.
- Horrocks, I., Patel-Schneider, P. F., Boley, H., Tabet, S., Grosz, B., & Dean, M. (2004). SWRL: A semantic web rule language combining OWL and RuleML. *W3C Member submission*, 21(79), 1-31.
- Hsu, I. C. (2008). Knowledge sharing practices as a facilitating factor for improving organizational performance through human capital: A preliminary test. *Expert Systems with applications*, 35(3), 1316-1326.
- Hoffman, R. R., Shadbolt, N. R., Burton, A. M., & Klein, G. (1995). Eliciting knowledge from experts: A methodological analysis. *Organizational behavior and human decision processes*, 62(2), 129-158.
- Hunter Alarcon, R., Rios Chueco, J., Perez Garcia, J.M. and Vizan Idoipe, A. (2010) 'Fixture knowledge model development and implementation based on a functional design approach', *Robotics and Computer-Integrated Manufacturing*, Vol. 26, No.1, pp. 56-66.

## I

- Imran, M., & Young, B. (2015). The application of common logic based formal ontologies to assembly knowledge sharing. *Journal of intelligent manufacturing*, 26(1), 139-158.

## K

- Krima, S., Barbau, R., Fiorentini, X., Sudarsan, R., & Sriram, R. D. (2009). Ontostep: OWL-DL ontology for step. National Institute of Standards and Technology, NISTIR, 7561.
- Kim, D., Han, S. C., Lin, Y., Kang, B. H., & Lee, S. (2018). RDR-based knowledge based system to the failure detection in industrial cyber physical systems. *Knowledge-Based Systems*, 150, 1-13.
- Kjellberg, T., von Euler-Chelpin, A., Hedlind, M., Lundgren, M., Sivard, G., & Chen, D. (2009). The machine tool model—A core part of the digital factory. *CIRP annals*, 58(1), 425-428.
- Karray, M. H., Chebel-Morello, B., & Zerhouni, N. (2012). A formal ontology for industrial maintenance. *Applied Ontology*, 7(3), 269-310.
- Kosanke, K., Vernadant, F. and Zelm, M. (1999) 'CIMOSA: enterprise engineering and integration', *Computers Industry*, 40(2-3), 83-97.

## L

- Labidi, S., & Sérgio, N. (2000, October). Student modeling and semi-automatic domain ontology construction for SHIECC [automatic read automatic]. In 30th Annual Frontiers in Education Conference. Building on A Century of Progress in Engineering Education. Conference Proceedings (IEEE Cat. No. 00CH37135) (Vol. 1, pp. F1B-14). IEEE.
- Liu, G., Wang, Y., & Wu, C. (2010, June). Research and application of geological hazard domain ontology. In 2010 18th International Conference on Geoinformatics (pp. 1-6). IEEE.
- Lawley, M. J., & Bousquet, C. (2010, December). Fast classification in Protégé: Snorocket as an OWL 2 EL reasoner. In *Proc. 6th Australasian Ontology Workshop (IAOA'10). Conferences in Research and Practice in Information Technology* (Vol. 122, pp. 45-49).
- López-Cuadrado, J. L., Colomo-Palacios, R., González-Carrasco, I., García-Crespo, Á., & Ruiz-Mezcua, B. (2012). SABUMO: Towards a collaborative and semantic framework for knowledge sharing. *Expert Systems with Applications*, 39(10), 8671-8680.
- Liu, W., Liu, Z. T., & Shao, K. (2003, November). UML-based domain ontology modeling for multi-agent system. In Proceedings of the 2003 International Conference on Machine Learning and Cybernetics (IEEE Cat. No. 03EX693) (Vol. 1, pp. 407-412). IEEE.
- Lemaignan, S., Siadat, A., Dantan, J. Y., & Semenenko, A. (2006, June). MASON: A proposal for an ontology of manufacturing domain. In *IEEE Workshop on Distributed Intelligent Systems: Collective Intelligence and Its Applications (DIS'06)* (pp. 195-200). IEEE.
- Lin, H. K., & Harding, J. A. (2007). A manufacturing system engineering ontology model on the semantic web for inter-enterprise collaboration. *Computers in Industry*, 58(5), 428-437.
- Lee, J. H., & Suh, H. W. (2008). Ontology-based multi-layered knowledge framework for product lifecycle management. *Concurrent Engineering*, 16(4), 301-311.
- Lin, L. F., Zhang, W. Y., Lou, Y. C., Chu, C. Y., & Cai, M. (2011). Developing manufacturing ontologies for knowledge reuse in distributed manufacturing environment. *International Journal of Production Research*, 49(2), 343-359.
- Li, X., Wu, Z., Goh, M., & Qiu, S. (2018). Ontological knowledge integration and sharing for collaborative product development. *International Journal of Computer Integrated Manufacturing*, 31(3), 275-288.
- Liu, H. C., Liu, L., Lin, Q. L., & Liu, N. (2013). Knowledge acquisition and representation using fuzzy evidential reasoning and dynamic adaptive fuzzy petri nets. *IEEE Transactions on Cybernetics*, 43(3), 1059-1072.
- Liou, Y. I. (1990, September). Knowledge acquisition: issues, techniques, and methodology. In Proceedings of the 1990 ACM SIGBDP conference on Trends and directions in expert systems (pp. 212-236).

- Lakhfif, A., & Laskri, M. T. (2015). A frame-based approach for capturing semantics from Arabic text for text-to-sign language MT. *International Journal of Speech Technology*.
- Louis-Sidney, L., Cheutet, V., Lamouri, S., Puron, O., & Mezza, A. (2012). A conceptual model for the implementation of an Inter-Knowledge Objects Exchange System (IKOES) in automotive industry. *Engineering Applications of Artificial Intelligence*, 25(5), 1090-1101.

## M

- Makki, J., Alquier, A. M., & Prince, V. (2008, October). Semi automatic ontology instantiation in the domain of risk management. In *International Conference on Intelligent Information Processing* (pp. 254-265). Springer, Boston, MA.
- Michailova, S., & Gupta, A. (2005). Knowledge sharing in consulting companies: Opportunities and limitations of knowledge codification. *Journal of Information & Knowledge Management*, 4(03), 201-212.
- Ma, W. W., & Chan, A. (2014). Knowledge sharing and social media: Altruism, perceived online attachment motivation, and perceived online relationship commitment. *Computers in Human Behavior*, 39, 51-58.
- Mourtzis, D., Doukas, M., & Milas, N. (2016). A knowledge-based social networking app for collaborative problem-solving in manufacturing. *Manufacturing Letters*, 10, 1-5.
- Mesmer, L., & Olewnik, A. (2015, August). Development of a part-focused manufacturing process ontology: Exploring use and applications. In *ASME 2015 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. American Society of Mechanical Engineers Digital Collection.
- Mazzola, L., Kapahnke, P., Vujic, M., & Klusch, M. (2016, November). CDM-Core: A Manufacturing Domain Ontology in OWL2 for Production and Maintenance. In *KEOD* (pp. 136-143).
- Musen, M. A. (2013). The knowledge acquisition workshops: a remarkable convergence of ideas. *International Journal of Human-Computer Studies*, 71(2), 195-199.
- Motta E (2013). 25 years of knowledge acquisition. *International Journal of Human-Computer Studies*, 71(2), 131-134.
- Maksimovic, M., Al-Ashaab, A., Shehab, E., Flores, M., Ewers, P., Haque, B., & Sulowski, R. (2014). Industrial challenges in managing product development knowledge. *Knowledge-Based Systems*, 71, 101-113.
- Mourtzis, D., & Doukas, M. (2014). Knowledge capturing and reuse to support manufacturing of customised products: A case study from the mould making industry. *Procedia CIRP*, 21, 123-128.
- Mourtzis, D., Doukas, M., & Milas, N. (2016). A knowledge-based social networking app for collaborative problem-solving in manufacturing. *Manufacturing Letters*, 10, 1-5.

Menaouer, B., Khalissa, S., Abdelbaki, B., & Abdelhamid, T. (2015). An approach of support innovation guided by knowledge capitalization: Application on FERTIAL. *Procedia-Social and Behavioral Sciences*, 181, 197-206.

## N

Noy, N. F., & McGuinness, D. L. (2001). *Ontology development 101: A guide to creating your first ontology*. Stanford Medical Informatics technical report no.SMI-2001-0880. Available from: [www.smi.stanford.edu/projects/protege/publications/ontology\\_development/ontology101.pdf](http://www.smi.stanford.edu/projects/protege/publications/ontology_development/ontology101.pdf) [Accessed 3 March 2008].

Naif Marouf, L. (2007). Social networks and knowledge sharing in organizations: a case study. *Journal of knowledge management*, 11(6), 110-125.

Nonaka, I. and Takeuchi, H. (1995) *the Knowledge-Creating Company: How Japanese Companies Create the Dynamics of Innovation*, Oxford University Press, USA.

Nonaka, I., Toyama, R., & Konno, N. (2000). SECI, Ba and leadership: a unified model of dynamic knowledge creation. *Long range planning*, 33(1), 5-34.

Negnevitsky, M. (2005). *Artificial intelligence: a guide to intelligent systems*. Pearson Education.

## O

O'Dell, C., & Grayson, C. J. (1998). *If only we knew what we know: the transfer of internal knowledge and best practice*. New York: Free Press.

O'Connor, M. J., & Das, A. K. (2009, October). SQWRL: a query language for OWL. In *OWLED* (Vol. 529, No. 2009).

## P

Patil, L., Dutta, D., Patil, L., & Sriram, R. D. (2005). *Ontology formalization of product semantics for product lifecycle management*. US Department of Commerce, National Institute of Standards and Technology.

Panahi, S., Watson, J., & Partridge, H. (2012). Social media and tacit knowledge sharing: Developing a conceptual model. *World academy of science, engineering and technology*, (64), 1095-1102.

Poveda-Villalón, M., Gómez-Pérez, A., & Suárez-Figueroa, M. C. (2014). Oops!(ontology pitfall scanner!): An on-line tool for ontology evaluation. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 10(2), 7-34.

Peretz, R. A., & Luria, G. (2017). Drivers' social-work relationships as antecedents of unsafe driving: A social network perspective. *Accident Analysis & Prevention*, 106, 348-357.

Panetto, H., Dassisti, M., & Tursi, A. (2012). ONTO-PDM: Product-driven ONTOlogy for Product Data Management interoperability within manufacturing process environment. *Advanced Engineering Informatics*, 26(2), 334-348.



- Petrash, G. (1996) "Managing knowledge assets for value," Knowledge-Based Leadership Conference, pp. 18–38.
- Prat, N., Jacky A. and I Comyn-Wattiau, I. (2012) 'An MDA approach to knowledge engineering', *Expert Systems with Applications*, Vol. 39, No. 12, pp. 10420–10437.

## R

- Rector, A. L., Nowlan, W. A., Kay, S., Goble, C. A., & Howkins, T. J. (1993). A framework for modelling the electronic medical record. *Methods of information in medicine*, 32(02), 109-119.
- Ramos, L., Gil, R., Anastasiou, D., & Martin-Bautista, M. J. (2014). Towards a Machine of a Process (MOP) ontology to facilitate e-commerce of industrial machinery. *Computers in Industry*, 65(1), 108-115.
- Ruiz, P. P., Foguem, B. K., & Grabot, B. (2014). Generating knowledge in maintenance from Experience Feedback. *Knowledge-Based Systems*, 68, 4-20.
- Rasovska, I. (2008). Contribution à une méthodologie de capitalisation des connaissances basée sur le raisonnement à partir de cas: Application au diagnostic dans une plateforme d'e-maintenance (Doctoral dissertation).

