



AVERTISSEMENT

Ce document est le fruit d'un long travail approuvé par le jury de soutenance et mis à disposition de l'ensemble de la communauté universitaire élargie.

Il est soumis à la propriété intellectuelle de l'auteur. Ceci implique une obligation de citation et de référencement lors de l'utilisation de ce document.

D'autre part, toute contrefaçon, plagiat, reproduction illicite encourt une poursuite pénale.

Contact : ddoc-theses-contact@univ-lorraine.fr

LIENS

Code de la Propriété Intellectuelle. articles L 122. 4

Code de la Propriété Intellectuelle. articles L 335.2- L 335.10

http://www.cfcopies.com/V2/leg/leg_droi.php

<http://www.culture.gouv.fr/culture/infos-pratiques/droits/protection.htm>

THÈSE

présentée pour obtenir le grade de

Docteur

de

L'Université De Lorraine

en

Automatique, Traitement du Signal et des Images, Génie Industriel

Spécialité : **Automatique**

par

Fahd AMJAD

Approche ontologique pour l'intégration des entreprises distribuées

Soutenue le 12 décembre 2012, devant le jury composé de :

M. Bernard ESPINASSE

Professeur à Aix-Marseille Université, Marseille

M. Bernard GRABOT

Professeur à l'École Nationale d'Ingénieurs, Tarbes

M. Didier ANCIAUX

HDR, Maître de Conférences à l'Université de Lorraine, Metz

M. Hervé PINGAUD

Professeur au Centre Universitaire J. F. Champollion, Albi

M. Daniel ROY

Maître de Conférences à l'École Nationale d'Ingénieurs, Metz

M. André THOMAS

Professeur à l'Université de Lorraine, Nancy

Rapporteur

Rapporteur

Directeur de thèse

Examineur

Co-directeur de thèse

Examineur

To my Parents

To my Brother and Sister

To my Nephews and Niece

To my Wife

Remerciements

Je tiens, tout d'abord, à remercier mon directeur de thèse, Monsieur Didier ANCIAUX pour ses précieux conseils basés sur son expertise. Ses remarques claires, concises et pertinentes au cours de diverses réunions qui ont aidé à l'établissement de l'orientation de ma recherche.

Je suis aussi reconnaissant de l'appui et des conseils fournis par mon co-directeur de thèse Monsieur Daniel ROY. Ses explications pertinentes, son aide et notre coopération sont les clés de la réussite de ce travail de recherche. Je suis également reconnaissant pour les efforts inlassables qu'il a déployés. Travailler avec lui a été une expérience d'apprentissage et l'un des principaux acquis de ces quatre années.

Je remercie, également, messieurs Bernard ESPINASSE et Bernard GRABOT pour l'intérêt qu'ils ont porté à cette thèse en acceptant de la rapporter. Je remercie, aussi, messieurs Hervé PINGAUD et André THOMAS de m'avoir fait l'honneur de juger mon travail. Enfin, je tiens à remercier mon agence de financement, « La Commission de l'Enseignement Supérieur du Pakistan », qui a rendu cette thèse possible.

Mon passage à l'Université de Lorraine a été agréable, en particulier grâce de mes collègues Maen Atli et Akram Zougari. Je les remercie pour leur soutien moral et aussi pour l'ambiance de bureau agréable et joyeuse.

Je tiens à remercier mes amis Liaqat, Armaghan, Sadiq, Shoaib, et Sajid pour leur temps, leurs efforts et pour leur soutien constant. Ils ont été ma famille de substitution au cours de ces années.

Je dois ma plus profonde gratitude à ma mère et à mon père, d'avoir été un soutien au cours de toutes les épreuves et difficultés que j'ai rencontrées au cours de cette thèse. Je les en remercie de m'avoir épaulé dans mes moments de stress et de désespoir. Sans leurs soutiens et leurs encouragements constants, cela n'aurait pas été possible.

Résumé

Dans cette thèse, nous fournissons un examen complet des technologies du Web sémantique et de leurs utilités dans le contexte actuel des petites et moyennes entreprises (PME). Les approches traditionnelles d'intégration des entreprises favorisent essentiellement les grandes entités. Les obligations contractuelles fortes sur les PME, mais en même temps leur volonté de garder leurs compétences individuelles, et ce, dans un environnement limitant leur choix, les obligent à prendre des décisions stratégiques et de conclure des accords sur le long terme avec leurs partenaires, limitant ainsi leur flexibilité aux fluctuations du marché. Nous proposons, donc, une approche ontologique basée sur Web sémantique pour l'intégration de l'information ainsi que des ressources matérielles de l'entreprise distribuée. Cette approche, basée sur le Web, agit comme un système d'aide à la décision pour utiliser des ressources de meilleure qualité ainsi que pour l'intégration de l'information distribuée. Les travaux relatifs à l'ontologie web, pour l'intégration d'information ne sont pas nouveaux, mais l'approche proposée par nous est une valeur ajoutée pour l'entreprise distribuée. De plus, nous avons également proposé l'ontologie Web sémantique comme un système de configuration pour gérer les ressources distribuées de l'entreprise virtuelle. Puis, nous avons modélisé l'ontologie OWL-DL en nous basant sur la sémantique de la norme ISA-95, relative à l'intégration d'entreprises industrielles. Ensuite, nous utilisons cet artefact ontologique comme un artefact de configuration permettant de gérer le matériel de l'entreprise virtuelle distribuée ainsi que les ressources matérielles. C'est la proposition principale de cette thèse : utiliser l'ontologie Web sémantique comme un système d'aide à la décision pour la configuration de l'utilisation des ressources.

Mots-clés : *Intégration d'Entreprise, Réseaux Logistiques, Partage de l'Information, Coordination, Approche Basée sur l'Ontologie, ISA-95, Langage Ontologique Web (OWL), Logique Descriptive, Protégé, Java Expert System Shell (JESS).*

Abstract

In this thesis, we have provided a complete review of the semantic web technologies and their corresponding utility in the current environment for small to medium sized enterprise (MSE). The traditional approaches to enterprise integration favour large enterprise entities and force contractual limitations on smaller partners, but at the same time the pressure to guard the individual enterprise competence is ever increasing, the distributed enterprise (MSE) in such an environment have limited number of choices, which forces them to make strategic decisions and enter into a long term agreements with their partners and this limits their flexibility to the market changes. We, in this thesis, propose a semantic web based ontology approach for integrating the information as well as physical resource of the distributed enterprise. This web based approach acts as a decision support for better resources utility as well as distributed information integration. The work related to web ontology's for information integration is not new, but the approach proposed in this thesis for distributed enterprise is an added value. Similarly, we have also proposed semantic web ontology as a configuration system to manage the distributed resources of the virtual enterprise, for this we have modelled OWL-DL ontology on the semantic of the industrial integration standard ISA-95, and subsequently used this ontology artefact as a configuration artefact to manage the distributed virtual enterprise material and equipment resources this is the main proposition of the thesis of utilizing semantic web ontology as resource configuration decision support.

Keywords: *Enterprise Integration, Networked Supply Chain, Information Sharing, Coordination, Ontology Based Approach, ISA-95, Ontology Web Language (OWL), Description Logic, Protégé, Java Expert System Shell (JESS).*

Table des matières

Remerciements	iii
Résumé	iv
Abstract	v
Table des matières	vi
Index des figures	ix
Index des tableaux	x
Chapitre 1 : Introduction	2
Chapitre 2 : État de l'Art	8
II.1 Introduction	8
II.2 Introduction à la Logique Descriptive	9
II.2.1 La base de connaissance de la Logique Descriptive	10
II.2.2 Modélisation de la base de connaissances et services de raisonnement	12
II.2.3 Algorithmes Tableaux	13
II.2.4 Technique d'optimisation des systèmes de Logique Descriptive	14
II.2.5 Système de Logique Descriptive	14
II.3 Opérations de gestion des ontologies	16
II.3.1 Gestion de l'évolution des ontologies	16
II.3.2 Résolution de l'hétérogénéité des ontologies	18
II.3.3 Raisonnement ontologique	19
II.4 Processus de développement d'ontologie	22
II.4.1 Développement d'ontologies collaboratives	23
II.5 L'ontologie, un media d'intégration dans l'entreprise collaborative	24
II.5.1 L'ontologie centrale comme media d'intégration de l'information	25
II.5.2 Ontologies pour la gestion du contenu	29
II.5.3 Les ontologies comme un media d'intégration pour les processus et produit de la chaîne logistique	38
II.6 Le model ISA-95	46
II.6.1 Classes représentant des ressources d'équipement	46
II.6.2 Classes représentant les entités de produit	47
II.6.3 Classes représentant les entités processus	48
II.6.4 Classes représentant les entités matériaux	50
II.6.5 Classes représentant les entités de la commande de production	51
II.7 Conclusion	52
Chapitre 3 : Modélisation	55
III.1 Introduction	55
III.2 Entreprise virtuelle et réseau logistique	56
III.3 Réseau logistique dans une perspective système	58

III.3.1	Réseau logistique comme un système configurable	59
III.3.2	Questions clés dans les systèmes configurables	60
III.3.3	Les Réseaux Logistiques configurable	61
III.3.4	Approche pour l'intégration des ressources distribuées du réseau logistique	62
III.3.5	Conclusion	64
III.4	Modélisation du réseau logistique	65
III.4.1	Considérations générales	65
III.4.2	Modèle	67
III.5	Le modèle référence du système pour la gestion des connaissances	67
III.5.1	Développement de composants pour les systèmes de gestion des connaissances	71
III.5.2	Conclusion	71
III.6	Approche ontologique pour la gestion de contenu	71
III.6.1	Architecture proposée pour la création et la gestion de contenu	72
III.6.2	Analyse des exigences pour l'intégration du contenu ontologique dans l'entreprise virtuelle	74
III.7	Modèle UML proposé	84
III.8	Conclusion	88
Chapitre 4 :	Application	90
IV.1	Introduction	90
IV.2	Présentation du cas d'étude	92
IV.3	Langage Ontologique Web (OWL)	95
IV.3.1	Constructeurs de langage OWL	95
IV.3.2	Langage de règles sémantique web (Semantic Web Rule Language – SWRL)	99
IV.4	Représentation ontologie d'ISA-95	99
IV.4.1	Introduction	99
IV.4.2	Nomenclature pour la représentation ontologie	100
IV.5	Application de la règle SWRL sur l'artefact ontologie modélisée	112
IV.6	Présentation et discussion des résultats	119
IV.7	Conclusion:	126
Chapitre 5 :	Conclusions et perspectives	129
V.1	Conclusion	129
V.2	Perspectives	133
Bibliographie		138
Annexes		148
Annexe - 1	Explication sur les services de raisonnement	148
Annexe - 2	Technique d'optimisation des algorithmes tableau par expansion des arbres.	149
Annexe - 3	Gestion de l'évolution des ontologies	150
A3.1	Correspondance d'ontologies	150
A3.2	Alignement, traduction et articulation des ontologies	151

A3.3	Modification/Édition d'ontologies	151
A3.4	Évolution des ontologies	152
A3.5	Débogage d'ontologies	152
A3.6	Versionnage des ontologies	153
A3.7	Fusion d'ontologies	153
Annexe - 4	Un Langage Ontologique pour Web sémantique basé sur la Logique Descriptive	155
Annexe - 5	Moteur de règles	157
A5.1	Mémoire de Travail	157
A5.2	Base de règles	158
A5.3	Moteur d'inférence	158