## S

- Su, X., Matskin, M., & Rao, J. (2003, October). Implementing explanation ontology for agent system. In *Proceedings IEEE/WIC International Conference on Web Intelligence (WI 2003)* (pp. 330-336). IEEE.
- Savonnet, M., Leclercq, E., & Naubourg, P. (2015). eClims: an extensible and dynamic integration framework for biomedical information systems. *IEEE journal of biomedical and health informatics*, 20(6), 1640-1649.
- Schreiber, G., Wielinga, B., Akkermans, H., Van de Velde, W., & Anjewierden, A. (1994, September). CML: The CommonKADS conceptual modelling language. In *International Conference on Knowledge Engineering and Knowledge Management* (pp. 1-25). Springer, Berlin, Heidelberg.
- Sirin, E., Parsia, B., Grau, B. C., Kalyanpur, A., & Katz, Y. (2007). Pellet: A practical owl-dl reasoner. *Web Semantics: science, services and agents on the World Wide Web*, 5(2), 51-53.
- Studer, R., Benjamins, V. R., & Fensel, D. (1998). Knowledge engineering: principles and methods. *Data & knowledge engineering*, 25(1-2), 161-197.
- Schreiber, A. T., Schreiber, G., Akkermans, H., Anjewierden, A., Shadbolt, N., de Hoog, R., & Wielinga, B. (2000). *Knowledge engineering and management: the CommonKADS methodology*. MIT press.
- Semenova, A. V., & Kureychik, V. M. (2015, October). Domain ontology development for linguistic purposes. In *2015 9th International Conference on Application of Information and Communication Technologies (AICT)* (pp. 83-87). IEEE.

- Semere, D. T., Dilshad, S., & Lindberg, B. (2007). Machining ontology and knowledge modelling. In *Swedish Production Symposium*.
- Sakka, O., Millet, P. A., & Botta-Genoulaz, V. (2011). An ontological approach for strategic alignment: a supply chain operations reference case study. *International journal of computer integrated manufacturing*, 24(11), 1022-1037.
- Saeidlou, S., Saadat, M., Amini Sharifi, E., & Jules, G. D. (2017). An ontology-based intelligent data query system in manufacturing networks. *Production & Manufacturing Research*, 5(1), 250-267.
- Serrat, O. (2009). Glossary of Knowledge Management Knowledge Solutions: Asian Development Bank.
- Senker, J. (1995). Tacit knowledge and models of innovation. *Industrial and corporate change*, 4(2), 425-447.
- Szulanski, G. (2003) *Sticky Knowledge: Barriers to Knowing in the Firm*, Thousand Oaks, London.
- Schreiber, G. (2013). Knowledge acquisition and the web. *International Journal of Human-Computer Studies*, 71(2), 206-210.
- Shadbolt, N., O'hara, K., & Crow, L. (1999). The experimental evaluation of knowledge acquisition techniques and methods: history, problems and new directions. *International journal of human-computer studies*, 51(4), 729-755.
- Shadbolt, N. (2013). Knowledge acquisition and the rise of social machines. *International Journal of Human-Computer Studies*, 71(2), 200-205.
- Schreiber, G. (2000) *Knowledge engineering and management: the CommonKADS methodology*, MIT press.
- Schreiber, G. (2000) *Knowledge engineering and management: the CommonKADS methodology*, MIT press.
- Schreiber, G., Akkermans, H., (2002). *Knowledge engineering and management: the CommonKADS methodology*, London: The MIT Press.
- Senker, J. (1995) . Tacit knowledge and models of innovation. *Industrial and Corporate Change*, 4(2). 425-47.
- Studer, R., Benjamins, V. R., & Fensel, D. (1998). Knowledge engineering: principles and methods. *Data & Knowledge Engineering*, 25(1), 161–197.
- Selva, D., & Crawley, E. F. (2012). A rule-based decision support tool for architecting Earth observing missions. In Paper presented at the aerospace conference, 2012 IEEE.
- Selva, D., Cameron, B., & Crawley, E. F. (2014). A rule-based method for scalable and traceable evaluation of system architectures. *Research in Engineering Design*, 25(4), 325–349.
- Said Saleh, M., Ismail, O., Kamel, A., & Hassan, H. (2018). From CommonKADS to SOA Environment: An Adaptation Model. *Arabian Journal for Science & Engineering (Springer Science & Business Media BV)*, 43(12).



## T

- Tartir, S., Arpinar, I. B., & Sheth, A. P. (2010). Ontological evaluation and validation. In *Theory and applications of ontology: Computer applications* (pp. 115-130). Springer, Dordrecht.
- Titah, M., Mouss, M. D. M., & Aitouche, S. (2014). An implementation tool for the expertise model using CommonKADS methodology. In *The Sixth International Conference on Information, Process, and Knowledge Management, eKNOW*.
- Titah, M., Aitouche, S., Mouss, M. D., & Soussa, A. (2017). Externalising and reusing of tacit knowledge in manufacturing task. *International Journal of Knowledge Management Studies*, 8(3-4), 351-374.
- Tijerino, Y. A., & Mizoguchi, R. (1993, September). MULTIS II: Enabling end-users to design problem-solving engines via two-level task ontologies. In *International Conference on Knowledge Engineering and Knowledge Management* (pp. 340-359). Springer, Berlin, Heidelberg.
- Tursi, A., Panetto, H., Morel, G., & Dassisti, M. (2009). Ontological approach for products-centric information system interoperability in networked manufacturing enterprises. *Annual Reviews in Control*, 33(2), 238-245.
- Thramboulidis, K., & Christoulakis, F. (2016). UML4IoT—A UML-based approach to exploit IoT in cyber-physical manufacturing systems. *Computers in Industry*, 82, 259-272.

## U

- Usman, Z., Young, R. I. M., Chungoora, N., Palmer, C., Case, K., & Harding, J. A. (2013). Towards a formal manufacturing reference ontology. *International Journal of Production Research*, 51(22), 6553-6572.
- Uriarte, F. A. (2008). *Introduction to Knowledge Management: A brief introduction to the basic elements of knowledge management for non-practitioners interested in understanding the subject*. Jakarta, Indonesia: ASEAN Foundation.

## V

- Van Heijst, G., Schreiber, A. T., & Wielinga, B. J. (1997). Using explicit ontologies in KBS development. *International journal of human-computer studies*, 46(2-3), 183-292.
- Vrandečić, D. (2009). Ontology evaluation. In *Handbook on ontologies* (pp. 293-313). Springer, Berlin, Heidelberg.
- Vogel, C. (1990). *KOD : la mise en œuvre*, Editions Masson, Paris.

## W

- Wielinga, B. J., Schreiber, A. T., & Breuker, J. A. (1992). KADS: A modelling approach to knowledge engineering. *Knowledge acquisition*, 4(1), 5-53.
- Weithöner, T., Liebig, T., Luther, M., Böhm, S., Von Henke, F., & Noppens, O. (2007, June). Real-world reasoning with OWL. In *European Semantic Web Conference* (pp. 296-310). Springer, Berlin, Heidelberg.

- Wickens, C. D., Hollands, J. G., Banbury, S., & Parasuraman, R. (2015). *Engineering psychology and human performance*. Psychology Press.
- Wielinga, B. J., Schreiber, A. T., & Breuker, J. A. (1992). KADS: a modelling approach to knowledge engineering. *Knowledge Acquisition*, 4(1), 5–53.p
- Wielinga B.J, Schreiber A.T, & Breuker J.A (1992). KADS: a modelling approach to knowledge engineering. *Knowledge Acquisition* 4, 5-53.
- Wang, Y. (2008). On concept algebra: a denotational mathematical structure for knowledge and software modeling. *International Journal of Cognitive Informatics and Natural Intelligence (IJCINI)*, 2(2), 1–19.

X

- Xing, X., Li, R., & Liu, K. (2009, October). Building ontology base on thesaurus. In *2009 2nd International Conference on Biomedical Engineering and Informatics* (pp. 1-4). IEEE.
- Xu, X. (2012). From cloud computing to cloud manufacturing. *Robotics and computer-integrated manufacturing*, 28(1), 75-86.

Y

- Yin, J. W., Zhang, W. Y., & Cai, M. (2010). Weaving an agent-based Semantic Grid for distributed collaborative manufacturing. *International Journal of Production Research*, 48(7), 2109-2126.

Z

- Zhou, L. (2007). Ontology learning: state of the art and open issues. *Information Technology and Management*, 8(3), 241-252.
- Zhou, J., & Dieng-Kuntz, R. (2004, June). Manufacturing ontology analysis and design: towards excellent manufacturing. In *2nd IEEE International Conference on Industrial Informatics, 2004. INDIN'04. 2004* (pp. 39-45). IEEE.
- Zhang, W. Y., Cai, M., Qiu, J., & Yin, J. W. (2009). Managing distributed manufacturing knowledge through multi-perspective modelling for semantic web applications. *International Journal of Production Research*, 47(23), 6525-6542.
- Zajaczkowski, J., & Verma, B. (2012). Selection and impact of different topologies in multi-layered hierarchical fuzzy systems. *Applied Intelligence*, 36(3), 564–584.
- Zhou, Y., Li, Y., & Wang, W. (2011). A feature-based fixture design methodology for the manufacturing of aircraft structural parts. *Robotics and Computer-Integrated Manufacturing*, 27(6), 986-993.
- Zhang, Q. A., & Mahfouf, M. (2011). A hierarchical Mamdani-type fuzzy modelling approach with new training data selection and multi-objective optimisation mechanisms: a special application for the prediction of mechanical properties of alloy steels. *Applied Soft Computing*, 11(2), 2419–2443.

## **Sites Web**

- <http://www.cs.man.ac.uk/~sattler/reasoners.html>, Consulté le 12/06/2016
- <https://webprotege.stanford.edu/>, Consulté le 20/01/2017
- <http://herzberg.ca.sandia.gov/>, Consulté le 25/02/2017
- <http://research.ict.csiro.au/software/snorocket> Consulté le 03/10/2016
- [www.franz.com/agraph/cresources/white\\_papers](http://www.franz.com/agraph/cresources/white_papers) Consulté le 10/01/2017
- <http://www.cs.man.ac.uk/~sattler/reasoners.html> Consulté le 03/02/2017
- <http://www.ontologyworks.com/products/iode> Consulté le 04/06/2019
- <http://www.ksl.stanford.edu/software/ontolingua/> Consulté le 11/10/2019
- <https://www.w3.org/Submission/SWRL/> Consulté le 11/05/2016
- <http://www.jessrules.com/> Consulté le 10/11/2017
- <http://oops.linkeddata.es/> Consulté le 22/04/2020
- <https://www.w3.org/2001/sw/wiki/Pellet> Consulté le 10/04/2018
- <https://ontometrics.informatik.uni-rostock.de> Consulté le 15/03/2020
- <https://www.w3.org/2009/01/sparql-charter> Consulté le 12/09/2018
- <https://jsp.java.net/>, Consulté le 24/03/2020
- <http://jena.apache.org/index.html>, Consulté le : 04/10/2018
- <http://tomcat.apache.org/> Consulté le 10/02/2020

## Annexes

Annexe A : CommonKADS.....	181
Annexe B : MATO.....	188
Annexe C : Requêtes SQWRL de l'ontologie MATO dans Protégé.....	196
Annexe D : Requêtes SPARQL.....	197
Annexe E : PPASD.....	199

## Annexe A : CommonKADS

**Tableau A1 : MO-II: Les aspects organisationnels ayant un impact ou pouvant être affectés par la solution de connaissances**

Modèle d'Organisation	Variant Aspects Worksheet OM-2
Structure	Un organigramme de l'organisation présenté dans la figureA.1
Processus	Une brève description du processus d'affaires. Il est divisé en tâches détaillées dans MO-3
Les acteurs	Les ingénieurs de production, les cadres d'exploitation (les experts), les Techniciens Principaux Conduite Centrale (TPCC), Techniciens Conduites Auxiliaires (TCA), techniciens Principaux chimistes (TPC), chauffeur VL (Voir Figure A.2).
Les ressources	<p>Pour commander et contrôler le fonctionnement des équipements relatifs aux trois groupes, la centrale dispose d'une salle de contrôle située dans le bâtiment de commande.</p> <ul style="list-style-type: none"> <li>➤ Un Système de surveillance en Ligne : pour la surveillance des installations de production des trois groupes, et contrôle l'état de rendement du système, et affiche les anomalies par des alarmes dans le panneau de signalisation.</li> <li>➤ Un logiciel GEIDE : qui permet la gestion des documentations.</li> <li>➤ Logiciel SAP : utilisé actuellement pour la gestion comptable et le contrôle de gestion.</li> </ul>
La connaissance	La connaissance est une ressource spéciale exploitée dans un processus d'affaires (décrit au tableau MO-4)
La culture et le pouvoir	La centrale de Jijel contient des experts compétents et des cadres capables d'assurer la continuité de production et la disponibilité des moyens de productions grâce à leurs expériences dans le domaine.

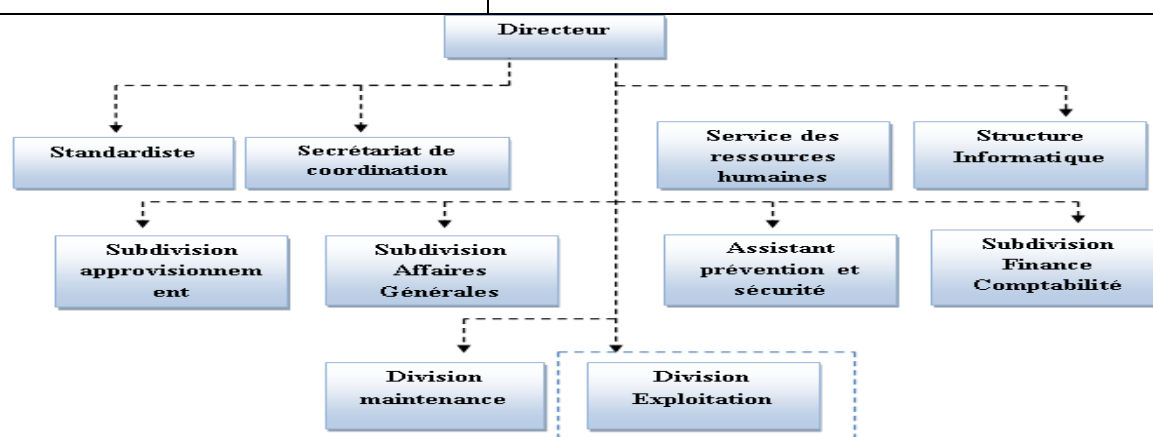


Figure A.1. Division exploitation

**Tableau A2 : Modèle de l'organisation-3 (MO-3)**

Modèle d'organisation		Processus Breakdown Worksheet OM-3				
No (identificateur de la tâche)	Tâche (nom de la tâche)	Effectuée par (agent)	Où? (Endroit dans la structure)	Unité de connaissance (voir MO-4)	Connaissances intensive? (Oui/non)	Signification (Sur une échelle de 1 à 5, en fonction de la fréquence, du coût, etc.
1	Observation du système de surveillance	Le Technicien Principal Conduite Centrale TPCC	Service quart de production (salle de commande)	A partir des panneaux de signalisations.	non	3
2	Vérification des paramètres de fonctionnement	Le TPCC et TCA et TPC	salle de commande	A partir du tableau de bord (affichages des paramètres : température, pression, charge..)	non	5
3	Transformer l'information de dysfonctionnement à l'ingénieur de production et cadres d'exploitation	TPCC	Salle de commande		non	2
4	Diagnostiquer les alarmes de signalisation	Les cadres d'exploitation, les ingénieurs de productions, TPCC, TCA, TPC.	Salle de commande	A partir des connaissances tacites des experts et leurs compétences.	oui	5
5	Consigne les anomalies de fonctionnement et les mesures prises pour y remédier	Ingénieur de production	Salle de commande	A partir de la consigne d'exploitation.	oui	4

**Tableau A3 : Modèle de l'organisation-4 (MO-4)**

Modèle d'organisation		Knowledge Asset Worksheet OM-4				
Nom <i>(unité de connaissance)</i> <i>(voir MO-3)</i>	Appartient à <i>(agent)</i> <i>(voir MO-3)</i>	Utilisée dans <i>(tâche)</i> <i>(voir MO-3)</i>	Bon format? <i>(Oui/non)</i>	Bon endroit? <i>(Oui/non)</i>	Bon moment? <i>(Oui/non)</i>	Bonne qualité? <i>(Oui/non)</i>
Dénomination de l'alarme	TPCC	Observation du SSE et diagnostiquer.	non	oui	oui	oui
Identifier les causes probables de dysfonctionnement	Ingénieurs de production et cadres d'exploitations.	Diagnostiquer les anomalies	non	non	oui	oui
Trouver le temps nécessaire pour l'intervention avant le déclenchement du système.	Cadres d'exploitations	Diagnostiquer les anomalies	non	non	oui	oui

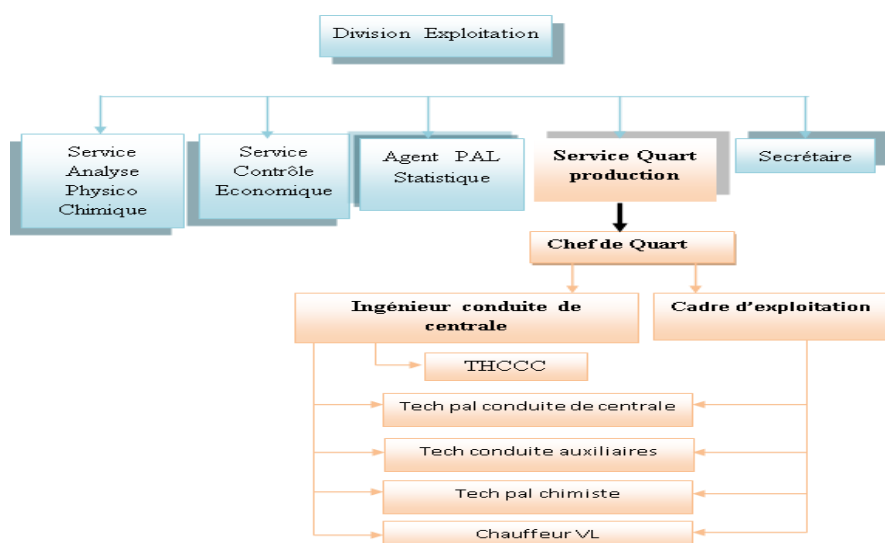


Figure A.2. Détails de division exploitation

**Tableau A4: Modèle de Tâche-1 (MT-1)**

<b>Task Model</b>	<b>Task Model Analysis Worksheet TM-1</b>	
Tâche	Voir MO-3	Diagnostiquer les alarmes de signalisation
Organisation	Voir MO-2	Division exploitation : Service quart, la salle de commande par les cadres d'exploitations
But et valeur		Aide aux personnels d'exploitation pour leur Permettre de réaliser le diagnostic d'état de leurs matériels. Il disposera pour cela D'outils performants et attrayants le déchargeant en outre du travail routinier.
Dépendance et flux	Les tâches précédentes  Les tâches suivantes	Vérification des paramètres de fonctionnement.  Consigne les anomalies de fonctionnement et les mesures prises pour y remédier
Objets traités	Objets en entrée  Objets en sortie  Objets internes	Objets d'entrée : alarme de signalisation.  Objets de sortie : identification de la cause probable par l'expert.
Temps et contrôle	Fréquence, Durée  Contrôle  Contraintes & Conditions	La tâche la plus important est identification de la cause de signalisation, le temps d'intervention avant le déclenchement ne dépasse plus de 5 minutes.
Agents	MO-2: Gens, Systèmes  Ressources;  MO-3: effectuée par	Les personnels qui sont responsables de ces tâches :  Les cadres d'exploitation (anciens exploitant) et les ingénieurs de production.
Connaissances et compétences	Voir MO-4	Compétences nécessaires pour bien effectuer la tâche. Pour les éléments de connaissances dans le tableau  MT-2
Ressources	Détails de MO 2	
Qualité et Performance	Mesures	Le but de la tâche est d'éliminer la cause du problème le plus vite possible avant le déclenchement de système.



**Tableau A5: Modèle de Tâche-2 (MT-2)**

Élément de connaissance : identification de la cause de dysfonctionnement.		
Agent : cadre d'exploitation		
Nom de tâche et identificateur : Diagnostiquer les alarmes de signalisation		
Nature de la connaissance		Goulôt d'étranglement/ à améliorer
Formelle, rigoureuse		×
Empirique, quantitative		
Heuristique, règles du pouce		
Hautement spécialisée, propre au domaine	×	
Basée sur l'expérience	×	
Basée sur les actions		
Incomplète		
Incertaine, pourrait être incorrecte		
Peut changer rapidement	×	
Difficile à vérifier		
Tacite, difficile à transférer	×	×
Forme de la connaissance		
Mémoire humaine	×	
Sur papier		
Électronique		
compétence		
autres		
Disponibilité de la connaissance		
Limites temporelles		
Limites spatiales	×	
Limites d'accès		
Limites de qualité		
Limites de forme	×	

**Tableau A6: Modèle d'Agent (AM-1)**

Modèle d'agent	Agent Worksheet AM-1
Nom	Cadre d'exploitation
Organisation	Division exploitation, service Quart, salle de commande.
Impliqué dans	<ul style="list-style-type: none"> <li>• diagnostique les anomalies de fonctionnement.</li> <li>• contrôler les équipements de production</li> <li>• superviser les opérations de démarrage et d'arrêt des installations.</li> <li>• Consigne les anomalies de fonctionnement et le remède.</li> </ul>
Communique avec	<ul style="list-style-type: none"> <li>• Le chef de quart, les ingénieurs de productions, les TPCC, TCA.</li> <li>• Le logiciel GMAO : service maintenance.</li> </ul>
Connaissance	<ul style="list-style-type: none"> <li>• Les connaissances tacites et explicites pour la surveillance de fonctionnement des installations et maintenir la production demandée sur la base des paramètres affichées (les panneaux de signalisation).</li> </ul>
Autres compétences	<ul style="list-style-type: none"> <li>• Participer lors des arrêts ou dans le cadre de la rotation de quart à des travaux d'entretien sur le matériel.</li> <li>• Etudier les rapports d'exploitation quotidiens et analyse les dysfonctionnement.</li> <li>• Proposer des consignes d'exploitation de fonctionnement.</li> </ul>
Responsabilités et contraintes	<ul style="list-style-type: none"> <li>• Gérer les équipements de mesure de rendement.</li> <li>• Responsable d'arrêt de groupe en cas d'un problème.</li> </ul>