Index des figures

FIGURE 1 : ARCHITECTURE DU SYSTEME DE REPRESENTATION DES CONNAISSANCES BASE SUR LES LOGIQUES DE DESCRIPTION (BAADER F. <i>ET AL.</i> , 2010)	9
FIGURE 2 : COMPOSANTS PRINCIPAUX DU RAISONNEUR PELLET (SIRIN E. <i>ET AL.</i> , 2007)	20
FIGURE 3 : PROCEDURE DE FUSION DE L'ONTOLOGIE POUR L'INTEGRATION DE L'INFORMATION SUR LE CYCLE DE VIE DU PRODUIT	27
FIGURE 4 : LES TROIS TYPES D'ECART SEMANTIQUES (CHU H.C. <i>ET AL.</i> , 2009)	30
FIGURE 5 : ROLE DE L'ANNOTATION DANS LA GESTION DES CONNAISSANCES CENTREE SUR LE DOCUMENT (UREN V. <i>ET AL.</i> , 2006)	31
FIGURE 6 : PROCESSUS DE DEVELOPPEMENT DU DOCUMENT SEMANTIQUE A, B ET C (ERIKSSON H., 2007)	33
FIGURE 7 : CONFIGURATIONS STRUCTURELLES DE LA CHAINE LOGISTIQUE	40
FIGURE 8 : CADRE DE GESTION DES CONNAISSANCES BASEE SUR L'ONTOLOGIE DE L'ENTREPRISE COLLABORATIVE	41
FIGURE 9 : ALIGNEMENT DES PROCESSUS D'AFFAIRES	44
FIGURE 10 : HIERARCHIE DE L'EQUIPEMENT D'ENTREPRISE	46
FIGURE 11 : MODELE OBJET DE LA CLASSE D'EQUIPEMENT	47
FIGURE 12 : SPECIFICATIONS OBJET DE LA DEFINITION DE PRODUIT	48
FIGURE 13 : SPECIFICATIONS OBJET DU SEGMENT DE PROCESSUS	49
FIGURE 14 : MODELE MATERIAUX	50
FIGURE 15 : DIAGRAMME COMPOSANT DU PLANNING DE PRODUCTION	51
FIGURE 16 : MODELE OBJET GENERIQUE DE LA CHAINE LOGISTIQUE	59
FIGURE 17 : MODELE OBJET POUR L'ENTREPRISE VIRTUELLE	73
FIGURE 18 : DIAGRAMME D'ACTIVITE DU PROCESSUS POUR L'ONTOLOGIE DE DOMAINE MATURE	75
FIGURE 19 : ARCHITECTURE DU SERVEUR D'ONTOLOGIE	77
FIGURE 20 : INTERFACE D'ENTRE UNIQUE AVEC COMPOSANTS CORRESPONDANTS POUR INTEGRATION DE CONTENU ET D'ONTOLOGIE	78
FIGURE 21 : REPRESENTATION STATECHART DES FONCTIONNALITES DE BASE DE DOSM	79
FIGURE 22 : REPRESENTATION STATECHART DES FONCTIONNALITES DE BASE DE CARMIM	79
FIGURE 23 : REPRESENTATION STATECHART DES FONCTIONNALITES DE BASE DE SDMM	80
FIGURE 24 : PROCESSUS ET DIAGRAMME SEQUENCE D'INTERACTION DES ELEMENTS ARCHITECTURAUX POUR L'INTEGRATION DE CONTENU ET D'ONTOLOGIE	82
FIGURE 25 : ARCHITECTURE DE MISE EN ŒUVRE ET LIMITES TECHNOLOGIQUES	84
FIGURE 26 : MODELE UML BASE SUR ISA-95	85
FIGURE 27 : ENTITES STRUCTURELLES DE L'ETUDE DE CAS	94
FIGURE 28 : HIERARCHIE DES CLASSES D'ONTOLOGIE	103
FIGURE 29 : INSTANCIATION DE LA CLASSE ET PRODUCTION DES PRODUITS DE CLASSE	104
FIGURE 30 : INSTANCIATION DU CLASSE PROCESSUS, DE LA CLASSE MATERIELLE ET DE LA CLASSE EQUIPEMENT	106
FIGURE 31 : INSTANCIATION DES RESSOURCES MATERIELLES	107
FIGURE 32 : DOMAINE ET GAMME DES PROPRIETES DE L'OBJET ET PROPRIETES ASSOCIEES DE TYPE DE DONNEES	108
FIGURE 33 : APPROCHE D'INTEGRATION DES EQUIPEMENTS (KHAN W.A. <i>ET AL.</i> , 2011)	109
FIGURE 34 : ONTOLOGIE MODELISEE AVEC DES DONNEES D'INSTANCE	110
FIGURE 35 : RELATIONS LOGIQUES ENTRE LES EMPLACEMENTS DES INDIVIDUS	111
FIGURE 36 : DIAGRAMME DE SEQUENCE POUR L'APPLICATION DES REGLES SUR L'ARTEFACT ONTOLOGIQUE POUR L'INTEGRATION DES RESSOURCES	112
FIGURE 37 : DONNEES DE CONFIGURATION EXTRAITES DE L'ARTEFACT ONTOLOGIQUE POUR LA DEMANDE DE PRODUCTION	122
FIGURE 38 : UTILITE DE L'INFORMATION EXTRAITE PAR L'ARTEFACT D'ONTOLOGIE DANS LE RESEAU LOGISTIQUE	123
FIGURE 39 : ALGORITHME D'EXTRACTION DE L'INFORMATION DE CONFIGURATION DU RESEAU LOGISTIQUE	125
FIGURE 40 : RESULTATS DE L'ALGORITHME DE CONFIGURATION DU RESEAU LOGISTIQUE	125
FIGURE 41 : MODELE DE LA STRUCTURE D'UN RESEAU COMPLEXE AVEC SES CONTRAINTES DE CONFIGURATION	127
FIGURE 42 : REPRESENTATION DE L'ARTICULATION D'ONTOLOGIES ET DE LA FUSION D'ONTOLOGIES (FLOURIS G. <i>ET AL.</i> , 2008)	151
FIGURE 43 : COMPOSANTS DU MOTEUR DE REGLES	158
FIGURE 44: ALGORITHME RETE	160

Index des tableaux

TABLEAU 1 : REGLES D'EXTENSION POUR ALGORITHME TABLEAU DE $\mathcal{AL}$ (BAADER F. <i>ET AL.</i> , 2010)	149
TABLEAU 2 : REGLES D'EXPANSION POUR LE LAZY UNFOLDING DE L'ALGORITHME TABLEAU (BAADER F. <i>ET AL.</i> , 2010)	150
TABLEAU 3 : RESUME DES SOUS-ZONES DU CHANGEMENT D'ONTOLOGIE (FLOURIS G. <i>ET AL.</i> , 2008)	153
TABLEAU 4 : SYNTAXE ET SEMANTIQUE DE LA FAMILLE $\mathcal{S}$ DE LOGIQUE DESCRIPTIVE	156

Chapitre I
Introduction

Chapitre 1 : Introduction

Actuellement, la place de l'information dans l'évaluation et la gestion de la performance des chaînes logistiques, des réseaux distribués ou des entreprises virtuelles n'est plus à démontrer. En effet, l'échange et le partage d'information entre les différents acteurs, processus et ressources distribuées deviennent une nécessité pour permettre aux entreprises de rester concurrentielles. Ainsi, de nouvelles typologies de structuration des informations doivent être étudiées ainsi que les processus associés. L'intégration de ces informations partageables par tous ne peut se faire que par une formalisation des connaissances sous une forme contrôlable par des progiciels. Il est donc nécessaire de pouvoir tenir un raisonnement sur ces connaissances, de définir des règles facilitant la navigation dans celles-ci et de permettre une recherche efficace d'information utilisable et compréhensible par tous.

Ainsi, dans cette thèse, nous nous proposons d'utiliser une ontologie web sémantique comme un outil d'intégration destiné aux entreprises virtuelles composées de réseaux de ressources distribués. En effet, les entreprises virtuelles distribuées, en particulier celles composées de petites et moyennes entreprises, doivent constamment évoluer dans un milieu de plus en plus concurrentiel. L'intégration de leurs ressources tant matérielles qu'humaines au sein d'une chaîne logistique ne peut que conduire à une meilleure utilisation de celles-ci. En plus du cycle d'intégration, l'approche d'intégration doit être suffisamment souple pour répondre aux changements physiques qui peuvent survenir dans le réseau de ressources distribué. Les solutions traditionnelles de la gestion des ressources de l'entreprise au sein d'une chaîne logistique sont les ERP (Enterprise Resource Planning) et les MES (Manufacturing Execution System). Ces systèmes gèrent les opérations intra et inter entreprises lorsqu'elles collaborent, mais l'étude Delphi (Akkermans H.A. *et al*, 2003) identifie une faiblesse de ces systèmes au niveau de l'adaptation aux besoins futurs principalement pour ce qui est du point d'interaction. En effet, cette étude, conduite auprès d'industriels, a montré l'inutilité des systèmes ERP ou MES actuels quant à leur adaptation aux besoins d'un réseau logistique moderne. Il y a été identifié que ces systèmes ne sont pas seulement chers dans leur implémentation, mais qu'ils ne sont pas très bien adaptés aux changements structurels dans une configuration réseau. De fait, ils demandent un engagement significatif à tous les partenaires de la chaîne logistique.

Nous proposons donc une approche web sémantique dédiée à une telle situation dans laquelle les ressources de la chaîne logistique et du réseau de production sont distribuées. Cette approche sémantique répond à deux objectifs :

- Le premier concerne l'intégration informationnelle, sujet sur lequel bon nombre de recherches traitant d'approches sémantiques ont déjà été menées (cf. Chapitre II). Cependant, notre contribution se situe dans le contexte de l'intégration d'information pour une entreprise virtuelle ce qui n'a pas encore été abordé auparavant. En nous basant sur les besoins d'une entreprise distribuée, nous proposerons dans ce document une architecture ainsi que le processus qui permet de répondre au problème d'intégration de l'information au sein d'un réseau d'entreprises collaboratives. Ce processus permettant de prendre en compte autant les contenus existants que ceux qui seront créés durant la phase de collaboration.
- Le second objectif est de permettre l'intégration des matériels et équipements. Nous proposerons une ontologie sémantique web, basée sur un système expert, dédiée à cette intégration. Cela constitue en fait la principale proposition et contribution de nos travaux, une telle approche nous apparaissant comme novatrice et originale.

Pour atteindre le premier de ces objectifs, un état de l'art sur les outils du web sémantique a été mené, incluant les bases théoriques des technologies sémantiques, les algorithmes correspondants (concrétisant ces capacités théoriques) et la terminologie standard pour les opérations de gestion des ontologies. Cette dernière partie est exposée en raison du fait que les ontologies sont applicables à de multiples domaines de recherche et que parfois des inexactitudes sont observées dans la représentation de cette terminologie normalisée. De fait, nous avons pris soin de respecter au mieux ce standard et le montrons puisque les ontologies présentées plus loin dans ce manuscrit s'y rapportent entièrement. Après la gestion des opérations ontologiques, nous parlons de l'architecture de l'artefact d'ontologie et de son approche d'intégration au sein de l'entreprise virtuelle ayant des ressources distribuées. Différents domaines de recherche proposent différentes idées concernant la structure de l'artefact d'ontologie, par exemple, le modèle modulaire pour l'artefact ontologique, par opposition à un artefact ontologique singulier et central. Comme la nature distribuée de l'entreprise virtuelle nécessite une solution appropriée qui répond aux besoins distribués, il est nécessaire de présenter la capacité de raisonnement du *Ontology Web Language* (OWL) en parallèle à cette architecture ontologique. En effet, la capacité de raisonnement a un impact

direct sur la mise en œuvre du modèle architectural ontologique. À la suite de l'analyse des besoins architecturaux, les pratiques actuelles du processus de modélisation de l'artefact ontologique sont présentées. Elles sont nécessaires pour l'identification des processus et des éléments architecturaux pour la génération d'un artefact ontologique lorsque des entités collaborent au sein de l'entreprise virtuelle. Ce point sera traité dans le Chapitre III.

Pour atteindre le deuxième objectif de cette thèse, qui est l'intégration des ressources matérielles et des ressources équipement des entreprises, une revue de littérature est menée dans le domaine de l'utilisation des approches basées sur la connaissance pour la représentation des produits et des processus. De ces approches découlent des outils différents pour la mise en œuvre des systèmes des connaissances. Des approches basées sur les connaissances exigent un moteur de règles ayant des capacités d'inférence et qui peut s'appliquer sur l'artefact de connaissance. Le moteur d'inférence Java Expert System Shell (JESS) est souvent choisi en raison de sa compatibilité avec les technologies du web sémantique. Celles-ci permettent un pont entre l'ontologie OWL-DL modélisée, JESS et l'API Java correspondant qui peut gérer et contrôler l'artefact ontologique et le moteur d'inférence. Dans le cas d'une entreprise virtuelle, non seulement les processus normalisés de modélisation ontologique sont utiles afin de rationaliser le processus d'intégration, mais aussi les modèles de données correspondants qui répondent spécifiquement à l'intégration de l'entreprise. Cette thèse identifie également l'utilité d'un modèle standard pour l'intégration de l'entreprise virtuelle. Pour atteindre cet objectif, le modèle d'intégration mono-entreprise ISA-95 est utilisé. Celui-ci identifie les modèles des données matérielles, des ressources humaines et des équipements au sein d'une entreprise. Cette norme est ensuite appliquée à l'étude de cas après des modifications permettant de la faire passer d'un modèle mono-entreprise à un modèle multi-entreprises. La démarche proposée consiste : d'abord à générer un modèle équivalent à l'ISA-95, mais en UML pour formaliser la sémantique du modèle, puis à effectuer une conversion de ce modèle UML dans un équivalent artefact ontologique OWL-DL. Ensuite, à développer et mettre en œuvre un système fondé sur des règles qui intègre les ressources de l'entreprise. Ces règles sont basées sur la disponibilité des ressources et sur la distance entre les partenaires. Cette approche est utilisée en vue de rationaliser les mises en œuvre entre les partenaires qui collaborent ainsi que d'intégrer notre approche aux systèmes existants d'information d'entreprise. La littérature mentionne des exemples d'intégration d'artefacts ontologiques locaux dans un artefact singulier ontologique à travers un processus d'intégration ontologique et de leurs fusions ultérieures. Mais cette approche, dans le contexte de

l'entreprise virtuelle distribuée peut, selon notre opinion, causer des difficultés au cours de la phase de mise en œuvre et provoquer des retards éventuels.

De l'introduction ci-dessus, le travail de base et les approches de recherche correspondants sont clarifiés. Maintenant, nous allons présenter les différentes sections de cette thèse.

Le Chapitre II est la revue de la littérature qui porte sur les bases théoriques de l'ontologie web sémantique et de l'algorithme utilisé correspondant, sur le raisonnement ontologique, la gestion d'ontologies et le processus de modélisation ontologique. Il présente également les recherches existantes sur le pouvoir d'intégration des ontologies dans les artefacts informationnels, leurs utilités actuelles dans un contexte industriel et permet de donner les bases pour proposer notre solution d'utiliser l'ontologie comme un artefact d'intégration pour les ressources matérielles au sein des réseaux logistiques. Enfin, nous présentons succinctement la norme ISA-95.

Dans le Chapitre III, nous présentons les bases de l'approche d'utilisation d'ontologie comme un artefact de configuration. Pour ce faire, nous établirons le contexte de travail, plus précisément, en définissant le concept d'entreprise virtuelle et surtout en rappelant quelles en sont les spécificités et contraintes particulières. Nous aborderons ensuite ce concept d'entreprise virtuelle sous l'angle de la systémique et présenterons un modèle de ce système. Puis, en nous basant sur une analyse des principaux problèmes rencontrés dans les réseaux logistiques (re)configurables, nous proposerons une méthode en huit étapes permettant de créer une architecture web utile à la gestion de la configuration de la décision via un système d'aide à la décision.

Nous développerons, ensuite, certaines étapes clés de cette méthode. Nous détaillerons ainsi non seulement l'approche ontologique de gestion du contenu et l'architecture proposée correspondante, mais aussi les processus attachés et le modèle objet des différentes entités d'information impliquées lors de la mise en place d'une collaboration interentreprises. Enfin, nous terminerons par l'adaptation de la norme ISA-95, mono-entreprise, en un modèle UML multi-entreprises permettant de dériver un artefact ontologique basé sur des éléments standards.

Dans le Chapitre IV, nous présentons un cas d'étude basé sur une chaîne logistique volontairement simple, afin de pouvoir valider nos concepts. Nous présenterons donc l'ontologie obtenue, ses classes et leurs relations. Puis, nous verrons comment, à partir de

cette capacité d'intégration de l'information, nous pouvons adjoindre un système d'aide à la décision permettant la configuration de la chaîne logistique. En effet, nous verrons comment, via un moteur de règles et un système d'implémentation toujours résolument web et partagé, nous pouvons nous appuyer sur l'information contenue dans l'ontologie pour décider, au cas par cas, de la chaîne la plus adaptée à une commande particulière et préparer l'ordonnancement de ladite commande.

Enfin, dans le Chapitre V, nous conclurons sur ce travail et développerons quelques perspectives de travaux futurs.

Chapitre II

État de l'art

Chapitre 2 : État de l'Art

II.1 Introduction

Dans ce chapitre, une revue de la littérature concernant l'utilité de l'ontologie comme un artefact d'intégration est fournie. Notre intention d'appliquer un système d'intégration basé sur la logique est due à la contrainte de temps ainsi qu'à la contrainte du partage de l'information au sein de l'entreprise virtuelle. Comme le système d'intégration classique nécessite un engagement contractuel de longue durée entre les partenaires de la collaboration, cet engagement limite l'utilisation des ressources. Notre objectif est de pouvoir utiliser ces ressources de manière efficace et flexible. Pour réaliser cette intégration, une approche basée sur la représentation logique des ressources est proposée. Les principales caractéristiques de cette approche d'intégration sont sa rapidité et sa facilité de mise en œuvre. Les systèmes logiques à la base ont trois axes fondamentaux de recherche. Ces axes sont le processus pour la modélisation de l'artefact logique dans un environnement distribué, la structure de l'artefact logique, c'est-à-dire soit un artefact central ou plusieurs artefacts modulaires, et les capacités de raisonnement et d'inférence possibles sur l'artefact logique.

Dans ce chapitre, l'état de l'art relatif à nos recherches sur l'intégration des ressources dans l'entreprise basée sur les ontologies et utilisant un système de connaissance augmenté de règles est présenté. Nos travaux utilisant un langage d'ontologies web sémantique (OWL-DL) comme outil de représentation logique, nous nous intéressons tout d'abord aux articles portant sur les « logiques descriptives » qui constituent la base indispensable à la compréhension du langage ontologique web. Nous aborderons, ensuite, différentes opérations de gestion du changement des ontologies ainsi que les différents concepts ontologiques, lesquels ne sont pas obligatoirement explicitement abordés dans la littérature. Puis nous discuterons du processus nécessaire à la création et à la modélisation des ontologies. En conséquence, trois contextes distincts d'utilisation des ontologies seront identifiés : les médiums d'intégration pour le contenu, les processus industriels et les informations produits.

Les systèmes de représentation de la connaissance basés sur la Logique Descriptive (DL) fournissent à leurs utilisateurs plusieurs possibilités de traitement afin de déduire une connaissance implicite à partir d'une représentation explicite de la connaissance (Baader F. and Sattler U., 2001). L'origine des systèmes DL se trouve dans KL-ONE (Brachman R.J. and Schmolze J.G., 1985) qui marquait la transition des réseaux sémantiques vers une logique

terminologique bien mieux fondée. KL-ONE fut le premier à introduire les notions de rôles et de concepts ainsi que leurs interrelations dans la définition des concepts avec, par exemple, la restriction de valeur et la restriction de numéro sur les rôles logiques dans l'artefact ontologique pour définir un nouveau concept. Les composants architecturaux du système KL-ONE, Figure 1, permettent les services d'inférences de classification et de subsumption¹ ainsi que, par conséquence, la distinction entre les T-BOX (Boite Terminologique) qui identifient la terminologie des concepts représentés par l'artefact de connaissance et les A-BOX (Boite d'Assertion/factuelle) qui relient l'individu aux concepts modélisés avec l'artefact de connaissance. L'objectif de la génération actuelle des systèmes de représentation de la connaissance basés sur la Logique Descriptive est la complétude² du raisonnement ainsi que l'expressivité du langage. Les systèmes récents qui permettent une implémentation efficace du raisonnement et du langage DL sont : FacT (Horrocks I., 1998) et RACER (Haarslev V. and Müller R., 2001).

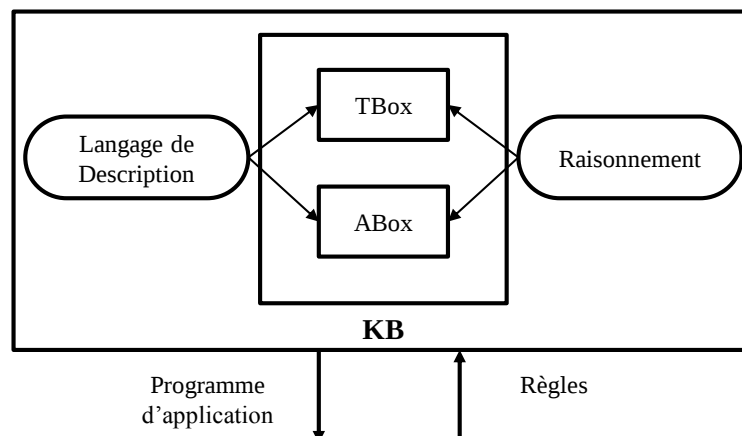


Figure 1 : Architecture du système de représentation des connaissances basé sur les logiques de description (Baader F. et al., 2010)

II.2 Introduction à la Logique Descriptive

La Logique Descriptive est le plus récent des formalismes de représentation de la connaissance qui permet, premièrement, de définir les concepts correspondants à un domaine particulier, puis, deuxièmement, d'utiliser ces concepts pour spécifier les propriétés d'objets ou d'individus liés à ce domaine. Ce processus est appelé description du monde. La Logique Descriptive a une logique formelle basée sur la sémantique avec une mise en évidence

¹ La subsumption désigne une relation hiérarchique entre des concepts, dans les logiques de description. Cette notion est proche de la relation « est impliqué par » en logique classique, ou encore « contient » en logique ensembliste. La relation de subsumption permet de construire un treillis de Galois à partir d'un ensemble d'individus et de propriétés.

² Caractère d'un système logique qui ne contient pas de propositions indécidables.

particulière des services de raisonnement qui permettent une inférence de la connaissance implicite à partir de bases de connaissance explicites (Baader F. *et al.*, 2010).

Comme la Logique Descriptive est un formalisme de représentation de la connaissance, des services tels que la réponse aux requêtes, la décidabilité du raisonnement et la procédure de vérification de la complétude de la base de connaissances sont importants pour une utilisation pratique. Ces procédures ne sont pas explicitement discutées dans le formalisme de la logique du premier ordre. Les procédures décisionnelles, pour la décidabilité du raisonnement et pour la complétude de la base de connaissances, sont directement liées à l'expressivité du langage de représentation de connaissance. C'est-à-dire un langage logique avec une expressivité de logique supérieure qui exige des procédures décisionnelles plus complexes pour la vérification de la décidabilité du raisonnement et la complétude de la base de connaissance. Cependant, l'expressivité logique de la langue augmente les résultats dans des algorithmes de raisonnement plus complexes qui mettent la restriction sur l'utilisation de la mémoire ainsi que sur le temps de calcul. Ces idées ont d'abord été présentées par Brachmen (Brachman R.J and Schmolze J.G., 1985) dans son travail sur les structures de réseau logique, qui ont orientées le développement ultérieur vers les logiques descriptives :

- Les blocs syntaxiques de base sont les concepts atomiques (prédicats unaires), les rôles atomiques (prédicats binaires) et les individus (constantes).
- La puissance expressive du langage logique est restreinte, en particulier, car il utilise des constructions limitées pour représenter des rôles et des concepts complexes.
- La connaissance implicite au sujet de concepts ou d'individus peut être déduite automatiquement avec l'aide des procédures d'inférence, en particulier l'identification des hiérarchies des concepts et des rôles dans un réseau.

II.2.1 La base de connaissance de la Logique Descriptive

La base de connaissances de la Logique Descriptive se compose de deux parties : la TBox ($\mathcal{T}$) et l'ABox ($\mathcal{A}$). La TBox identifie la terminologie des concepts au sein du domaine d'application, alors que l'ABox identifie les individus se trouvant sur les concepts du domaine. Tous les systèmes de Logique Descriptive utilisent des constructeurs pour identifier les concepts et rôles atomiques ou complexes.

La sémantique formelle de $\mathcal{AL}^3$ -concepts est une interprétation $\mathcal{I}$ d'un ensemble non-vide $\Delta^{\mathcal{I}}$. Une fonction d'interprétation assigne à un concept atomique A , un ensemble $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ et chaque rôle atomique R est une relation binaire $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$

Le langage $\mathcal{AL}$ peut être étendu via des capacités additionnelles telles que l'union de concepts $[U]$, la qualification existentielle $[\mathcal{E}]$, le nombre de restrictions $[\mathcal{N}]$ ou la négation de concept $[C]$ et ces capacités peuvent être représentées en utilisant le méta langage $\mathcal{AL} [U] [\mathcal{E}] [\mathcal{N}] [C]$. Pour plus d'explications sur ses capacités voir section 2.5 de ce chapitre.

La définition de concepts complexes avec des concepts atomiques dans une TBox ($\mathcal{T}$) est réalisée grâce à des axiomes terminologiques. Ceux-ci utilisent les axiomes d'inclusion et d'égalité pour la définition de concepts complexes. Par exemple : Homme $\equiv$ Personne $\sqcap$ Male.

En plus de la TBox, une base de connaissance utilise aussi les assertions par rapport à la description du monde, à travers un ensemble fini d'assertions nommé ABox. Celle-ci introduit les individus en leur donnant un nom unique et en réalisant les assertions avec leurs propriétés individuelles. Ainsi, les individus a , b et c sont liés au concept C par l'assertion de concept $C(a)$ alors que l'assertion de rôle (b, c) indique une relation entre les concepts individus b et c . En simplifiant, l'ABox peut être vue comme une instance de la base de données relationnelle avec des relations unaires et binaires et constitue le niveau factuel, alors que la TBox impose des relations sémantiques entre les concepts et les rôles de l'Abox et constitue donc le niveau conceptuel. À l'opposé de la théorie classique de base de données, la sémantique de l'ABox reprend la théorie du « monde ouvert » puisque la base de connaissances est mise en œuvre dans une situation considérée comme incomplètement connue et que l'absence d'information est synonyme d'ignorance et non plus d'information négative.

La fonction d'interprétation $\mathcal{I}$ de la base de connaissances $\mathcal{I} = (\Delta^{\mathcal{I}} \times \cdot^{\mathcal{I}})$ établie une correspondance non seulement entre les rôles et les concepts atomiques et un ensemble de relations, mais apparie en plus un individu à un élément $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$. Cette interprétation $\mathcal{I}$ remplit l'assertion de concept $C(a)$ si $a^{\mathcal{I}} \in C^{\mathcal{I}}$ et elle remplit l'assertion de rôle $R(a, b)$ si $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$. Une interprétation $\mathcal{I}$ est un *modèle* de la base de connaissance lorsqu'elle satisfait à la fois $\mathcal{A}$ et $\mathcal{T}$, fournissant ainsi une interprétation du monde dans laquelle les concepts sont interprétés

³ $\mathcal{AL}$ - de langue attributive a été introduite par (Schmidt-Schauss and Smolka, 1991) comme un formalisme minimal de représentation des connaissances qui est d'un intérêt pratique. Les logiques de description sont une extension de $\mathcal{AL}$.

comme des sous-ensembles de domaines (tels que représentés dans $\mathcal{T}$) et l'appartenance des individus à ces concepts ainsi que leurs relations les uns avec les autres par des assertions dans $\mathcal{A}$ (Baader F. *et al.*, 2010).

II.2.2 Modélisation de la base de connaissances et services de raisonnement

Modéliser un nouveau domaine dans la base de connaissances demande la définition des terminologies de concept ainsi que leurs relations dans $\mathcal{T}$. Pour vérifier la cohérence de la base de connaissances, des tâches de raisonnements logiques, permettant de déterminer la validité des concepts définis, sont nécessaires. Voici quelques tâches de raisonnement, utiles pour la maintenance de la consistance du modèle de base de connaissances (Baader F. *et al.*, 2010) :

1. **Satisfiabilité** : un concept C est satisfiable par rapport à $\mathcal{T}$ s'il existe un *modèle* $\mathcal{I}$ de $\mathcal{T}$ tel que $C^{\mathcal{I}}$ est non-vide. Dans ce cas, nous pouvons dire que $\mathcal{I}$ est un *modèle* de C .
2. **Subsomption** : un concept C est subsumé par un concept D par rapport à $\mathcal{T}$ si $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ pour chaque *modèle* $\mathcal{I}$ de $\mathcal{T}$. Dans ce cas, nous écrivons $C \sqsubseteq_{\mathcal{T}} D$ ou $\mathcal{T} \models C \sqsubseteq D$.
3. **Équivalence** : deux concepts C et D sont équivalents par rapport à $\mathcal{T}$ si $C^{\mathcal{I}} = D^{\mathcal{I}}$ pour chaque *modèle* $\mathcal{I}$ de $\mathcal{T}$. Dans ce cas, nous écrivons $C \equiv_{\mathcal{T}} D$ ou $\mathcal{T} \models C \equiv D$.
4. **Disjonction** : deux concepts C et D sont disjoints par rapport à $\mathcal{T}$ si $C^{\mathcal{I}} \cap D^{\mathcal{I}} = \emptyset$ pour chaque *modèle* $\mathcal{I}$ de $\mathcal{T}$.