**Tableau A7: Modèle de l'Organisation-Tâche-Agent OTA-1**

Modèles d'Organisation-Tâche-Agent	Checklist for Impact and Improvement Decision Document: Worksheet OTA-1
Impacts et Changements dans l'organisation	<p>Un nouveau système de connaissances dans l'entreprise basé sur l'expérience et compléter le système de surveillance, On appelle poste de surveillance et aide au diagnostic.</p> <p>Et une base commune entre la division exploitation et division maintenance.</p>
Impacts et changements spécifiques aux tâches/agents.	<p>Le système de connaissance offrira aux personnels de centrale thermique de Jijel, service quart de production une aide indispensable pour leur Permettre de réaliser le diagnostic d'état de leurs matériels.</p>
Actions proposées	<ul style="list-style-type: none"> <li>• Créé un service commun entre la division exploitation et la division maintenance au niveau de service Quart de production pour assurer la continuité de production d'électricité.</li> <li>• Faire des partages de connaissances entre les anciens experts et nouveaux ingénieurs et techniciens.</li> </ul>

## Annexe B : MATO

### Extrait de l'ontologie MATO instanciée

```
<?xml version="1.0"?>
<rdf:RDF
xmlns="http://www.semanticweb.org/userr/ontologies/2020/2/AlarmDiagnosis#"
xml:base="http://www.semanticweb.org/userr/ontologies/2020/2/AlarmDiagnosis"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  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#"
  xmlns:swrl="http://www.w3.org/2003/11/swrl#"
  xmlns:swrla="http://swrl.stanford.edu/ontologies/3.3/swrla.owl#"
  xmlns:swrlb="http://www.w3.org/2003/11/swrlb#"

xmlns:titah="http://www.semanticweb.org/userr/ontologies/2020/2/AlarmSystemDiagnosis#"
xmlns:ProblemData="http://www.semanticweb.org/userr/ontologies/2020/2/AlarmSystemDiagnosis#ProblemData#"
xmlns:ProblemData1="http://www.semanticweb.org/userr/ontologies/2020/2/AlarmDiagnosis#ProblemData#"
xmlns:AlarmDiagnosis="http://www.semanticweb.org/userr/ontologies/2020/2/AlarmDiagnosis#">
  <owl:Ontology
rdf:about="http://www.semanticweb.org/userr/ontologies/2020/2/AlarmDiagnosis">
    <owl:versionIRI
rdf:resource="http://www.semanticweb.org/userr/ontologies/2020/2/AlarmDiagnosis/1.1"/>
      <Description xml:lang="en">
-MATO : A Semantic Technology for improvement Manufacturing Problem-Solving methods.
-MATO : An off line knowledge sharing process.
-MATO : Facilitate reusing and knowledge Sharing in manufacturing system.
- MATO : Facilitates knowledge sharing between manufacturing tasks.
-MATO : Helps to capture and reuse knowledge during task execution
</Description>
      <Realized_By xml:lang="en">Mawloud TITAH</Realized_By>
      <Title xml:lang="en">MATO : MANufacturing Task
Ontology</Title>
      <rdfs:comment xml:lang="en"> A generic model can be applied for all manufacturing tasks, it includes manufacturing process, manufacturing tasks, manufacturing problems, manufacturing solutions, manufacturing features, manufacturing equipmentâ€™s and manufacturing agents.</rdfs:comment>
      <rdfs:label xml:lang="en">MATO</rdfs:label>
      <rdfs:seeAlso xml:lang="en">Laboratory of Automation and manufacturing, University of batna 2, Algeria.</rdfs:seeAlso>
      <owl:deprecated xml:lang="en">Mawloud TITAH</owl:deprecated>
      <owl:versionInfo xml:lang="en">Version 1.0 /2017 : Design and building MATO.
Version1.1/2018 : Added manufacturing problem and manufacturing solution concepts.
Version 1.2/ 2019 : Added new two types of manufacturing Feature (Informational and Physical feature)</owl:versionInfo>
```

```

    </owl:Ontology>
    <!--

////////////////////////////////////
////////////////////////////////////
    //
    // Annotation properties
    //

////////////////////////////////////
////////////////////////////////////
    -->
    <!--
http://swrl.stanford.edu/ontologies/3.3/swrla.owl#isRuleEnabled -->
    <owl:AnnotationProperty
rdf:about="http://swrl.stanford.edu/ontologies/3.3/swrla.owl#isRuleEn
abled"/>

    <!--
http://www.semanticweb.org/userr/ontologies/2020/2/AlarmDiagnosis#Des
cription -->
    <owl:AnnotationProperty
rdf:about="http://www.semanticweb.org/userr/ontologies/2020/2/AlarmDi
agnosis#Description"/>
    <!--
http://www.semanticweb.org/userr/ontologies/2020/2/AlarmDiagnosis#Rea
lized_By -->

    <owl:AnnotationProperty
rdf:about="http://www.semanticweb.org/userr/ontologies/2020/2/AlarmDi
agnosis#Realized_By"/>
    <!--
http://www.semanticweb.org/userr/ontologies/2020/2/AlarmDiagnosis#Tit
le -->
    <owl:AnnotationProperty
rdf:about="http://www.semanticweb.org/userr/ontologies/2020/2/AlarmDi
agnosis#Title"/>
    <!--

////////////////////////////////////
////////////////////////////////////
    //
    // Object Properties
    //

////////////////////////////////////
////////////////////////////////////
    -->
    <!--
http://www.semanticweb.org/userr/ontologies/2020/2/AlarmDiagnosis#aff
ects -->

    <owl:ObjectProperty
rdf:about="http://www.semanticweb.org/userr/ontologies/2020/2/AlarmDi
agnosis#affects">
    <rdfs:domain
rdf:resource="http://www.semanticweb.org/userr/ontologies/2020/2/Alar
mDiagnosis#ManufacturingProcess"/>
    <rdfs:range
rdf:resource="http://www.semanticweb.org/userr/ontologies/2020/2/Alar
mDiagnosis#ManufacturingTask"/>

```

```

    </owl:ObjectProperty>
    <!--
http://www.semanticweb.org/userr/ontologies/2020/2/AlarmDiagnosis#causes -->
    <owl:ObjectProperty
rdf:about="http://www.semanticweb.org/userr/ontologies/2020/2/AlarmDiagnosis#causes">
    <rdfs:domain
rdf:resource="http://www.semanticweb.org/userr/ontologies/2020/2/AlarmDiagnosis#ManufacturingEquipment"/>
    <rdfs:domain
rdf:resource="http://www.semanticweb.org/userr/ontologies/2020/2/AlarmDiagnosis#Problem"/>
    <rdfs:range
rdf:resource="http://www.semanticweb.org/userr/ontologies/2020/2/AlarmDiagnosis#Alarm"/>
    <rdfs:range
rdf:resource="http://www.semanticweb.org/userr/ontologies/2020/2/AlarmDiagnosis#ManufacturingProblem"/>
    </owl:ObjectProperty>
    <!--
http://www.semanticweb.org/userr/ontologies/2020/2/AlarmDiagnosis#checkOf -->

    <owl:ObjectProperty
rdf:about="http://www.semanticweb.org/userr/ontologies/2020/2/AlarmDiagnosis#checkOf">
    <rdfs:domain
rdf:resource="http://www.semanticweb.org/userr/ontologies/2020/2/AlarmDiagnosis#ManufacturingAgent"/>
    <rdfs:range
rdf:resource="http://www.semanticweb.org/userr/ontologies/2020/2/AlarmDiagnosis#ManufacturingEquipment"/>
    <rdfs:range
rdf:resource="http://www.semanticweb.org/userr/ontologies/2020/2/AlarmDiagnosis#ManufacturingProblem"/>

////////////////////////////////////
////////////////////////////////////
//
// Data properties
//

////////////////////////////////////
////////////////////////////////////
-->
<!--
http://www.semanticweb.org/userr/ontologies/2020/2/AlarmDiagnosis#AgentAge -->
    <owl:DatatypeProperty
rdf:about="http://www.semanticweb.org/userr/ontologies/2020/2/AlarmDiagnosis#AgentAge">
    <rdfs:subPropertyOf
rdf:resource="http://www.semanticweb.org/userr/ontologies/2020/2/AlarmDiagnosis#AgentDefinition"/>
    <rdfs:domain
rdf:resource="http://www.semanticweb.org/userr/ontologies/2020/2/AlarmDiagnosis#EngineerAgent"/>

```

```

        <rdfs:domain
rdf:resource="http://www.semanticweb.org/userr/ontologies/2020/2/Alar
mDiagnosis#ExpertAgent"/>
        <rdfs:domain
rdf:resource="http://www.semanticweb.org/userr/ontologies/2020/2/Alar
mDiagnosis#ManagerAgent"/>
        <rdfs:domain
rdf:resource="http://www.semanticweb.org/userr/ontologies/2020/2/Alar
mDiagnosis#OperatorAgent"/>
        <rdfs:range
rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
        </owl:DatatypeProperty>
        <!--
http://www.semanticweb.org/userr/ontologies/2020/2/AlarmDiagnosis#Age
ntDefinition -->
        <owl:DatatypeProperty
rdf:about="http://www.semanticweb.org/userr/ontologies/2020/2/AlarmDi
agnosis#AgentDefinition"/>
        <!--
http://www.semanticweb.org/userr/ontologies/2020/2/AlarmDiagnosis#Age
ntDiploma -->
        <owl:DatatypeProperty
rdf:about="http://www.semanticweb.org/userr/ontologies/2020/2/AlarmDi
agnosis#AgentDiploma">
        <rdfs:subPropertyOf
rdf:resource="http://www.semanticweb.org/userr/ontologies/2020/2/Alar
mDiagnosis#AgentDefinition"/>
        <rdfs:domain
rdf:resource="http://www.semanticweb.org/userr/ontologies/2020/2/Alar
mDiagnosis#EngineerAgent"/>
        <rdfs:domain
rdf:resource="http://www.semanticweb.org/userr/ontologies/2020/2/Alar
mDiagnosis#ExpertAgent"/>
        <rdfs:domain
rdf:resource="http://www.semanticweb.org/userr/ontologies/2020/2/Alar
mDiagnosis#ManagerAgent"/>
        <rdfs:domain
rdf:resource="http://www.semanticweb.org/userr/ontologies/2020/2/Alar
mDiagnosis#OperatorAgent"/>

////////////////////////////////////
////////////////////////////////////
//
// Classes
//

////////////////////////////////////
////////////////////////////////////
-->
<!--
http://www.semanticweb.org/userr/ontologies/2020/2/AlarmDiagnosis#Age
ntFeature -->
        <owl:Class
rdf:about="http://www.semanticweb.org/userr/ontologies/2020/2/AlarmDi
agnosis#AgentFeature">
        <rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/userr/ontologies/2020/2/Alar
mDiagnosis#ManufacturingFeature"/> </owl:Class>