Une implémentation d'un système de langage descriptif dispose généralement de la subsomption comme mécanisme de raisonnement. En effet, celui-ci est suffisant pour implémenter les autres inférences. On peut trouver en Annexe - 1, quatre propositions explicitant ces services de raisonnement.

Nous pouvons noter qu'une procédure de décision nécessite des algorithmes capables de décider de la satisfiabilité des concepts et apporte la conjonction aussi bien que la négation de concepts arbitraires.

Les systèmes récents de Logique Descriptive, tels que KRIS (Baader F. and Hollunder B., 1991) ou Fact (Patel-Schneider P. and Horrocks I., 1999) sont basés sur des algorithmes de vérification de la satisfiabilité nommés « Algorithme Tableaux » (TA) ainsi que sur les techniques correspondantes pour l'optimisation des TA pour différents langages $\mathcal{AL}$.

II.2.3 Algorithmes Tableaux

Les algorithmes tableaux prouvent la satisfiabilité d'un concept C en construisant soit une interprétation $\mathcal{I}$, soit un *modèle* dans le lequel $C^{\mathcal{I}}$ est non-vide. Un concept est satisfiable si aucune des extensions possibles de l'interprétation $\mathcal{I}$ ne donne une structure contradictoire.

L'algorithme s'initialise avec un individu satisfaisant le concept D et construit un tableau à travers l'inférence de l'existence d'un individu supplémentaire ou de contraintes additionnelles sur les individus. Ce mécanisme d'inférence consiste à appliquer un ensemble de règles d'extension qui correspondent aux constructeurs logiques du langage. L'algorithme se termine soit lorsque la structure est complète (plus d'autres inférences possibles), soit lorsqu'une contradiction évidente a été trouvée.

D'un point de vue théorie des graphes, les algorithmes tableau construisent un graphe direct dans lequel chaque nœud x est labélisé avec un ensemble de concepts ($\mathcal{L}(x) = \{C_1, \dots, C_n\}$) et chaque arc $\langle x, y \rangle$ est labélisé avec des rôles ($\mathcal{L}(\langle x, y \rangle) = R$) (Baader F. and Sattler U., 2001). Quant un concept C est associé à un nœud x ($C \in \mathcal{L}(x)$), il représente un *modèle* dans lequel l'individu x est une représentation de C . Quant un arc $\langle x, y \rangle$ est associé à un rôle R ($\mathcal{L}(\langle x, y \rangle) = R$), il représente un *modèle* dans lequel le couple correspondant $\langle x, y \rangle$ est une interprétation de R . Une contradiction (ou clash) est détectée lorsque $\{C, \neg C\} \subseteq \mathcal{L}(x)$ pour un nœud x .

Pour tester la satisfiabilité d'un concept D , l'algorithme de base initialise un arbre qui contient un seul nœud x (nommé nœud racine) avec $\mathcal{L}(x) = \{D\}$ puis étend cet arbre en appliquant des règles soit pour étendre les associations à chaque nœud, soit pour ajouter de nouveaux nœuds feuilles. Nous pouvons noter que les concepts sont supposés être dans la forme négative normale (FNN) c'est-à-dire que le « $\neg$ » n'apparaît que et uniquement devant le nom du concept. Arbitrairement, les concepts de $\mathcal{ALC}$ peuvent être convertis en FNN en utilisant une combinaison des lois de De Morgan⁴ et les équivalences $\neg(\exists R.C) \Leftrightarrow (\forall. \neg C)$ et $\neg(\forall R.C) \Leftrightarrow (\exists R. \neg C)$. Cette procédure peut être étendue à une logique plus expressive en utilisant des équivalences additionnelles telles que : $\neg(\leq nR) \Leftrightarrow (\geq (n+1) R)$.

⁴ La négation de la conjonction de deux propositions est équivalente à la disjonction des négations des deux propositions, ce qui signifie non(A et B) est (non A) ou (non B). La négation de la disjonction de deux propositions est équivalente à la conjonction des négations des deux propositions, ce qui signifie que non(A ou B) est (non A) et (non B)

Les concepts disjonctifs $(C \sqcup D) \in \mathcal{L}(x)$ entraînent une expansion non déterministe. Ces problèmes de raisonnement pratique peuvent être améliorés par des techniques d'optimisation et seront abordés ultérieurement (voir section II.2.4). De même, les constructeurs logiques plus expressifs peuvent guider l'expansion non déterministe.

La restriction existentielle de rôle sur les concepts $\exists R. C \in \mathcal{L}(x)$ provoque la création de nouveaux nœuds R -successeurs et la restriction universelle de rôle sur les concepts $\forall R. C \in \mathcal{L}(x)$ étend les associations des nœuds R -successeurs. Ces restrictions sont déjà discutées dans l'introduction initiale de la Logique Descriptive dans la section II.2.1.

Une des techniques théoriques d'optimisation pour l'expansion de l'arbre est la technique de trace qui impose des limitations sur l'application des règles d'expansion. Une autre technique qui aide à l'expansion de l'arbre est le « lazy unfolding » qui signifie qu'un concept n'est déployé que s'il correspond à un nœud. Un ensemble de règles d'extension pour la Logique Descriptive $\mathcal{ALC}$ (Langage Attributif pour les Concepts) est donné en Annexe - 2.

II.2.4 Technique d'optimisation des systèmes de Logique Descriptive

Les systèmes de Logique Descriptive utilisent différentes techniques d'optimisation pour améliorer la complexité des cas typiques. Quelques-unes des techniques les plus courantes sont définies ci-dessous et classifiées suivant (Baader F. *et al.*, 2010) :

1. Optimisations de prétraitement qui modifient la base de connaissances pour optimiser la classification et les tests de subsomption ;
2. Optimisations de classement partiel qui essaient de minimiser le nombre de tests de subsomption pour classer une base de connaissance ;
3. Optimisations de subsomption qui tentent d'éviter des tests de satisfiabilité potentiellement coûteux, généralement en leur substituant un test moins coûteux ;
4. Optimisations de satisfiabilité qui essaient d'augmenter la performance des cas typiques du testeur de satisfiabilité.

II.2.5 Système de Logique Descriptive

Dans les parties précédentes, les algorithmes de base impliqués dans le fonctionnement des bases de connaissances de la Logique Descriptive sont présentés de même que les références correspondantes. Le Langage Attributif ($\mathcal{AL}$) a été introduit par (Schmidt-Schauß M. and

Smolka G., 1991) comme le formalisme minimal nécessaire à la représentation de la connaissance pour une utilisation pratique. La Logique Descriptive est une extension de $\mathcal{AL}$ pour $\mathcal{C}$ (concepts) avec des capacités additionnelles. Ainsi, la lettre $\mathcal{N}$ représente la capacité de restriction à des nombres simples et la lettre $\mathcal{Q}$ est utilisée pour qualifier la restriction à des nombres. $\mathcal{H}$ est utilisée pour la hiérarchie des rôles avec parents multiples, la conjonction des rôles est notée avec $\mathcal{R}$. L'index R^+ est utilisé pour représenter le support des rôles transitifs. Dans le cas où des rôles inverses sont inclus dans le système DL, cela est indiqué soit par l'indice $(^{-1})$ ou bien la lettre $\mathcal{J}$. Les services de raisonnement de base de la base de connaissance DL sont rappelés ci dessous.

Si le système permet un raisonnement sur l'ABox, les inférences suivantes sont proposées (Franz Baader *et al.*, 2010) :

1. Vérifier la consistance de l'ABox ($\mathcal{A}$) par rapport à la TBox ($\mathcal{T}$).
2. Vérifier les instances présentées dans la base de connaissance par rapport aux assertions de la TBox ($\mathcal{T}$) et de l'ABox ($\mathcal{A}$).
3. Identifier l'ensemble des rôles présents entre deux individus i et j par rapport à la connaissance ($\mathcal{T}, \mathcal{A}$). Représenté mathématiquement comme $\{R \mid (\mathcal{T}, \mathcal{A}) \models (i, j) : R\}$.
4. Identifier les instances d'individus avec les noms de concepts les plus spécifiques mentionnés dans la TBox ($\mathcal{T}$) et les assertions dans l'ABox ($\mathcal{A}$).
5. Identifier tous les individus cités dans l'ABox ($\mathcal{A}$) qui sont les instances d'un concept $\mathcal{C}$ donné dans la TBox ($\mathcal{T}$), appelé le problème de l'inférence de récupération.
6. Identifier l'ensemble des charges de rôle R pour un individu i à l'égard de la TBox ($\mathcal{T}$) et de l'ABox ($\mathcal{A}$) représentée mathématiquement comme $\{x \mid (\mathcal{T}, \mathcal{A}) \models (i, j) : R\}$.

Les services de raisonnement de la TBox sont les suivants :

1. Vérification de la cohérence des concepts identifiés dans la TBox ($\mathcal{T}$).
2. Vérification du concept de subsumption à l'égard de TBox ($\mathcal{T}$).
3. Vérification de la cohérence de la TBox ($\mathcal{T}$) en vérifiant la cohérence de tous les concepts mentionnés dans la TBox ($\mathcal{T}$) sans calculer les relations avec les concepts descendants / ascendants.
4. Identification de la hiérarchie des noms de concepts au sein d'une TBox ($\mathcal{T}$), ce service est appelé classement TBox.

II.3 Opérations de gestion des ontologies

Les évolutions d'ontologie englobent plusieurs champs de recherche qui sont liés les uns aux autres. La plus grande partie de la littérature sur les évolutions des ontologies aborde des problèmes en relation avec au moins deux champs de recherches différents. Dans le contexte de l'utilité d'une ontologie, la supposition prédominante est que l'ontologie est un objet unique et central sur lequel des algorithmes de raisonnement, une conduite du changement et des opérations d'intégrations de sources de données peuvent être appliqués. L'état de l'art aborde aussi l'ontologie comme une entité modulaire transitoire. Ainsi, des ontologies distribuées peuvent localement représenter des concepts de domaine, et/ou des appels à des algorithmes de raisonnement et à des techniques de correspondance qui permettent de s'adapter aux besoins particuliers de ces ontologies modulaires distribuées. Le thème central de cette section est d'identifier la classification des opérations de conduite du changement des ontologies ainsi que l'état actuel de l'état de l'art sur les capacités de raisonnement de la Logique Descriptive. Dans cette revue de la littérature, les dernières techniques qui abordent le problème de correspondance des ontologies distribuées et les techniques de raisonnement seront donc présentées, permettant ainsi une compréhension accrue des limitations techniques de l'extension d'un artéfact de connaissance simple à un artefact modulaire et distribué.

II.3.1 Gestion de l'évolution des ontologies

Le besoin de gestion de l'évolution peut émerger de différents scénarios tels que, par exemple, la découverte d'une nouvelle information (qui peut être une nouvelle instance de donnée, une autre ontologie, une nouvelle observation, etc.), la modification dans le focus ou le point de vue d'une conceptualisation, une information reçue de sources extérieures, un domaine dynamique résultant du changement d'une conceptualisation ontologique, la découverte de schémas problématiques dans le processus de modélisation, la communication entre des sources d'informations hétérogènes ou encore la fusion d'informations issues de différentes ontologies (Flouris G. *et al.*, 2008).

Il apparaît aussi dans la littérature sur l'utilisation des ontologies que les inférences ou les services de raisonnement (comme l'identification de la subsomption de concepts, par exemple) sont appliqués à une ontologie centrale. En effet, les services de raisonnement distribués ne sont pas encore assez matures pour assurer la complétude du raisonnement ou conserver une complexité computationnelle basse.

En plus de ces scénarios et de cette limitation computationnelle, si l'on considère les caractéristiques de l'environnement d'application des ontologies considérées ici (Web sémantique), à savoir : dynamique, distribuée et multi-acteurs, il devient évident que les ontologies ne peuvent être stables et qu'en effectuer un modèle efficace ainsi que prendre en compte leurs évolutions sont des nécessités.

La gestion de l'évolution des ontologies est classée par (Flouris G. *et al.*, 2008) en dix sous domaines :

- **Correspondance** : fait référence à la tâche qui consiste à relier le vocabulaire de deux ontologies qui partagent le même domaine (Kalfoglou Y. and Schorlemmer M., 2003) (Choi N. *et al.*, 2006) (Calvanese D. *et al.*, 2002).
- **Alignement, articulation, traduction** : l'alignement des ontologies est une tâche qui consiste à établir une collection de relations binaires entre le vocabulaire de deux ontologies.
- **Évolution** : l'évolution de l'ontologie peut être considérée comme de l'édition d'ontologies (Flouris G. *et al.*, 2008) et comme un processus d'adaptation des ontologies pour absorber les modifications dans les domaines correspondant, tout en maintenant à la fois la consistance de l'ontologie elle-même, mais aussi des artefacts correspondants (Plessers P. *et al.*, 2007). L'évolution dans l'artefact ontologique est directement liée aux modifications temporelles dans les concepts. Il est important d'aborder la notion de différence entre l'évolution d'ontologie et le problème de divergence d'impédance. L'évolution des ontologies prend en considération l'évolution des concepts ontologique au cours du temps, alors que la divergence d'impédance concerne les problèmes de correspondance entre les concepts ontologiques et les sources de données. Ces deux éléments sont importants dans la gestion des ressources d'information distribués. Deux approches existent pour gérer l'évolution des artefacts ontologiques particulièrement des ontologies des logiques descriptives en termes d'historique : « Timestamp » et « Snapshot ». La technique de « timestamp » intègre l'étiquetage du temps, alors que la technique « snapshot » maintient l'état des ontologies à différents intervalles de temps. Ces deux mécanismes ont un certain désavantage dans la détection des modifications dans les ontologies. L'approche conventionnelle dans les applications pour gérer l'historique de modification des ontologies est dans la diffusion des changements importants dans la

représentation des concepts, dans les relations logiques entre concepts et dans les propriétés des types de données ou d'objet des concepts.