```

```

    <!--
http://www.semanticweb.org/userr/ontologies/2020/2/AlarmDiagnosis#Ala
rm -->
    <owl:Class
rdf:about="http://www.semanticweb.org/userr/ontologies/2020/2/AlarmDi
agnosis#Alarm">
    <rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/userr/ontologies/2020/2/Alar
mDiagnosis#AlarmSystem"/>
    </owl:Class>
    <!--
http://www.semanticweb.org/userr/ontologies/2020/2/AlarmDiagnosis#Ala
rmSystem -->
    <owl:Class
rdf:about="http://www.semanticweb.org/userr/ontologies/2020/2/AlarmDi
agnosis#AlarmSystem">
    <rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/userr/ontologies/2020/2/Alar
mDiagnosis#ProductionControl"/>
    </owl:Class>
    <!--
http://www.semanticweb.org/userr/ontologies/2020/2/AlarmDiagnosis#Cau
se -->
    <owl:Class
rdf:about="http://www.semanticweb.org/userr/ontologies/2020/2/AlarmDi
agnosis#Cause">
    <rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/userr/ontologies/2020/2/Alar
mDiagnosis#ManufacturingProblem"/>
    </owl:Class>
    <!--
http://www.semanticweb.org/userr/ontologies/2020/2/AlarmDiagnosis#Com
ercial -->
    <owl:Class
rdf:about="http://www.semanticweb.org/userr/ontologies/2020/2/AlarmDi
agnosis#Comercial">
    <rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/userr/ontologies/2020/2/Alar
mDiagnosis#ManufacturingTask"/>
    </owl:Class>

////////////////////////////////////
////////////////////////////////////
//
// Individuals
//

////////////////////////////////////
////////////////////////////////////
-->
<!--
http://www.semanticweb.org/userr/ontologies/2020/2/AlarmDiagnosis#AL_
SRH_3 -->

    <owl:NamedIndividual
rdf:about="http://www.semanticweb.org/userr/ontologies/2020/2/AlarmDi
agnosis#AL_SRH_3">
    <rdf:type
rdf:resource="http://www.semanticweb.org/userr/ontologies/2020/2/Alar
mDiagnosis#Alarm"/>

```



```

    <AlarmDEscription
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">apparition de
l'hydrogène dans les carters de piliers et les gaines de jeux
de barres</AlarmDEscription>
    <AlarmID
rdf:datatype="http://www.w3.org/2001/XMLSchema#NMToken">Al-SRH-
03</AlarmID>
    <AlarmState
rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean">true</AlarmSt
ate>
    <AlarmValue
rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean">true</AlarmVa
lue>
    <rdfs:comment
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">APPARITION DE
L HYDROGENE DANS LE CARTERS DE PALIERS ET LES GAINES DE JEUX DE
BARRES</rdfs:comment>
    </owl:NamedIndividual>
    <!--
http://www.semanticweb.org/userr/ontologies/2020/2/AlarmDiagnosis#AL_
SRH_7 -->

    <owl:NamedIndividual
rdf:about="http://www.semanticweb.org/userr/ontologies/2020/2/AlarmDi
agnosis#AL_SRH_7">
    <rdf:type
rdf:resource="http://www.semanticweb.org/userr/ontologies/2020/2/Alar
mDiagnosis#Alarm"/>
    <AlarmDEscription
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">augmentation
de concentration d oxygène dans les verrouillages
hydrauliques</AlarmDEscription>
    <AlarmID
rdf:datatype="http://www.w3.org/2001/XMLSchema#NMToken">Al-SRH-
07</AlarmID>
    <AlarmState
rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean">true</AlarmSt
ate>

////////////////////////////////////
////////////////////////////////////
//
// General axioms
//

////////////////////////////////////
////////////////////////////////////
-->
    <owl:Restriction>
    <owl:onProperty
rdf:resource="http://www.semanticweb.org/userr/ontologies/2020/2/Alar
mDiagnosis#CorrectiveAction"/>
    <owl:someValuesFrom
rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <owl:equivalentClass>
    <owl:Restriction>
    <owl:onProperty
rdf:resource="http://www.semanticweb.org/userr/ontologies/2020/2/Alar
mDiagnosis#PreventiveAction"/>

```

```

        <owl:someValuesFrom
rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
        </owl:Restriction>
    </owl:equivalentClass>
</owl:Restriction>
<owl:Restriction>
    <owl:onProperty
rdf:resource="http://www.semanticweb.org/userr/ontologies/2020/2/Alar
mDiagnosis#CorrectiveAction"/>
        <owl:someValuesFrom
rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
        <owl:equivalentClass>
            <owl:Restriction>
                <owl:onProperty
rdf:resource="http://www.semanticweb.org/userr/ontologies/2020/2/Alar
mDiagnosis#RootCause"/>
                <owl:someValuesFrom
rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
                </owl:Restriction>
            </owl:equivalentClass>
        </owl:Restriction>
    <owl:onProperty
rdf:resource="http://www.semanticweb.org/userr/ontologies/2020/2/Alar
mDiagnosis#EquipmentPurchaseDate"/>
        <owl:someValuesFrom
rdf:resource="http://www.w3.org/2001/XMLSchema#dateTime"/>
        <owl:equivalentClass>
            <owl:Restriction>
                <owl:onProperty
rdf:resource="http://www.semanticweb.org/userr/ontologies/2020/2/Alar
mDiagnosis#I-Features"/>
                <owl:someValuesFrom
rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
                </owl:Restriction>
            </owl:equivalentClass>
        </owl:Restriction>
    <owl:onProperty
rdf:resource="http://www.semanticweb.org/userr/ontologies/2020/2/Alar
mDiagnosis#EquipmentPurchaseDate"/>

```

////////////////////////////////////  
////////////////////////////////////  
//  
// Rules  
//  
////////////////////////////////////  
////////////////////////////////////  
-->

```

    <rdf:Description
rdf:about="http://www.semanticweb.org/userr/ontologies/2020/2/AlarmSy
stemDiagnosis#AF">
        <rdf:type
rdf:resource="http://www.w3.org/2003/11/swrl#Variable"/>
    </rdf:Description>
    <rdf:Description
rdf:about="http://www.semanticweb.org/userr/ontologies/2020/2/AlarmSy
stemDiagnosis#EA">

```

```

    <rdf:type
rdf:resource="http://www.w3.org/2003/11/swrl#Variable"/>
    </rdf:Description>
    <rdf:Description
rdf:about="http://www.semanticweb.org/userr/ontologies/2020/2/AlarmSy
stemDiagnosis#N">
    <rdf:type
rdf:resource="http://www.w3.org/2003/11/swrl#Variable"/>
    </rdf:Description>
    <rdf:Description
rdf:about="http://www.semanticweb.org/userr/ontologies/2020/2/AlarmSy
stemDiagnosis#If">
    <rdf:type
rdf:resource="http://www.w3.org/2003/11/swrl#Variable"/>
    </rdf:Description>
    <rdf:Description
rdf:about="http://www.semanticweb.org/userr/ontologies/2020/2/AlarmSy
stemDiagnosis#Pf">
    <rdf:type
rdf:resource="http://www.w3.org/2003/11/swrl#Variable"/>
    </rdf:Description>
    <rdf:Description
rdf:about="http://www.semanticweb.org/userr/ontologies/2020/2/AlarmDi
agnosis#Pr-Refroidissement-Alternateur-Hydrogène"/>
    <rdf:Description
rdf:about="http://www.semanticweb.org/userr/ontologies/2020/2/AlarmSy
stemDiagnosis#FE">
    <rdf:type
rdf:resource="http://www.w3.org/2003/11/swrl#Variable"/>
    </rdf:Description>
    <rdf:Description
rdf:about="http://www.semanticweb.org/userr/ontologies/2020/2/AlarmSy
stemDiagnosis#Cp">
    <rdf:type
rdf:resource="http://www.w3.org/2003/11/swrl#Variable"/>
    </rdf:Description>
    <rdf:Description
rdf:about="http://www.semanticweb.org/userr/ontologies/2020/2/AlarmDi
agnosis#Al_SRH_1"/>
    <rdf:Description
rdf:about="http://www.semanticweb.org/userr/ontologies/2020/2/AlarmSy
stemDiagnosis#Sol">

```

## Annexe C : Requêtes SQWRL de l'ontologie MATO dans Protégé

Name	Query
SWRL_R2_Generic	AlarmDiagnosis:ManufacturingProblem(?Pr) ^ AlarmDiagnosis:ManufacturingFeature(?F) -> AlarmDiagnosis:hasFeature(?Pr, ?F)
SWRL_R3_Generic	AlarmDiagnosis:hasProblem(?E, ?Pr) ^ AlarmDiagnosis:ManufacturingEquipment(?E) ^ AlarmDiagnosis:ManufacturingAgent(?A) ^ AlarmDiagnosis:ManufacturingProblem(?Pr) -> AlarmDiagnosis:hasProblem(?E, ?Pr)
SWRL_R4_Generic	AlarmDiagnosis:hasProblem(?E, ?Pr) ^ AlarmDiagnosis:ManufacturingEquipment(?E) ^ AlarmDiagnosis:ManufacturingProblem(?Pr) -> AlarmDiagnosis:hasProblem(?E, ?Pr)
SWRL_R5_Generic	AlarmDiagnosis:ManufacturingAgent(?A) ^ AlarmDiagnosis:ManufacturingProblem(?Pr) -> AlarmDiagnosis:identifies(?A, ?Pr)
SWRL_R6_Generic	AlarmDiagnosis:ManufacturingSolution(?Sol) ^ AlarmDiagnosis:ManufacturingAgent(?A) ^ AlarmDiagnosis:ManufacturingProblem(?Pr) -> AlarmDiagnosis:proposes(?A, ?Pr)
SWRL_R7_Generic	AlarmDiagnosis:ManufacturingSolution(?Sol) ^ AlarmDiagnosis:ManufacturingAgent(?A) ^ AlarmDiagnosis:ManufacturingProblem(?Pr) ^ AlarmDiagnosis:FaultyAutomatic(?fa) ^ AlarmDiagnosis:hasComponent(?fa, ?Cp) ^ AlarmDiagnosis:Component(?Cp) -> sqwrl:select(?fa, ?Cp)
Select.3	AlarmDiagnosis:Component(?Cp) ^ AlarmDiagnosis:ComponentID(?Cp, ?CpID) ^ AlarmDiagnosis:ComponentType(?Cp, ?CpT) -> sqwrl:select(?Cp, ?CpID, ?CpT)
Select.4	AlarmDiagnosis:Problem(AlarmDiagnosis:Pr-Réfrigérissement-Alternateur-Hydrogène) ^ AlarmDiagnosis:ProblemID(AlarmDiagnosis:Pr-Réfrigérissement-Alternateur-Hydrogène) -> AlarmDiagnosis:ProblemID(AlarmDiagnosis:Pr-Réfrigérissement-Alternateur-Hydrogène)
Sqwrl.1-5	AlarmDiagnosis:ManufacturingEquipment(AlarmDiagnosis:Alternateur) ^ AlarmDiagnosis:EquipmentFeature(?EF) ^ AlarmDiagnosis:hasFeature(AlarmDiagnosis:Alternateur, ?EF) -> AlarmDiagnosis:EquipmentFeature(AlarmDiagnosis:Alternateur, ?EF)
Sqwrl.1-6	AlarmDiagnosis:ManufacturingEquipment(AlarmDiagnosis:Alternateur) ^ AlarmDiagnosis:EquipmentID(AlarmDiagnosis:Alternateur, ?ID) ^ AlarmDiagnosis:EquipmentType(AlarmDiagnosis:Alternateur, ?ID) -> AlarmDiagnosis:EquipmentID(AlarmDiagnosis:Alternateur, ?ID)
Sqwrl.1-7	AlarmDiagnosis:Problem(AlarmDiagnosis:Pr-Réfrigérissement-Alternateur-Hydrogène) ^ AlarmDiagnosis:ExpertAgent(?EA) ^ AlarmDiagnosis:identifies(AlarmDiagnosis:Pr-Réfrigérissement-Alternateur-Hydrogène, ?EA) -> AlarmDiagnosis:ExpertAgent(AlarmDiagnosis:Pr-Réfrigérissement-Alternateur-Hydrogène, ?EA)
Sqwrl1-4	AlarmDiagnosis:Problem(AlarmDiagnosis:Pr-Réfrigérissement-Alternateur-Hydrogène) ^ AlarmDiagnosis:ProblemFeature(?PF) ^ AlarmDiagnosis:hasFeature(AlarmDiagnosis:Pr-Réfrigérissement-Alternateur-Hydrogène, ?PF) -> AlarmDiagnosis:ProblemFeature(AlarmDiagnosis:Pr-Réfrigérissement-Alternateur-Hydrogène, ?PF)