(Klein M. *et al.*, 2004) fournissent les mécanismes qui assurent la compréhension de plus haut niveau élémentaire et des opérations de changements composites⁵. On peut aussi noter la nécessité de disposer d'un mécanisme d'identification des conséquences liées à une opération de modification (Plessers P. and De Troyer O., 2005).

- **Débogage** : qui est étroitement lié aux évolutions d'ontologies. Ce processus de débogage, par contre, corrige les inconsistances logiques survenant dans des ontologies statiques, ainsi le contexte dans lequel ces inconsistances sont résolues est déterminant pour différencier débogage et évolution d'ontologies (Meyer T. *et al.*, 2005).
- **Versionnage** : dans le cas de l'évolution des ontologies, on s'intéresse juste à la validation d'une nouvelle version d'ontologie, alors que le versionnage comprend en plus la gestion des problèmes de validité, d'interopérabilité et de gestion par rapport à toutes les versions précédentes (Noy N.F. *et al.*, 2004).
- **Intégration et fusionnement** : cette catégorie fournit une agrégation de deux opérations ontologiques : l'intégration et l'assemblage. L'objectif de la fusion d'ontologie est donc l'obtention d'une nouvelle ontologie à partir de deux ou plus ontologies sources liées au même sujet. Les ontologies sources sont donc remplacées et unifiées dans l'ontologie résultat de la fusion. Pour parvenir à ce résultat, les approches les plus utilisées sont l'union ou l'intersection.

II.3.2 Résolution de l'hétérogénéité des ontologies

La résolution de l'hétérogénéité se rapporte à la résolution des différences dans la terminologie ainsi que dans le langage de représentation ou la syntaxe. L'approche usuelle pour une telle résolution est de définir les règles de traduction qui annulent l'hétérogénéité dans le langage de représentation.

Les correspondances, transformations, alignements, articulations et traductions d'ontologies sont des champs de recherche qui découlent de la résolution de l'hétérogénéité.

⁵ L'opération de modification est considérée comme composite, lorsque cette opération de modification englobe plusieurs opérations élémentaires du changement.

II.3.3 Raisonnement ontologique

Dans la section précédente, nous avons abordé les besoins opérationnels concernant la gestion de l'évolution des ontologies, les opérations associées ainsi que la classification des évolutions d'ontologies. Nous allons nous intéresser maintenant à l'identification des besoins en raisonnement associés aux ontologies ainsi qu'à l'état actuel de la recherche dans le domaine et ses limitations.

La Logique Descriptive basée sur les services de raisonnement ontologique a déjà été introduite plus haut dans ce chapitre, mais dans ce paragraphe, nous allons nous focaliser plus particulièrement sur les besoins de raisonnements des ontologies modulaires distribuées. Dans un premier temps, nous présenterons les capacités des raisonneurs des algorithmes tableaux, puis les spécificités des raisonneurs dans les ontologies modulaires distribuées et enfin nous aborderons l'utilité des techniques de raisonnement.

Il est donc impératif que les raisonneurs pratiques fournissent au moins cet ensemble standard de services d'inférences de la Logique Descriptive (Sirin E. *et al.*, 2007) :

- **Vérification de la consistance** : s'assure que l'ontologie ne contient aucun fait contradictoire ;
- **Satisfiabilité des concepts** : vérifie s'il est possible pour chaque classe d'avoir une instance ;
- **Classification** : calcule les relations de sous-classe entre chaque classe nommée pour créer la hiérarchie de classe complète. Cette hiérarchie peut être utilisée pour répondre aux requêtes telles que « trouver toutes les sous-classes d'une classe ou les sous-classes directes » ;
- **Réalisation** : trouve la classe la plus spécifique dans laquelle un individu a sa place. En d'autres termes, calcule le type direct pour chaque individu. La Réalisation peut seulement être effectuée après une Classification puisque les types directs sont définis en fonction de la hiérarchie de classe. En utilisant une hiérarchie de Classification, il est possible d'obtenir tous les types pour les individus.

Pellet (Sirin E. *et al.*, 2007) et Fact (Horrocks I., 1998) utilisent les algorithmes tableaux pour vérifier ces services d'inférences.

II.3.3.1 Le raisonneur Pellet

Le système central du raisonnement Pellet (Figure 2) est l'algorithme tableau qui vérifie la consistance de la base de connaissances. Les ontologies sont chargées dans un raisonneur après la validation d'espèce et la réparation d'ontologie ce qui permet de s'assurer que toutes les ressources ont un type approprié pour le raisonneur. C'est pendant cette phase de chargement que les axiomes sur les classes sont mis dans la TBox et que les assertions sur les individus sont stockées dans l'ABox. Les axiomes de la TBox passent alors à travers le pré processus standard du raisonneur DL. Le système fournit en effet une couche pour les accès programmatiques à travers l'interface de programmation (Sirin E. *et al.*, 2007). Pellet permet aussi les ontologies modulaires en fournissant un service, de vérification de la consistance, dédié aux ontologies $\mathcal{E}$ Connexion auxquelles nous allons nous intéresser maintenant.

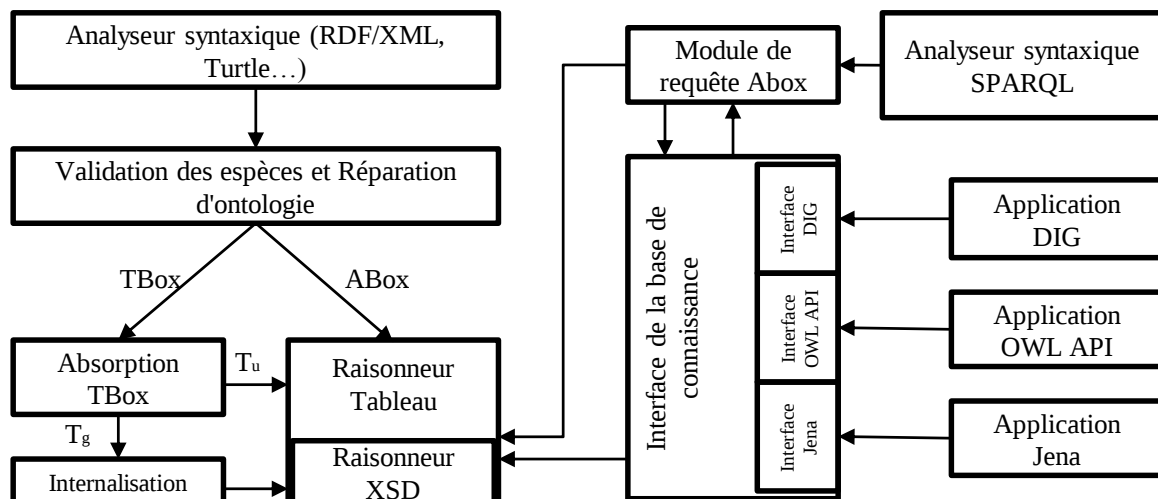


Figure 2 : Composants principaux du raisonneur Pellet (Sirin E. *et al.*, 2007)

II.3.3.2 Raisonnement des ontologies modulaires

Au fur et à mesure que la taille des ontologies a augmenté, elles sont devenues de plus en plus difficiles d'une part à traiter pour les raisonneurs et à être réutilisées d'autre part. De façon à rendre la réutilisation et le partage plus facile, les ontologies doivent être conçues comme fortement indépendantes et modulaires. L'ontologie $\mathcal{E}$ Connexion (Grau B.C. *et al.*, 2006) est une approche visant à la modularité et à la réutilisabilité et contient un ensemble d'ontologies connectées. Ce type d'ontologies contient habituellement des informations sur les propriétés de classes et leurs instances, mais aussi sur une nouvelle sorte de propriété appelée : propriété lien. Cette nouvelle propriété est similaire aux propriétés de type de données. Elle permet la

création de classe dans une ontologie basée sur des informations issues de différentes ontologies, à condition que les domaines des ontologies soient disjoints.

Le principal intérêt des $\mathcal{E}$ Connexions est de combiner des logiques décidables de telle manière que le formalisme résultant reste décidable, même si, dans le pire des cas, l'augmentation de l'expressivité peut amener à une plus grande complexité. La propriété lien est interprétée d'un point de vue logique comme une relation binaire, où le premier élément se rapporte à l'ontologie « source » et le second à la « cible ». Une ontologie $\mathcal{E}$ Connexion peut être décrite comme une ontologie étendue par l'habilité de définir la propriété lien et de construire de nouvelles classes en termes de restriction sur elles. Notons que la propriété lien ne peut être marquée comme transitive ou symétrique. En effet, puisque cette propriété est définie comme étant un lien entre une source unique et une cible unique, la transitivité ou la symétrie obligerait de pouvoir la définir avec plusieurs sources et/ou cibles et donc étendre le Framework $\mathcal{E}$ Connexion de façon non triviale.

De plus, le Framework $\mathcal{E}$ Connexion impose quelques restrictions sur les axiomes et les faits qui, par exemple, limitent les possibilités de déclarations croisées. L'utilisation de $\mathcal{E}$ Connexion tant pour l'intégration que pour la décomposition de la base de connaissances OWL suppose donc une nouvelle méthode de modélisation applicable à la connaissance traitée par la Logique Descriptive en général et par le web sémantique en particulier. L'idée centrale est donc de garder les ontologies petites et disjointes et d'utiliser ces ontologies comme des unités réutilisables qui peuvent être combinées de diverses façons via $\mathcal{E}$ Connexion en fonction des besoins des concepteurs.

Intuitivement, la propriété lien relie des individus correspondant à différentes interprétations de domaines. De fait, la propriété lien agit comme une passerelle entre des ontologies différentes qui restent pourtant séparées et gardent leur propre identité.

Pour ce qui est des besoins de raisonnements pour les ontologies modulaires $\mathcal{E}$ Connexion, les algorithmes tableau suffisent. La stratégie est alors d'étendre les algorithmes tableaux DL avec la prise en compte de $\mathcal{E}$ Connexion en se basant sur les graphes colorés. Les nœuds de différentes « couleurs » correspondent alors à différents domaines (ontologies). Du point de vue des algorithmes tableaux, la présence de nœuds colorés a différentes conséquences : un nœud peut seulement être bloqué par un nœud ancêtre de la même couleur et la condition de blocage appliquée à ces nœuds dépend de la logique du composant ontologique.

En conclusion, dans le contexte du web sémantique, le Framework basique pour la création d'ontologies collaboratives ainsi que le processus correspondant sont arrivés à maturité dans leur compréhension commune, alors que les systèmes de gestion de l'évolution des ontologies sont toujours dans un processus d'amélioration. Les travaux sur le versionnage des ontologies du web sémantique ainsi que sur l'évolution des ontologies sont donc très importants à cet égard (Plessers P., 2006). De l'état de l'art sur le sujet, on peut dire que les processus de correspondance d'ontologies ou les algorithmes de versionnage demandent encore une intervention humaine et que pouvoir gérer ces interventions et leurs interactions seraient une véritable innovation.

II.4 Processus de développement d'ontologie

L'état de l'art sur le développement des ontologies insiste sur l'utilité de processus standardisés, problème auquel le projet Européen Neon⁶ apporte une réponse. Celui-ci identifie les besoins opérationnels et fonctionnels de différents projets de développement d'ontologies. Cela a permis de mettre en évidence le fait que les besoins fonctionnels du développement d'ontologies changent en fonction du degré de réutilisabilité de l'ontologie, de son étendue et de sa flexibilité. De même, le processus de développement d'ontologies change d'un projet à l'autre de manière significative. De fait, une architecture et un processus suffisamment flexibles et robustes sont nécessaires pour prendre en compte différentes situations de développement d'ontologies.

Parmi les ontologies les plus connues, on peut citer :

- L'ontologie Gene (Noy N.F. *et al.*, 2006) qui est une ontologie bien connue pour le développement des informations génétiques. L'équipe d'experts applique régulièrement des évolutions à l'ontologie qui devient une partie des évolutions avec la validation de décideurs centraux. Ainsi, l'évolution des différentes versions de l'ontologie est entretenue et les changements sont sauvegardés dans la base centrale de l'ontologie devenant une partie intégrante de celle-ci.
- Le Thésaurus du NCI (National Cancer Institute) (Tudorache T. *et al.*, 2008) qui utilise un processus intégré pour la formalisation et la distribution des évolutions de l'ontologie. Le corps central transmet à intervalles réguliers, une liste de suggestions possibles et ces changements sont validés ou refusés par les experts du domaine

⁶ <http://www.neon-project.org/>

distribué qui publient leurs suggestions a la fin du processus. Enfin, l'éditeur principal surveille les évolutions faites et met à jour la version de l'ontologie.

- L'ontologie du FAO (Food Agricultural Organization), dont le processus de développement applique les évolutions de l'artefact ontologique en créant tout d'abord un « brouillon » qui sera approuvé avant publication.

II.4.1 Développement d'ontologies collaboratives

Le processus ontologique générique présenté par (Sebastian A. *et al.*, 2008) et (Tudorache T. *et al.*, 2008) identifie les besoins suivant pour développer une ontologie collaborative :

1. Intégration des annotations dans le développement d'ontologie ;
2. Support de différents niveaux dans l'expressivité du changement des ontologies ;
3. Gestion des utilisateurs et identification de la provenance de l'information d'évolution ;
4. Contrôle d'accès ;
5. Support de processus ;
6. Accès synchrones et asynchrones aux ontologies partagées.

Un autre système de gestion des ontologies collaboratives présenté dans la littérature est (Gaeta M. *et al.*, 2009) qui supporte une maturation d'une ontologie collaborative à partir d'une ontologie « graine », basée sur le principe de maturation des ontologies présenté dans (Braun S. *et al.*, 2007). De façon similaire, (Tudorache T. *et al.*, 2008) et (Noy N.F. *et al.*, 2006) discutent de la création d'une ontologie collaborative via l'éditeur OWL Protégé en introduisant un processus d'évolution des ontologies. Tous les processus de maturation proposés sont constitués des activités suivantes :

1. **Activité d'initialisation** : cette activité est nécessaire pour éditer un premier brouillon de l'ontologie qui sera utilisé comme une version initiale pour guider les processus de collaboration.
2. **Activité de perfectionnement** : cette activité est utilisée pour modifier une version précédente de l'ontologie de façon à obtenir une version améliorée.
3. **Activité d'harmonisation** : cette activité est exécutée lorsqu'il est nécessaire d'harmoniser deux ou plus ontologies d'un même domaine. La tâche d'harmonisation est réalisée par un algorithme semi-automatique de fusion d'ontologie.

4. **Activité de validation :** cette activité est exécutée de façon à vérifier l'existence d'un consensus entre tous les participants au processus de collaboration et pour réaliser le dernier perfectionnement collaboratif. Un coordinateur spécial modère l'activité de collaboration et décide lorsque celle-ci est terminée.

Au final, il semble bien qu'un système de gestion des ontologies collaboratives soit considérable comme un véritable avantage dans l'accumulation de la connaissance. Cependant, un Framework de maturation des ontologies existe, mais aucun processus formel ni aucune étude sur les besoins d'une architecture n'ont été présentés auparavant dans la littérature.

Il nous a donc semblé intéressant de contribuer dans ce domaine en proposant une architecture qui présente les composants basiques pour la gestion collaborative du processus de création d'ontologies, appliquée aux réseaux d'entreprises, en basant notre maturation d'ontologie sur les approches existantes. De même, les spécificités du processus appliqué aux composants de l'architecture reprendront les tâches présentées ici.

Pour y parvenir, nous avons premièrement identifié et catégorisé toutes les structures pertinentes en lien avec la collaboration d'entreprise, puis deuxièmement, identifié la séquence d'activités nécessaire pour intégrer les objets d'information dans l'artefact ontologique, tout en prenant en considération le processus de développement d'ontologie. Nous présenterons ce modèle dans le Chapitre III.

II.5 L'ontologie, un media d'intégration dans l'entreprise collaborative

Dans le contexte de l'intelligence artificielle, l'ontologie est donc une structure qui spécifie une conceptualisation d'un domaine, ou plus précisément, une spécification de la conceptualisation partagée d'un domaine (R T. and Gruber, 1993). La future génération de Framework pour la gestion partagée de la connaissance prévoit que les ontologies soient la technologie centrale pour l'intégration de l'information dans un environnement distribué. Les ontologies et les règles d'affaires sont les composants clés d'une approche guidée par modèle de l'intégration d'entreprise. La recherche appliquée dans ce type d'approches s'intéresse à améliorer les mécanismes des règles d'affaires ainsi que les systèmes de gestion des processus d'affaires via les ontologies (Lin H. and Harding J., 2007). D'autre part, la recherche sur l'intégration d'entreprise se penche sur les technologies de description de services, basées sur les ontologies, pour la collaboration interentreprises (Cheng Y. *et al.*, 2009).

L'objectif principal de cette section est de fournir un état de l'art sur différents scénarii d'application des ontologies et plus spécifiquement sur les ontologies sémantiques web basées sur la Logique Descriptive avec OWL-DL, dans le cadre de la gestion de la connaissance et de l'information pour les collaborations d'entreprise. Nous allons donc d'abord discuter des difficultés dans l'utilisation des technologies sémantiques en ce qui concerne la gestion de l'information industrielle, puis nous aborderons des perspectives pratiques sur les ontologies et l'intégration de documents tout en considérant la limitation des langages de représentation ontologique ainsi que la limitation actuelle des mécanismes de raisonnement et d'inférence.

Dans l'environnement actuel, la connaissance créée par/dans les différentes organisations doit être intégrée avec des artefacts informationnels pour fournir un contexte correct pour l'information. Cette intégration facilite la réutilisabilité ainsi que la recherche des artefacts informationnels. Un des mécanismes les plus populaires pour une telle intégration est l'utilisation d'ontologies. Les limitations de ce type d'intégration sont dans les efforts supplémentaires et simultanés nécessaires à la gestion des artefacts d'information et des ontologies ainsi que dans le manque de Framework et de méthode pour une représentation correcte de la connaissance et des sources d'informations dans une entreprise collaborative. Une autre limitation notable de l'utilité de l'intégration ontologique des informations et des sources de données est la définition d'un modèle d'intégration, c'est-à-dire soit une seule ontologie centralisée, soit de multiples ontologies distribuées, soit encore une approche hybride de combinaison d'ontologies avec un vocabulaire partagé. Il faut cependant noter que les services de raisonnement et d'inférences des différents outils ontologiques utilisés en pratique sont limités uniquement au modèle de l'ontologie locale centralisée (Horrocks I. *et al.*, 2003), (McGuinness D.L. *et al.*, 2004), (Horrocks I. *et al.*, 2000), (Sirin E *et al.*, 2007). Le Framework de raisonnement distribué est quant à lui toujours en développement (Serafini L. and Tamin A., 2005).

II.5.1 L'ontologie centrale comme media d'intégration de l'information

L'utilité de l'ontologie en tant que média d'intégration dans la conception organisationnelle est abordée dans (Pirro G. *et al.*, 2010) où différentes ontologies d'organisations sont utilisées pour gérer les différents flux de connaissances dans les différents processus d'organisation au sein de l'entreprise collaborative. L'espace de travail ontologique définit les relations entre les différents groupes de travail dans l'organisation. Les ontologies personnelles mettent en relief et connectent les caractéristiques de chaque acteur dans l'organisation. Par exemple, COKE

(Core Organizational Knowledge Entities ; Entités de Connaissances Organisationnelles Centrales) est un ensemble d'ontologies définissant des Objets de Connaissance (KO), des Services d'interrelations (SER), les interrelations de Ressources Humaines (HR), les Interconnexions des Ressources Technologiques (TR), etc. auxquels s'ajoute l'Ontologie Supérieure qui contient un ensemble de concepts Méta pertinents pour l'organisation. Le seul défaut de cette représentation se situe dans les capacités de raisonnement au travers de multiples ontologies individuelles qui ne sont pas totalement développées dans l'article.

L'avantage basique des systèmes basés sur des ontologies pour l'intégration de données ou d'informations est dans la satisfaction de la demande. En effet, d'une part, le demandeur d'informations peut formaliser la requête en termes de concepts de domaine et d'autre part, les entreprises collaboratives peuvent partager leurs connaissances sur le cycle de vie du produit avec les connaissances d'autres intervenants pour augmenter la valeur de connaissance du produit. Ces avantages des systèmes d'information basés sur les ontologies, en particulier pour la création collaborative de cycle de maintenance du produit à travers des partenaires distribués, sont abordés dans (Chen Y. *et al.*, 2009). Le principal intérêt dans cette application se situe dans le fait que les différents partenaires utilisent des ontologies locales séparées durant la phase de développement du produit puis, après un processus de fusion, utilisent l'artefact ontologique résultant pour l'intégration d'information et la satisfaction des requêtes. Les contributions marquantes de cette étude sont :

- L'identification de modèles opérationnels pour l'intégration de la connaissance au sein de l'entreprise collaborative ;
- L'identification d'un Framework pour intégrer différents modèles d'ontologies dans l'entreprise distribuée ;
- L'intégration de différentes ontologies dans le contexte du « Product Lifecycle Management » (PLM) ;
- La proposition de modifier l'algorithme ASMOV pour calculer les similarités entre les concepts ontologiques et les relations dans le processus d'intégration d'ontologies en se basant sur leurs caractéristiques lexicales et structurales ;
- La proposition d'une procédure de satisfaction des requêtes basée sur la connaissance produit.

Un résumé de la procédure de fusion d'ontologie est représenté Figure 3.

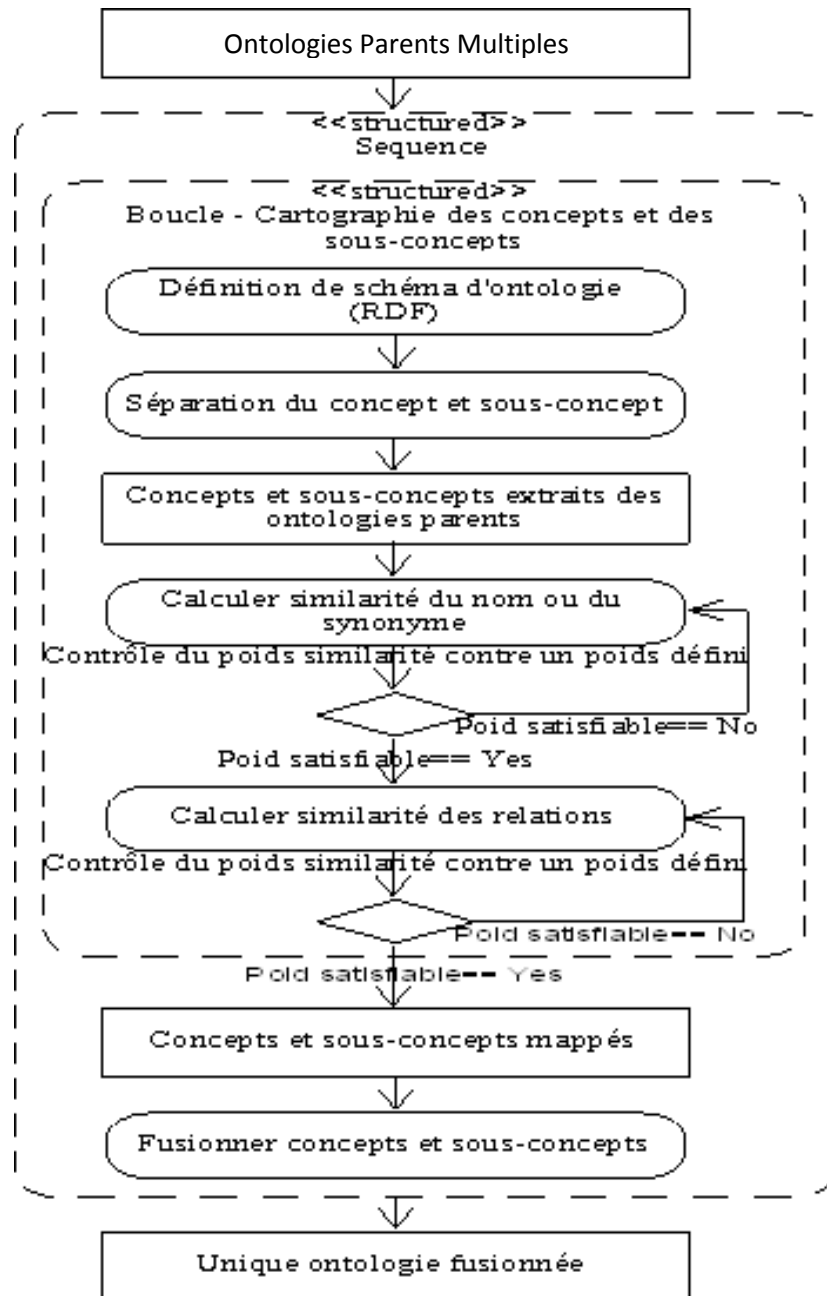


Figure 3 : Procédure de fusion de l'ontologie pour l'intégration de l'information sur le cycle de vie du produit

II.5.1.1 Intégration d'ontologie : le standard industriel

(Juhnyoung Lee *et al.*, 2006) proposent une solution autour d'une base de donnée centrale pour utiliser la puissance d'intégration des ontologies. Ils présentent SnoBase, développée par IBM, comme un système de gestion des ontologies. SnoBase répond aux limitations actuelles dans l'utilisation de la mémoire pour les inférences d'ontologies. En effet, bien que ces services d'inférence soient utiles pour traiter avec les hiérarchies de classes sur les ontologies de petites ou moyennes tailles, ils ne sont pas développés pour des applications qui impliquent une grande quantité d'instances de données. Les domaines existants de l'application

d'ontologies (e-commerce et bio-informatique principalement) traitent une très grande quantité de données préexistantes qui doivent être liées aux ontologies. Les outils conventionnels supportent la migration des instances de données vers les artefacts ontologiques ce qui a pour résultat de répliquer continuellement les instances de données, augmentant ainsi le problème de synchronisation entre les sources de données et l'artefact ontologique. La solution pour une inférence efficace de SnoBase est de limiter la capacité expressive de l'artefact ontologique. De plus, des services supplémentaires comme, par exemple, les accès concourants, l'intégrité des données ou la gestion mature de la base de données sont aussi pris en compte dans la conception de l'architecture du système de gestion d'ontologies SnoBase.