Problem	Agent intervenant	name	AgentID	Exper...	F
AlarmDiagnosis:Pr-Réfrigérissement-Alternateur-Hydrogène	AlarmDiagnosis:Expert3	Bouchemal hassan	"402131..."	"28"AA...	Résolution des problème industriel
AlarmDiagnosis:Pr-Réfrigérissement-Alternateur-Hydrogène	AlarmDiagnosis:Expert4	Zemamouche Fodel	"402336..."	"28"AA...	Assurer le bon fonctionnement de p
AlarmDiagnosis:Pr-Réfrigérissement-Alternateur-Hydrogène	AlarmDiagnosis:Expert2	Ben Akrab elhadi	"402838..."	"24"AA...	Assurer la production Diagnostique
AlarmDiagnosis:Pr-Réfrigérissement-Alternateur-Hydrogène	AlarmDiagnosis:Expert1	Ayet Amran Mohamed	"402536..."	"25"AA...	Assurer la productionResolution de
AlarmDiagnosis:Pr-Réfrigérissement-Alternateur-Hydrogène	AlarmDiagnosis:Expert6	Krid Noureddine	"402139..."	"20"AA...	Assure la bonne continuité de prod
AlarmDiagnosis:Pr-Réfrigérissement-Alternateur-Hydrogène	AlarmDiagnosis:IngProduction7	Lahmar Badredine	"122434..."	"7"AA...	Assurer la production
AlarmDiagnosis:Pr-Réfrigérissement-Alternateur-Hydrogène	AlarmDiagnosis:TPCC4	Arouj Khaled	"602352..."	"9"AA...	Technicien Principale Conduite Cer
AlarmDiagnosis:Pr-Réfrigérissement-Alternateur-Hydrogène	AlarmDiagnosis:IngénieurMaintenance8	Madjedoub Djailil	"125227..."	"3"AA...	Analyser et programmer les interver
AlarmDiagnosis:Pr-Réfrigérissement-Alternateur-Hydrogène	AlarmDiagnosis:IngénieurMaintenance5	Boufelaas Nouredine	"124328..."	"20"AA...	Analyser et programmer les interver
AlarmDiagnosis:Pr-Réfrigérissement-Alternateur-Hydrogène	AlarmDiagnosis:TPC2	Hamata Faycal	"172436..."	"19"AA...	Technicien Principale Chimiste
AlarmDiagnosis:Pr-Réfrigérissement-Alternateur-Hydrogène	AlarmDiagnosis:ChefMaintenance3	Harouche Mustafa	"122937..."	"13"AA...	Analyser et programmer les interver
AlarmDiagnosis:Pr-Réfrigérissement-Alternateur-Hydrogène	AlarmDiagnosis:IngProduction9	Feridja karnel	"122437..."	"16"AA...	assurer la production
AlarmDiagnosis:Pr-Réfrigérissement-Alternateur-Hydrogène	AlarmDiagnosis:ChefMaintenance1	Aissani Abdellah	"125026..."	"11"AA...	Maintenance des instrument en aut
AlarmDiagnosis:Pr-Réfrigérissement-Alternateur-Hydrogène	AlarmDiagnosis:IngProduction6	Merdassi khalid	"127733..."	"20"AA...	assurer la production

Figure C.1 Sélection la connaissance des agents intervenant par SQWRL (sqwrl1-7)

Expert intervenant	Name	ExpertFeature	P-Features	I-Features
AlarmDiagnosis:Expert5	Saidani habib	AlarmDiagnosis:ExpFeature5	travail dans le système de froid	resolution des problème de réfrigérisseme
AlarmDiagnosis:Expert4	Zemamouche Fodel	AlarmDiagnosis:ExpFeature4	la capacité de huit heur par jr	résolution de probleme lié à l hydrogène
AlarmDiagnosis:Expert3	Bouchemal hassan	AlarmDiagnosis:ExpFeature3	travail le temps d'arrêt en maintenance préventive et cor...	mise en marche d'arrêt du système de refr
AlarmDiagnosis:Expert2	Ben Akrab elhadi	AlarmDiagnosis:ExpFeature2	maintenir le système de réfrigérissement et l'installation ...	responsable de mise en marche et d'arrêt
AlarmDiagnosis:Expert3	Bouchemal hassan	AlarmDiagnosis:ExpFeature5	travail dans le système de froid	resolution des problème de réfrigérisseme
AlarmDiagnosis:Expert3	Bouchemal hassan	AlarmDiagnosis:ExpFeature2	maintenir le système de réfrigérissement et l'installation ...	responsable de mise en marche et d'arrêt
AlarmDiagnosis:Expert3	Bouchemal hassan	AlarmDiagnosis:ExpFeature4	la capacité de huit heur par jr	résolution de probleme lié à l hydrogène
AlarmDiagnosis:Expert5	Saidani habib	AlarmDiagnosis:ExpFeature3	travail le temps d'arrêt en maintenance préventive et cor...	mise en marche d'arrêt du système de refr
AlarmDiagnosis:Expert5	Saidani habib	AlarmDiagnosis:ExpFeature2	maintenir le système de réfrigérissement et l'installation ...	responsable de mise en marche et d'arrêt
AlarmDiagnosis:Expert5	Saidani habib	AlarmDiagnosis:ExpFeature4	la capacité de huit heur par jr	résolution de probleme lié à l hydrogène
AlarmDiagnosis:Expert2	Ben Akrab elhadi	AlarmDiagnosis:ExpFeature3	travail le temps d'arrêt en maintenance préventive et cor...	mise en marche d'arrêt du système de refr

Figure C.2 Sélection les caractéristiques physiques et informationnelles des agents intervenant par SQWRL (sqwrl1-8)

Problem	Alarm	Faulty Equipment	
AlarmDiagnosis:Pr-Réfrigérissement-Alternateur-Hydrogène	AlarmDiagnosis:AL_SRH_1	AlarmDiagnosis:FaultyEquip.SRH.1	AlarmDiagnosis:Etanchéité-aux-gaz-du-TurboAlte
AlarmDiagnosis:Pr-Réfrigérissement-Alternateur-Hydrogène	AlarmDiagnosis:AL_SRH_1	AlarmDiagnosis:FaultyEquip.SRH.2	AlarmDiagnosis:Système-de-Gaz

Figure C.3 Sélection les causes des problèmes par SQWRL (sqwrl1-9)

## Annexe D : Requêtes SPARQL

```

1 PREFIX ns: <http://www.semanticweb.org/userc/ontologies/2020/2/AlarmDiagnosis#>
2 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
3
4 SELECT ?Pr ?PrF ?IFeatures ?PFeatures
5 WHERE {?PrC rdf:type ns:Problem .
6         ?PrF rdf:type ns:ProblemFeature .
7         ?PrC ns:hasFeature ?PrF .
8         ?PrF ns:P-Features ?PFeatures .
9         ?PrF ns:I-Features ?IFeatures .
10        }
11

```

Console

```

<terminated> Main (5) [Java Application] C:\Program Files\Java\jre1.8.0_111\bin\javaw.exe (25 juin 2020 à 00:58:26 – 00:58:31)
-----
| Pr                                | PrF                                | IFeatures
-----|-----|-----
| ns:Pr-Refroidissement-Alternateur-Hydrog??ne | ns:PrRefroidissementFeature | "Le maintien des param??
-----

```

Figure D.1 Résultats SPARQL pour l'étape 3.1

```

1 PREFIX ns: <http://www.semanticweb.org/userc/ontologies/2020/2/AlarmDiagnosis#>
2 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
3
4 SELECT ?PrC ?AgentIntervenant ?Name ?AgentID ?Age ?Diploma ?Experience ?Fonction
5 WHERE {?PrC rdf:type ns:Problem .
6         ?AgentIntervenant rdf:type ns:ExpertAgent .
7         ?AgentIntervenant ns:identifies ?PrC .
8         ?AgentIntervenant ns:AgentName ?Name .
9         ?AgentIntervenant ns:AgentID ?AgentID .
10        ?AgentIntervenant ns:AgentAge ?Age .
11        ?AgentIntervenant ns:AgentDiploma ?Diploma .
12        ?AgentIntervenant ns:AgentExperience ?Experience .
13        ?AgentIntervenant ns:AgentFunction ?Fonction .
14        }
15
16

```

Console

```

<terminated> Main (5) [Java Application] C:\Program Files\Java\jre1.8.0_111\bin\javaw.exe (25 juin 2020 à 00:59:56 – 01:00:02)
-----
| Pr                                | AgentIntervenant                    | Name
-----|-----|-----
| ns:Pr-Refroidissement-Alternateur-Hydrog??ne | ns:Expert2                          | "Ben Akrab elhadi"^^<http:
| ns:Pr-Refroidissement-Alternateur-Hydrog??ne | ns:Expert4                          | "Zemamouche Fodel"^^<http:
| ns:Pr-Refroidissement-Alternateur-Hydrog??ne | ns:Expert3                          | "Bouchemal hassan"^^<http:
| ns:Pr-Refroidissement-Alternateur-Hydrog??ne | ns:Expert1                          | "Ayet Amran Mohamed"^^<htt
| ns:Pr-Refroidissement-Alternateur-Hydrog??ne | ns:Expert5                          | "Saidani habib"^^<http://w
| ns:Pr-Refroidissement-Alternateur-Hydrog??ne | ns:IngProduction6                   | "Merdassi khalid"^^<http://
| ns:Pr-Refroidissement-Alternateur-Hydrog??ne | ns:IngProduction2                   | "Hamadouche sallah"^^<http
| ns:Pr-Refroidissement-Alternateur-Hydrog??ne | ns:Ing??nieurMaintenance9          | "Guellil Aissam"^^<http://
| ns:Pr-Refroidissement-Alternateur-Hydrog??ne | ns:ChefQuartDeProduction2          | "Tebboub Billel"^^<http://
| ns:Pr-Refroidissement-Alternateur-Hydrog??ne | ns:TPC5                             | "Sauli Abdelhakim"^^<http:
| ns:Pr-Refroidissement-Alternateur-Hydrog??ne | ns:Ing??nieurMaintenance2          | "Amara Rabah"^^<http://w
-----