D'autres explorations de l'aspect intégratif des ontologies, utilisant les ontologies de Logique Descriptive pour relier une collection de données existantes de façon autonome, peuvent être trouvées dans (Poggi *et al.*, 2008). Dans ces accès aux données via ontologies, celles-ci décrivent le domaine d'intérêt à un haut niveau d'abstraction et les problèmes spécifiques liés aux interactions entre les ontologies et les sources de données sont les suivants :

- Un des principaux défis à cet égard, est d'identifier un langage ontologique qui fournisse un compromis acceptable entre la puissance expressive et la complexité computationnelle du raisonnement tant au niveau des ontologies que des sources de données liées. Un moyen de contourner ce problème est d'imposer une restriction sur l'expressivité du langage ontologique, garantissant ainsi que le raisonnement restera faisable en ressources finies du point de vue de la complexité des données. Des restrictions possibles qui garantissent le raisonnement dans un espace polynomial ont été proposées dans le contexte de la Logique Descriptive (Hustadt U. *et al.*, 2005). Ces logiques permettent de répondre à des requêtes complexes (conjonctives) dans un espace logarithmique.
- Pour l'accès aux données basé sur les ontologies, la correspondance est un outil formel qui détermine comment lier les données aux ontologies. Les assertions de correspondances des données sur les ontologies sont utilisées pour faire correspondre des données de la base à des instances de concepts, rôle et attributs dans les ontologies.
- En général, il y a une divergence entre la façon dont les données sont représentées dans la base relationnelle et celle dont les informations correspondantes sont rendues

dans l'ontologie (la base de données stocke des données, les instances de concepts dans l'ontologie sont des objets, chacun identifié par un identifiant). Un tel problème est nommé « divergence d'impédance » (Meseguer J. and Qian X., 1993). D'autres problèmes sont nécessaires à prendre en compte lorsque l'on met en correspondance une source de données relationnelle avec une ontologie, en particulier au niveau de l'intégration et de l'alignement des ontologies.

- La principale raison pour construire un système d'accès aux données ontologiques est de fournir des services de haut niveau aux clients du système d'information. Le plus important de ces services est la réponse aux requêtes. Le client exprime sa requête en termes de vue conceptuelle (l'ontologie) et le système doit être en mesure de traduire la demande en questions appropriées destinées aux sources.

La recherche dans ce domaine s'attache à accroître les capacités de représentation aussi bien que de raisonnement des familles de Logique Descriptive (Grau B.C. *et al.*, 2008).

II.5.1.2. Ontologies et Bio-informatique

Heraclitus II (Mikroyannidis A. and Theodoulidis B., 2010) est un autre Framework qui approche le problème de l'intégration sémantique des données. Il fournit, en effet, un mécanisme tangible pour exploiter l'intelligence d'affaires par une population automatique d'instances d'ontologie avec les données utiles des processus d'affaires puis applique des règles d'affaires pour extraire les informations appropriées plus tard dans le processus.

II.5.2 Ontologies pour la gestion du contenu

Les ontologies sont un moyen puissant de représenter la connaissance, en particulier dans le contexte du travail collaboratif où chaque partenaires à une perspective différente. En effet, les ontologies fournissent alors un artefact interactif central pour la satisfaction des requêtes qui réduit l'écart sémantique entre les humains et les machines (Chu H.C. *et al.*, 2009) (Figure 4). Les trois types d'écarts sémantiques possibles sont :

1. **Humain/Machine** : la sémantique est constituée de concepts de domaine et leurs combinaisons logiques dans une forme implicite à partir d'un modèle mental humain. Lorsque la sémantique est exprimée en mots, l'absence d'un vocabulaire commun peut mener à une altération de la sémantique originale, puisque des personnes différentes peuvent avoir des choix de mots différents, d'où l'écart sémantique.

2. **Machine/Machine** : dans la gestion de contenu, les tâches telles que la recherche de contenu sont confiées à un agent de recherche pour une recherche fédérée sur de multiples serveurs. Comme le langage humain contient des synonymes et des homonymes, un même vocable peut refléter différentes significations dans différents contextes. De telles nuances ne sont pas traitées par des recherches basées sur mots clés, d'où un écart sémantique.
3. **Machine/Humain** : il s'agit d'une reproduction du premier type d'écarts sémantique humain/machine. Puisque les humains interprètent les informations données à partir de leur propre modèle mental, un écart sémantique apparaît.

Cette approche centrée sur un document utilisant les ontologies comme un facilitateur pour réduire les écarts sémantiques demande toujours trois entités de données de base :

1. Artefact de connaissance (ontologie) ;
2. Artefact d'information (contenu/document) ;
3. Annotation de l'artefact de connaissance vers l'artefact informationnel.

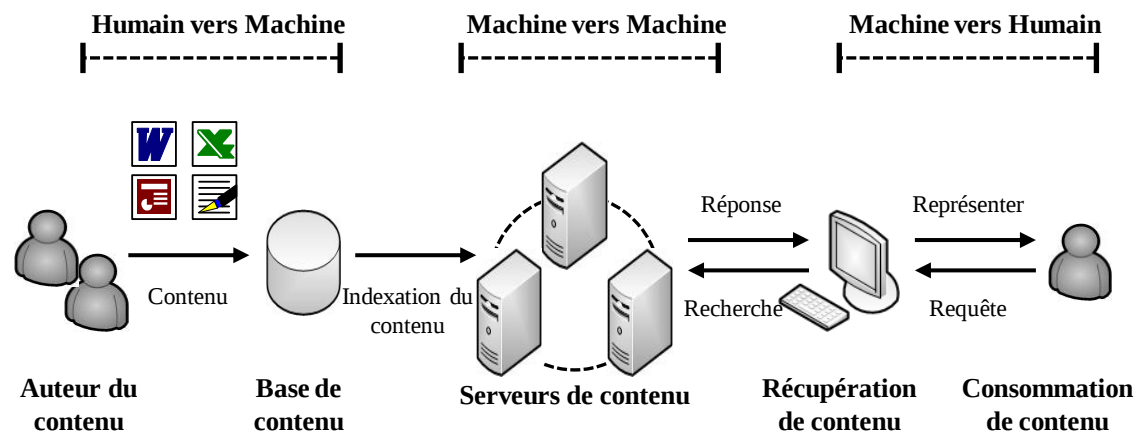


Figure 4 : Les trois types d'écart sémantiques (Chu H.C. et al., 2009)

Les spécifications pour un système d'annotation sont identifiées par (Uren V. et al., 2006) (Figure 5) et les principales sont :

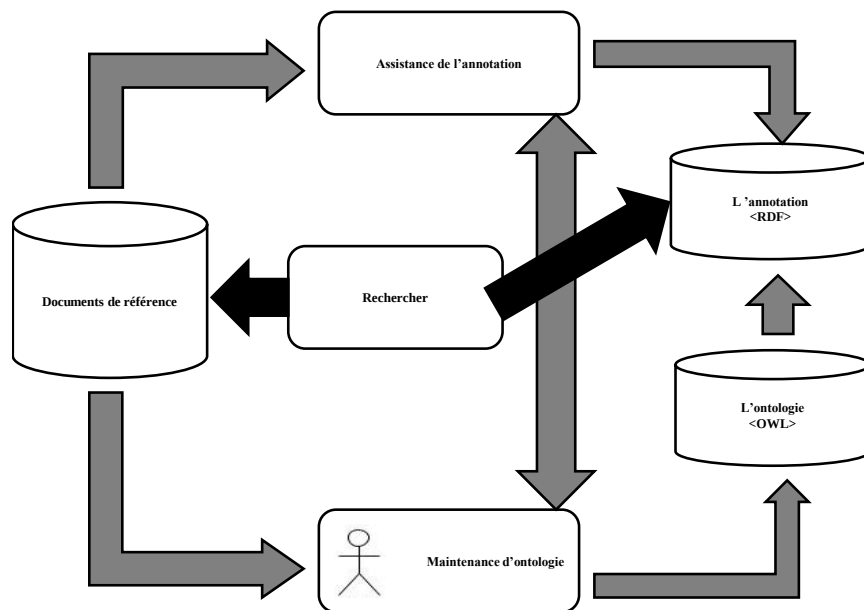


Figure 5 : Rôle de l'annotation dans la gestion des connaissances centrée sur le document (Uren V. *et al.*, 2006)

1. Un format standard pour la représentation de l'annotation. Ceci afin de faciliter le partage des annotations entre les utilisateurs collaborant.
2. Un processus d'annotation qui est une tâche longue et ardue, ce goulot d'étranglement dans le processus peut être soulagé en utilisant une interface à un seul point d'entrée. Cela facilite la collaboration entre les différents annotateurs et peut fournir un mécanisme pour résoudre les problèmes de provenance des annotations, droits d'accès, etc. En plus de supporter les formats d'ontologies, le système d'annotation doit aussi supporter les évolutions/modifications d'ontologies via une déclaration explicite de la version de l'ontologie dans l'annotation.
3. La possibilité de supporter de multiples formats hétérogènes de documents
4. La possibilité de supporter les évolutions de documents puisque le contenu d'un document annoté peut être réordonné, ajouté, soustrait de manière à conserver la consistance de l'artefact informationnel par rapport à l'artefact ontologique.
5. Un modèle sémantique web qui suppose que l'annotation sera stockée séparément du document originel, alors que le modèle de traitement de mots suppose que le stockage d'annotation soit une partie intégrante de l'artefact informationnel. Tout système d'annotation doit donc respecter ces deux points de vue dans son processus.
6. L'intégration des technologies d'extraction de la connaissance dans l'environnement d'annotation est vitale. Celles-ci permettent, en effet, d'identifier automatiquement les

concepts de classes, leurs interrelations et les entités individuelles qui peuplent les classes.

On peut noter que les recherches sur les interactions homme/machine sont aussi utiles ici puisque des outils automatisés peuvent être utilisés pour aider les utilisateurs avec une expertise limitée, en particulier via des méthodes automatiques permettant des processus en langage naturel.

La recherche sur l'intégration des ontologies avec les artefacts d'information pour la gestion de l'information s'intéresse plus particulièrement à des domaines tels que les documents sémantiques, les bureaux sémantiques, le E-learning (pour la gestion de contenu) et l'intégration de l'information web. La stratégie de base de l'intégration est de définir à un méta-niveau la structure de contenu et la provenance de son domaine correspondant via une annotation ontologique. Les différents artefacts ontologiques utilisés pour accomplir cette stratégie diffèrent en fonction des besoins et peuvent être, par exemple : ontologie document, ontologie annotation, ontologie domaine, évolution d'ontologie, etc.

II.5.2.1. Document sémantique

Peu de projets ont véritablement exploité le plein potentiel des annotations de document ontologiques pour la gestion d'information. En partie à cause des besoins additionnels des annotations et en partie en raison du manque de méthodes standards pour intégrer les artefacts informationnels et ontologiques. En fait, le concept de document sémantique est le premier effort en ce sens. La recherche sur les documents sémantiques a toujours supposé plusieurs représentations ontologiques et (Eriksson H., 2007) fournit un modèle et une épreuve d'implémentation pour combiner les formats de documents existants avec des ontologies de domaine.

Les trois processus illustrés en Figure 6 correspondent aux trois développements alternatifs pour un document sémantique. Le processus (a) illustre les activités de développement d'un document sémantique à partir d'un artefact document existant, le processus (b) illustre le développement ou la réutilisation d'ontologies domaines suivies par la réalisation et l'annotation de document (ici le processus inclus la rédaction du document sémantique en accord avec l'ontologie domaine) et, enfin, le processus (c) illustre la génération de document à partir d'une ontologie source, approche aussi abordée dans (Weal *et al.*, 2007) pour qui les

ontologies sont utilisées comme un artefact central pour réorienter le matériel existant à partir du web.

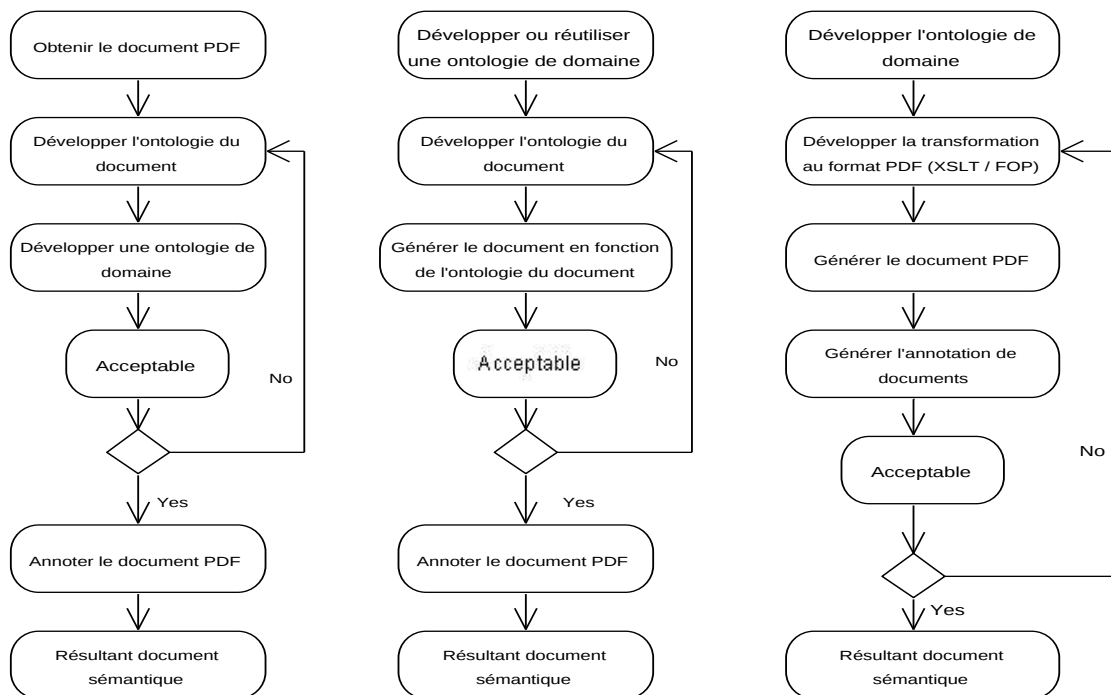


Figure 6 : Processus de développement du document sémantique a, b et c (Eriksson H., 2007)

PDFTab (Eriksson H., 2007) est un outil d'annotation sémantique dédié aux artefacts informationnels existant en PDF (Portable Document Format) avec une interface à entrée unique. Le but de cet outil est d'annoter l'artefact informationnel avec l'ontologie domaine en supposant une structure de document constante. Cependant, par rapport à l'analyse de besoins des outils d'annotation, PDFTab ne permet pas les processus collaboratifs pour l'activité de notation, ni ne s'intéresse aux mécanismes de changement/évolution des annotations associés aux modifications dans la structure ou le contenu du document. Un tel mécanisme de changement dans la structure du contenu est présenté dans (Nešić S., 2009) avec la définition d'une ontologie additionnelle de changement ainsi que d'une ontologie document à un niveau méta bien plus compréhensible qui représente une large gamme d'artefacts informationnels. De fait, nous allons décrire les ontologies correspondantes telles qu'expliquées par (Eriksson H., 2007) et (Nešić S., 2009).