```

Figure D.2 Résultats SPARQL pour l'étape 3.2



```

Main.java step3.3.txt
1 PREFIX ns: <http://www.semanticweb.org/user/ontologies/2020/2/AlarmDiagnosis#>
2 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
3
4 SELECT ?Problem ?FE ?Componentfailer
5 WHERE {
6 ?Problem rdf:type ns:Problem .
7 ?FE rdf:type ns:FaultyEquipment .
8 ?Problem ns:hasCause ?FE .
9 ?Componentfailer rdf:type ns:Component .
10 ?FE ns:hasComponent ?Componentfailer .
11 }

```

Console

<terminated> Main (5) [Java Application] C:\Program Files\Java\jre1.8.0\_111\bin\javaw.exe (25 juin 2020 à 00:51:50 – 00:51:59)

Problem	FE	Componentfailer
ns:Pr-Refrondissement-Alternateur-Hydrog??ne	ns: faultyEquip.SRH.2	ns:Syst??me-de-Gaz
ns:Pr-Refrondissement-Alternateur-Hydrog??ne	ns: FaultyEquip.SRH.1	ns: Etancheit??-aux-gaz-du-TurboA:

Figure D.3 Résultats SPARQL pour l'étape 3.3

```

Main.java step3.4.txt
1 PREFIX ns: <http://www.semanticweb.org/user/ontologies/2020/2/AlarmDiagnosis#>
2 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
3 PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
4
5 SELECT ?Pr ?alarm ?AgentIntervenant ?Name ?Function
6 WHERE {
7 ?alarm rdf:type ns:Alarm .
8 ?Pr ns:hasAlarm ?alarm .
9 ?Pr rdf:type ns:Problem .
10 ?AgentIntervenant rdf:type ns:ExpertAgent .
11 ?AgentIntervenant ns:identifies ?Pr .
12 ?AgentIntervenant ns:AgentName ?Name .
13 ?AgentIntervenant ns:AgentID ?AgentID .
14 ?AgentIntervenant ns:AgentFunction ?Function .
15 }
16
17

```

Console

<terminated> Main (5) [Java Application] C:\Program Files\Java\jre1.8.0\_111\bin\javaw.exe (25 juin 2020 à 00:55:07 – 00:55:12)

	alarm	AgentIntervenant	Name	Function
ydrog??ne	ns:AL_SRH_1	ns:Expert2	"Ben Akrab elhadi"^^xsd:string	"Assurer la pr
ydrog??ne	ns:AL_SRH_1	ns:Expert4	"Zemamouche Fodel"^^xsd:string	"Assurer le bc
ydrog??ne	ns:AL_SRH_1	ns:Expert3	"Bouchemal hassan"^^xsd:string	"R??solution c
ydrog??ne	ns:AL_SRH_1	ns:Expert1	"Ayet Amran Mohamed"^^xsd:string	"Assurer la pr
ydrog??ne	ns:AL_SRH_1	ns:Expert5	"Saidani habib"^^xsd:string	"Assurer le bc
ydrog??ne	ns:AL_SRH_1	ns:IngProduction6	"Merdassi khalid"^^xsd:string	"assurer la pr
ydrog??ne	ns:AL_SRH_1	ns:IngProduction2	"Hamadouche sallah"^^xsd:string	"Assurer la pr
ydrog??ne	ns:AL_SRH_1	ns:Ing??nieurMaintenance9	"Guellil Aissam"^^xsd:string	"Analyser et
ydrog??ne	ns:AL_SRH_1	ns:ChefQuantDeProduction2	"Tebboub Billel"^^xsd:string	"Assure la bor
ydrog??ne	ns:AL_SRH_1	ns:TPC5	"Sauli Abdelhakim"^^xsd:string	"Technicien Pr

Figure D.4 Résultats SPARQL pour l'étape 3.4

```

Main.java step4.1.txt
1 PREFIX ns: <http://www.semanticweb.org/user/ontologies/2020/2/AlarmDiagnosis#>
2 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
3
4 SELECT ?Pr ?FE ?componentFailure ?Sol ?DSolution
5 WHERE {
6 ?Pr rdf:type ns:Problem .
7 ?FE rdf:type ns:FaultyEquipment .
8 ?componentFailure rdf:type ns:Component .
9 ?Sol rdf:type ns:Solution .
10 ?Pr ns:hasCause ?FE .
11 ?FE ns:hasComponent ?componentFailure .
12 ?componentFailure ns:hasSolution ?Sol .
13 ?Solution ns:SolutionDescription ?DSolution .
14 }
15

```

Console

<terminated> Main (5) [Java Application] C:\Program Files\Java\jre1.8.0\_111\bin\javaw.exe (25 juin 2020 à 01:01:11 – 01:01:17)

Sol	DSolution
ns:Sol.SRH.2	"A la diff??rence nominale Huile Hydrog??ne"^^<http://www.w3.org/2001/XMLSchema#string
ns:Sol.SRH.2	"Mettre sousPression le corp deStator ensemble avec le syst??me de Gaz et d Huile"^^<h
ns:Sol.SRH.1	"A la diff??rence nominale Huile Hydrog??ne"^^<http://www.w3.org/2001/XMLSchema#string
ns:Sol.SRH.1	"Mettre sousPression le corp deStator ensemble avec le syst??me de Gaz et d Huile"^^<h

Figure D.5 Résultats SPARQL pour l'étape 4.1

The screenshot shows a Java IDE with two tabs: 'Main.java' and 'step4.2.txt'. The 'Main.java' tab contains a SPARQL query. The console window below shows the execution of the query, resulting in a table of two rows. The first row has columns 'I' and 'Itp'. The second row has columns 'ns:InterVtype2' and a long string describing an intervention: '"Intervention Corective et appele aux service maintenance"^^<http://www.w3.org/2001/XMLSchema#textLiteral>'. The first row has columns 'I' and 'Itp'. The second row has columns 'ns:InterVtype1' and a long string describing an intervention: '"Intervention Hydraulique et Intervention Chimique"^^<http://www.w3.org/2001/XMLSchema#textLiteral>'. The console output is as follows:

```

1 PREFIX ns: <http://www.semanticweb.org/user/ontologies/2020/2/AlarmDiagnosis#>
2 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
3
4 SELECT ?Sol ?DS ?I ?Itp
5 WHERE {?Sol rdf:type ns:Solution .
6        ?I   rdf:type ns:InterventionType .
7        ?Sol ns:SolutionDescription ?DS .
8        ?Sol ns:hasInterventionType ?I .
9        ?I   ns:interventionType    ?Itp .
10       }
11
12
<terminated> Main (5) [Java Application] C:\Program Files\Java\jre1.8.0_111\bin\javaw.exe (25 juin 2020 à 01:02:42 - 01:02:47)
-----
| I           | Itp
-----
| ns:InterVtype2 | "Intervention Corective et appele aux service maintenance"^^<http://www.w3.org/2001/XMLSchema#textLiteral>
g> | ns:InterVtype1 | "Intervention Hydraulique et Intervention Chimique"^^<http://www.w3.org/2001/XMLSchema#textLiteral>
-----

```

Figure D.6 Résultats SPARQL pour l'étape 4.2

## Annexe E: PPASD (Power Plant Alarm System Diagnosis)

### MVC (Modèle Vue-Contrôleur)

#### E.1. Le modèle

##### E.1.1. lecture des fichiers

```

import java.io.BufferedReader;
import java.io.File;
import java.io.FileReader;
import java.io.IOException;

/**
 *
 * @author DO.IT-TITAH
 * Ce fichier sert à
 * + Lire le contenu d'un fichier: getContents
 * + Entrée: Objet File
 * + Sortie: une chaîne de caractère
 */
public class FileTool {

    static public String getContents(File a File) {
        //...checks on a File
        StringBuilder contents = new StringBuilder();

        try {
            //FileReader always default encoding is OK!
            BufferedReader input = new BufferedReader(new
FileReader(aFile));
            try {

```

```

        String line = null; // * Reader always ,it returns an
empty String in a row.
        */
        while ((line = input.readLine()) != null) {
            contents.append(aline);
            contents.append(System.getProperty("line.separator"));
        }
    } finally {
        input.close();
    }
} catch (IOException ex) {
    ex.printStackTrace();
}

```

## E.1.2. Chargement de l'ontologie, moteur Jena, et l'exécution des requêtes SPARQL

```

package utils;

import java.io.File;
import java.io.InputStream;
import java.util.List;
import com.hp.hpl.jena.query.Query;
import com.hp.hpl.jena.query.QueryExecution;
import com.hp.hpl.jena.query.QueryExecutionFactory;
import com.hp.hpl.jena.query.QueryFactory;
import com.hp.hpl.jena.query.ResultSet;
import com.hp.hpl.jena.query.ResultSetFormatter;
import com.hp.hpl.jena.rdf.model.*;
import com.hp.hpl.jena.reasoner.rulesys.GenericRuleReasoner;
import com.hp.hpl.jena.reasoner.rulesys.Rule;
import com.hp.hpl.jena.util.FileManager;
import java.io.IOException;
import java.io.OutputStream;

/**
 *
 * @author DO.TITAH
 */
public class JenaEngine {
    static private String RDF = "http://www.w3.org/1999/02/22-rdf-syntax-ns#";

    /**
     * Charger un modèle à partir d'un fichier owl
     * @param args
     * + Entree: le chemin vers le fichier owl
     * + Sortie: l'objet model jena
     */
    static public Model readModel(String inputDataFile) {
//        create an empty model
        Model model = ModelFactory.createDefaultModel();

//        use the FileManager to find the input file
        InputStream in = FileManager.get().open(inputDataFile);
        if (in == null) {
            System.out.println("Ontology file: " + inputDataFile + " not
found");

```



```

        return null;
    }
    // read the RDF/XML file
    model.read(in, "");
    try {
        in.close();
    } catch (IOException e) {
        // TODO Auto-generated catch block
        return null;
    }
    return model;
}
/**
 * Faire l'inference
 * @param args
 * + Entree: l'objet model Jena avec le chemin du fichier de regles
 * + Sortie: l'objet model infere Jena
 */
static public Model readInferencedModelFromRuleFile(Model model, String
inputRuleFile) {
    InputStream in = FileManager.get().open(inputRuleFile);
    if (in == null) {
        System.out.println("Rule File: " + inputRuleFile + " not
found");
        return null;
    } else {
        try {
            in.close();
        } catch (IOException e) {
            // TODO Auto-generated catch block
            return null;
        }
    }
    List rules = Rule.rulesFromURL(inputRuleFile);
    GenericRuleReasoner reasoner = new GenericRuleReasoner(rules);
    reasoner.setDerivationLogging(true);
    reasoner.setOWLTranslation(true); // not needed in
RDFS case
    reasoner.setTransitiveClosureCaching(true);
    InfModel inf = ModelFactory.createInfModel(reasoner, model);
    return inf;
}
/**
 * Executer une requete
 * @param args
 * + Entree: l'objet model Jena avec une chaine des caracteres SparQL
 * + Sortie: le resultat de la requete en String
 */
static public String executeQuery(Model model, String queryString) {
    Query query = QueryFactory.create(queryString);
    QueryExecution qe = QueryExecutionFactory.create(query, model);
    ResultSet results = qe.execSelect();
    OutputStream output = new OutputStream() {
        private StringBuilder string = new StringBuilder();
        @Override
        public void write(int b) throws IOException {
            this.string.append((char) b);
        }
    }
}

```

```

        public String toString() {
            return this.string.toString();
        }
    };
    ResultSetFormatter.out(output, results, query);
    return output.toString();
}
static public ResultSet executeQueryResultSet(Model model, String
queryString) {
    Query query = QueryFactory.create(queryString);

    QueryExecution qe = QueryExecutionFactory.create(query, model);
    ResultSet results = qe.execSelect();
    return results;
}

```

## E.2. La vue

### E.2.1 extrait de code JSP

```

<%
        if (selectedEquip == null) { %>
            Select Equipment: <input type="text" id="Equip" name="Equip"
                readonly="readonly" /> <span
class="error">${messages.Equip}</span>
            <%
                } else {
                    %>
            Select Equipment: <input type="text" id="Equip" name="Equip"
                readonly="readonly" value=<%=selectedEquip%>> <span
class="error">${messages.Equip}</span>
            <%
                }
            %
            <br /> <br /> <br /> <br /> 
            <map name="production">
                <area target="_blank" alt="Alternateur"
title="Alternateur" href=""
                onclick="myFunction('Alternateur');return false;"
                coords="587,161,27" shape="circle">
                <area target="_blank" alt="condenseur"
title="condenseur" href=""
                onclick="myFunction('condenseur');return false;"
                coords="462,304,561,354" shape="rect">
                <area target="_blank" alt="pompes d'extraction-1 -étage "
                title="pompes d'extraction-1 -étage " href=""
                onclick="myFunction('pompes d'extraction-1 -
étage');return false;"

```

### E.2.2 Extrait de code html de la page d'accueil

```

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<link rel="stylesheet" type="text/css" href="titahStyle.css" />
<title>Home</title>
</head>
<body>
    <div class="header_area">
        <div class="main_area">

```

```

        <p class="collus_text"></p>
        <p class="Logo_pad">
border="0" /></a>
        <a href="#">
        </div>
        <li><a
        <div class="header_text_wrap">
        <p class="fresh_text">Find Now! Your</p>
        <br /> <span class="you_text"> Alarm System Diagnosis
        </span> <br />
        <!-- <p style="padding: 20px 0 0 0;">
        <a href="Menu.html" class="read_botn">Click Here</a>
        </p -->
        <ul>
        <li><a href="/AlarmDiagnosisApp/Diagnosis">Etape.1.1:
        Identification la tâche, le processus
et la tâche lié au problème</a></li>

```

## E.3 Le contrôleur

### E.3.1 Class AlarmStepsServlet

```

package web;
import com.hp.hpl.jena.rdf.model.Model;
import utils.JenaEngine;
import web.AlarmStepsServlet;
import java.io.IOException;
import java.util.HashMap;
import java.util.Map;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
public class AlarmStepsServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;
    protected void doGet(HttpServletRequest request, HttpServletResponse
response)
        throws ServletException, IOException {
        request.getRequestDispatcher("/jsp/index1.1.jsp").forward(request,
response);
    }
    protected void doPost(HttpServletRequest request, HttpServletResponse
response)
        throws ServletException, IOException {
        Map<String, String> messages = new HashMap<String, String>();
        String prob = request.getParameter("Prob");
        String al = request.getParameter("Al");
        String equip = request.getParameter("Equip");
        String step = request.getParameter("Step");
        if (prob.equals("0")) {
            messages.put("Prob", "Please select number of Problem");
        }
        if (al.equals("0")) {
            messages.put("Al", "Please select alarm code");
        }
        if (equip == "") {
            messages.put("Equip", "Please select equipment");
        }
    }
}

```

```

        if (step.equals("0")) {
            messages.put("Step", "Please select the step");
        }
        if (!messages.isEmpty()) {
            messages.clear();
            request.getRequestDispatcher("/jsp/index1.1.jsp").forward(request,
response);
        } else {
            HttpSession session = request.getSession();
            session.setAttribute("Prob", prob);
            session.setAttribute("Al", al);
            session.setAttribute("Equip", equip);
            Model model =
JenaEngine.readModel("data/ApplicationMAT01.owl");

            if (prob.equals("1") && al.equals("1") &&
step.equals("1")) {
                String query11 = getQueryStep11();
                String result =
JenaEngine.executeQuery(model, query11);
                /*
                System.out.println("Result : " + result);
                request.getRequestDispatcher(
                    (new
StringBuilder()).append("/jsp/result1.1.jsp?result=").append(result).toStri
ng())
                    .forward(request, response);
            } else if (prob.equals("1") && al.equals("1") &&
step.equals("2")) {
                String query12 = getQueryStep12();
                String result =
JenaEngine.executeQuery(model, query12);

```

### E.3.2 Exemple d'exécution d'une requête

```

private String getQueryStep11() {
    String query11 = "PREFIX ns:
<http://www.semanticweb.org/user/ontologies/2020/2/AlarmDiagnosis#>\r\n"
        + "PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-
syntax-ns#>\r\n"
        + "PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
\r\n"
        + "SELECT ?Al ?AD ?AS ?D ?ID ?N ?T
?Tp \r\n"
        + "WHERE {?D rdf:type ns:Diagnosis .\r\n" + "
?Al rdf:type ns:Alarm .\r\n"
        + " ?Al ns:AlarmState ?AS .\r\n" + "
?Al ns:AlarmDEscription ?AD .\r\n"
        + " ?Al ns:AlarmDEscription \"BASE PURETE
D'HYDROGENE DANS LE CORPS TURBOALTERNATEUR\" .\r\n"
        + " ?D ns:TaskID ?ID .\r\n" + " ?D
ns:TaskName ?N .\r\n"
        + " ?D ns:TaskTime ?T .\r\n" + " ?D
ns:TaskType ?Tp .\r\n" + "}" ;

    System.out.println("Query 1.1 : " + query11);
    return query11;
}

```

Dans cette thèse, nous nous sommes intéressés à un modèle de gestion des connaissances des entreprises industrielles. Certaines tâches manufacturières impliquent un niveau élevé de connaissance tacite des opérateurs qualifiés. L'industrie a besoin des méthodes fiables pour la capture et l'analyse de ces connaissances tacites afin qu'elles puissent être partagées et sans aucune perte. Nous proposons, un modèle de gestion contenant deux processus de gestion, le premier processus est la capitalisation des connaissances basée sur une tâche industrielle. Nous avons utilisé une combinaison de deux méthodologies : une méthodologie d'ingénierie de connaissances CommonKADS et une méthodologie d'élicitation des connaissances MACTAK. Dans la phase de modélisation, nous avons utilisé deux différentes techniques de modélisation, une modélisation basée sur les connaissances d'expert et la deuxième une représentation ontologique. Ce modèle facilite la capture des connaissances d'experts et transforme les connaissances tacites en explicites avec une maximisation des règles de production. Le deuxième processus concerne le partage des connaissances à base d'une ontologie des Tâches Manufacturières MATO en identifiant un ensemble des concepts de fabrication et leurs relations, cette ontologie proposée facilite le partage des connaissances entre les tâches de fabrication et aide à partager et à réutiliser les connaissances durant l'exécution des tâches. Ensuite, une application proposée pour le diagnostic de système d'alarme dans une centrale thermique a été présentée pour démontrer l'importance et l'apport de l'ontologie.

**Mots-clés :** *Connaissances tacites, capitalisation des connaissances, partage des connaissances, CommonKADS, Ontologie, MATO, OWL, SWRL, requêtes SQWRL, requêtes SPARQL*

### Improvement of capitalization and sharing of knowledge to maximize the value of production system

### Abstract

In this thesis, we are interested in a knowledge management model for industrial companies. Some manufacturing tasks imply a high level of tacit knowledge which exists only in internal cognitions of qualified operators. Industry needs reliable methods for capturing and analyzing this tacit knowledge so that it can be shared without any lost. Therefore, we propose two management model processes. The first process is knowledge capitalization based on an industrial task, where we used a combination of two methodologies: a CommonKADS knowledge engineering methodology and an elicitation methodology. MACTAK knowledge is used in the modeling phase. Two different modeling techniques are used, an expert-based modeling and the second an ontological representation. This model facilitates the capture of expert knowledge and transforms tacit knowledge into explicit knowledge with the maximization of production rules. The second process is sharing knowledge based on ontology of MATO Manufacturing Tasks by identifying set of manufacturing concepts and their relationships. The proposed ontology facilitates the sharing of knowledge between manufacturing tasks and helps to share and reuse knowledge during task execution. Finally, an application of our contribution to the alarm system diagnosis in a thermal power plant was presented to demonstrate the importance of our contribution.

**Keywords:** *Tacit knowledge, knowledge capitalization, knowledge sharing, CommonKADS, Ontology, MATO, OWL, SWRL, SQWRL query, SPARQL query.*

### تحسين عملية رسملة و تبادل المعرفة لزيادة قيمة نظام الإنتاج

### ملخص

في هذه الأطروحة نهتم بنموذج إدارة المعرفة للشركات الصناعية. تتطلب بعض مهام التصنيع مستوى عالٍ من المعرفة الضمنية التي لا توجد إلا في الإدراك الداخلي للمشغلين المؤهلين. تحتاج الصناعة إلى طرق موثوقة لالتقاط هذه المعرفة الضمنية وتحليلها بحيث يمكن مشاركتها وعدم فقدها. لذلك نقترح أن يحتوي نموذج الإدارة على عمليتين إداريتين، الأولى هي رسملة المعرفة على أساس مهمة صناعية. استخدمنا مزيجاً من منهجيتين: منهجية هندسة المعرفة CommonKADS ومنهجية الاستنباط MACTAK. في مرحلة النمذجة، استخدمنا طريقتين مختلفتين، النمذجة القائمة على الخبراء والثانية التمثيل الأنطولوجي. يسهل هذا النموذج الحصول على معرفة الخبراء ويحول المعرفة الضمنية إلى معرفة واضحة مع تعظيم قواعد الإنتاج. من خلال تحديد مجموعة من مفاهيم التصنيع وعلاقاتها. تسهل هذه الأنطولوجيا المقترحة مشاركة المعرفة بين مهام التصنيع وتساعد على مشاركة المعرفة وإعادة استخدامها أثناء تنفيذ المهمة. تم تطبيق هذا النموذج في تشخيص نظام الإنذار في محطة الطاقة الحرارية لإثبات أهمية بحثنا.

**المفاتيح:** *المعرفة الضمنية، رسملة المعرفة، تبادل المعرفة، علم الوجود Ontologie، علم الوجود لمهمة MATO، لغة قواعد الويب الدلالية SWRL، طلبات SQWRL، طلبات SPARQL، لغة الويب الوجود OWL.*