Ontologie annotation : décrit le modèle d'annotation du document. Elle contient les classes demandées pour instancier les individus annotation. De plus, cette ontologie capture les propriétés des annotations telles que la position du numéro de page, les textes en surbrillance, etc. Dans le modèle de document sémantique de (Nešić S., 2009), le principal objectif de

l'ontologie d'annotation est la découverte ou la réutilisabilité du contenu. Cette ontologie est divisée en deux catégories : annotation « sans contexte » et annotation « contexto-dépendantes ». La première est utilisée pour annoter du contenu avec un vocabulaire internationalement reconnu comme Dublin Core (DC) (Weibel, 1999) ou IEEE LOM (Learning Object Metadata) (Urban & Barriocanal, 2003), alors que la seconde est très semblable à la définition générale des ontologies vue précédemment.

Ontologie document : modélise la structure logique du document. Cette ontologie dépend du format du document et contient des concepts correspondants à chaque partie du document, comme, par exemple, chapitres, sections, paragraphes, figures, etc. Dans (Nešić S., 2009) l'ontologie document modélise la structure du document comme des unités de contenu identifiées uniquement par des ressources, lesquelles renferment les différentes parties du contenu et peuvent être extraites du contexte du document pour être réutilisées dans un autre document.

Ontologie domaine : contient le modèle de contenu du document. Elle décrit donc la terminologie utilisée dans le document et modélise au moins une partie des sémantiques sous-jacentes. L'ontologie domaine est dépendante du contexte de son utilisation. Par exemple, si le propos de l'intégration ontologique est d'améliorer la collaboration dans un réseau d'organisations, alors l'ontologie domaine définit les concepts basiques des objectifs de collaboration (Santoro F.M. *et al.*, 2006). De même, si l'objectif est la gestion du contenu, alors le vocabulaire de contrôle, les taxinomies, etc. peuvent être modélisés comme des ontologies domaines avec les relations pertinentes entre les concepts (Dotsika F., 2009), (Kiyavitskaya N. *et al.*, 2009), (Kim M. and Compton P., 2006), (Buffa M. *et al.*, 2008), (Price S.L. *et al.*, 2009), (Chu H.C. *et al.*, 2009).

Ontologie modification : seulement décrite par (Nešić S., 2009), elle fournit un modèle pour les modifications appliquées à la structure logique en accord avec l'ontologie document. Les modifications à la structure logique du document peuvent intervenir par addition, soustraction ou réorganisation des unités de contenu.

Nous avons donc vu quelle peut être l'utilité des ontologies comme artefact d'intégration pour l'information, identifié différents contextes dans lesquels l'utilisation des ontologies est possible et les procédures correspondantes. Pour ce qui est du contexte particulier de

l'intégration de l'information inter-entreprise, un certain nombre d'observations fondamentales ressortent de la littérature :

1. L'annotation de documents sémantique est le premier pas pour l'intégration des artefacts informationnels d'entreprise, mais pour une utilité plus grande, la standardisation du modèle de représentation de l'artefact informationnel est importante.
2. Le processus d'annotation peut être amélioré par la collaboration entre utilisateurs.
3. Une interface à une seule entrée munie d'un contrôle d'accès et de l'identification de la provenance des utilisateurs est une vraie amélioration dans le processus d'annotation.
4. Avec l'intégration des artefacts d'information (documents) avec les multiples artefacts de connaissance (ontologies), un mécanisme de raisonnement est aussi nécessaire afin de répondre aux requêtes sur de multiples ontologies modulaires, ce qui n'est pas encore possible.

II.5.2.2. Rédaction collaborative

Bien que la Co-rédaction d'un simple document texte soit une sous partie bien explorée du travail collaboratif assisté par ordinateur, il n'y a actuellement pas de solution avancée pour étendre la Co-rédaction au niveau sémantique. Nous allons voir dans cette partie quelles options sont disponibles pour la rédaction collaborative et comment elles peuvent s'intégrer dans un processus holistique pour la création de documents sémantiques dans le cadre de la collaboration d'entreprise.

Les besoins principaux d'un processus de rédaction collaboratif conventionnel sont une identification des activités des collaborateurs, une collaboration intra et inter activités adéquate et l'existence d'informations de sensibilisation. En effet, les corédacteurs doivent être conscients du sujet, de l'artefact en cour de création (sensibilisation d'artefact), des acteurs impliqués dans le processus de rédaction (sensibilisation d'acteurs), de l'avancée du processus (sensibilisation processus) ainsi que des tâches, intentions ou contextes (mécanismes de sensibilisation auxiliaires), etc. (Decouchant D. *et al.*, 2009), ce qui leur permettra de gérer au mieux leurs activités.

Ces propriétés de sensibilisation n'ont pas encore été appliquées aux documents sémantiques et il semble donc exister un besoin d'un processus adapté pour gérer les fonctionnalités de

sensibilisation conventionnelles tout en respectant les limitations d'une collaboration dans un environnement internet plus centré sur les données.

(Sun C. *et al.*, 2006) introduisent une approche d'intégration pour la conversion d'un programme mono-utilisateur commercial en un programme multi-utilisateurs de rédaction collaborative. Cette approche est nommée l'Adaptation Transparente (TA). Elle utilise les techniques de la Transformation Opérationnelle (OT) pour atteindre ces capacités. L'approche de l'adaptation transparente ne modifie pas le code source de l'application, mais reçoit les actions de l'utilisateur et les transforme opérationnellement dans un environnement collaboratif tout en maintenant la consistance des données et des opérations. De fait, les utilisateurs finaux peuvent collaborer en utilisant une application avec laquelle ils sont familiers.

Pour que cette approche soit applicable, il faut que les applications partagées soient adaptables aux modèles de données et d'opérations des techniques de la transformation opérationnelles. Ces deux modèles sont, premièrement, le modèle de données qui définit la façon dont les objets données du document sont adressés par les opérations et, deuxièmement, le modèle opération qui définit l'ensemble des opérations qui peuvent directement être transformées par les fonctions de transformation d'opérations.

Afin que ces techniques de transformation d'opération puissent être appliquées dans le contexte des ontologies sémantiques web, le modèle de données d'OWL-DL ainsi que les opérations d'évolution des ontologies apparaissent comme un bon point de départ et la recherche sur cet objectif est discutée dans (Plessers P. *et al.*, 2007). L'approche de l'adaptation transparente est applicable pour les collaborations centrées sur les données dans lesquelles l'objectif principal est un partage de données concourant et cohérent. Il existe une variété de techniques d'OT définies pour des documents en texte dans (Adler A. *et al.*, 2006), (Sun C., 2002), (Sun C. *et al.*, 1998), (Ressel M. *et al.*, 1996) et pour des documents de type XML/HTML (Davis A.H. *et al.*, 2002).

L'architecture du système d'adaptation transparente est composée de :

1. L'application mono-utilisateur (SA) qui fournit les fonctionnalités habituelles d'une interface mono-utilisateur. Ce composant ne participe pas directement à la collaboration multiutilisateur et n'en est pas « averti ».

2. L'adaptateur de collaboration (CA) qui offre les capacités spécifiques de l'application et joue un rôle central dans l'adaptation du SA au MCG (Moteur de Collaboration Générique). Ce composant est lié aussi bien à l'application mono-utilisateur qu'aux mécanismes de collaboration multiutilisateur.
3. Le MCG fournit les capacités de collaboration indépendantes de l'application. Ce composant encapsule un ensemble de techniques permettant la collaboration avec OT au cœur du support de la consistance. Ce composant est une partie de la collaboration multiutilisateur, mais ignore tout de l'application mono-utilisateur.

L'utilisation du CA entre le SA et le MCG cache les problèmes spécifiques de l'application au MCG et facilite un débogage et un test indépendant du MCG. Cela permet aussi la réutilisation du MCG puisque pour appliquer le MCG à un nouveau SA il suffit simplement de concevoir et implémenter un nouveau CA correspondant au SA cible.

Par analogie, les besoins utilisateurs et les données pour la création d'un document sémantique, ainsi que les spécificités nécessaires à une architecture et un processus de rédaction partagée seraient :

1. Construire une base partagée qui pourrait surveiller et gérer le processus collaboratif de génération et de représentation de connaissances et fournir des fonctionnalités de sensibilisation aux utilisateurs de manière synchrone ou asynchrone ;
2. Fournir de la flexibilité aux utilisateurs (coordinateurs et collaborateurs) pour la gestion du processus ;
3. Développer une correspondance descendante des connaissances en termes d'ontologies pour aider les utilisateurs à définir et structurer leurs concepts ;
4. Supporter une gamme d'expertise et de points de vue dans la connaissance ;
5. Fournir un processus de relecture intégré ;
6. Être compatible avec un ensemble large de systèmes et de sources de connaissance ;
7. Fournir un cycle de gestion de la connaissance simple, mais structuré ;
8. Permettre la propriété primaire sur les domaines utilisateurs et non celle d'une partie tierce (ingénieurs de connaissance) ;
9. Utiliser des interfaces familières pour faciliter l'utilisation.

Une telle architecture et les processus correspondant pour l'annotation sémantique des contenus dans le cadre de l'entreprise virtuelle seront présentés au Chapitre III.

II.5.3 Les ontologies comme un media d'intégration pour les processus et produit de la chaîne logistique

Dans la section précédente, les aspects intégratifs des ontologies dans la gestion de contenu et de données sont évoqués. Nous avons présenté les différentes opérations de modification ontologiques existantes dans la littérature et les technologies correspondantes qui sont utiles dans l'implémentation des ontologies comme média d'intégration. Nous avons aussi introduit le besoin de processus pour l'amélioration des ontologies et des annotations de contenus collaboratives ainsi que les limitations de raisonnement correspondantes.

Dans cette section, nous allons aborder les aspects intégratifs des ontologies en termes d'opérations des entreprises collaboratives, l'intégrateur de système d'information et les articles pertinents à propos des modèles d'intégration d'entreprise.

L'intégration d'entreprise à des objectifs spécifiques comme établir, de façon efficace, la coordination et l'intégration des processus d'affaires et des technologies d'information ainsi que d'augmenter leur flexibilité et l'interopérabilité entre les systèmes. Un environnement intégré de production doit être composé de ressources, d'applications et de systèmes d'information qui produiront des biens et des services à travers des processus d'affaires bien établis au sein d'une entreprise collaborative. De fait, les flux physiques et informationnels doivent être clairement définis et reliés (Grubic T. and Fan I.S., 2010).

Les entreprises qui travaillent en collaboration peuvent devenir plus efficaces et plus efficaces en encourageant l'intégration dans leur chaîne logistique. Une chaîne logistique intégrée a pour avantage le partage d'information d'une manière juste et opportune ce qui induit une optimisation du flux de matériels le long du réseau d'approvisionnement et l'élimination des processus qui n'optimisent pas la valeur ajoutée (Simatupang T.M. *et al.*, 2002). L'intégration des chaînes logistiques est aussi un moyen d'implémenter des processus standards avec des modèles de données classiques pour partager l'information et optimiser la performance globale de la chaîne.

Les entreprises individuelles impliquées dans une chaîne logistique souffrent des limitations suivantes selon (Simatupang T.M. *et al.*, 2002) :

1. Fragmentation de la chaîne (chaque membre est concentré sur ces propres processus uniquement) ;

2. Manque d'intégration des processus et de décisions collaboratives entre les fournisseurs et les clients ;
3. Manque de connectivité des systèmes d'informations entre les différents membres de la chaîne logistique ainsi que de systèmes capables de décisions « intelligentes » qui peuvent déduire de manière opportune des informations adéquates pour aider la collaboration entre les partenaires.

Dans le contexte très concurrentiel actuel, de nombreuses entreprises cherchent une plus grande flexibilité sur différents aspects de leurs opérations, soit en redéfinissant leurs processus d'affaires, soit en adoptant une structure variable de leur organisation (flexibilité organisationnelle), soit encore en utilisant et partageant des équipements et des outils (flexibilité opérationnelle), soit enfin en détectant les variations de la demande ou les tendances du marché et/ou les exigences de la clientèle (flexibilité production). Dans ces conditions, l'entreprise cherche à éliminer les opérations non-productives, à optimiser la coordination des opérations entre les différents partenaires, à rationaliser l'utilisation des ressources humaines et à moderniser les systèmes d'information et de communication au sein de l'entreprise. Les objectifs majeurs sont (De la Fuente M. *et al.*, 2010) :

1. La gestion de l'intégration et la coordination des processus pour assurer des opérations efficaces et atteindre les objectifs globaux ;
2. Des systèmes de production holoniques dans lesquels, via la décentralisation de la prise de décision, les différentes entités de l'organisation sont autonomes, mais coopèrent les unes avec les autres ;
3. Des réseaux de coopération (entreprise virtuelle) permettant de créer facilement et rapidement des groupes capables de répondre aux changements constants et à la demande du marché ;
4. Des échanges d'informations et de documents électroniquement, via des réseaux informatiques qui facilitent les relations de travail, les transferts d'ordres, la gestion des stocks, etc.

De fait, l'intégration de la chaîne logistique peut amener à une meilleure performance opérationnelle et d'affaires. L'intégration des processus d'affaires clés de l'utilisateur final jusqu'au fournisseur originel qui fournit produits, services et informations ajoute de la valeur pour les clients et les autres parties prenantes. Les différentes définitions de la chaîne

logistique, qui a déjà été étendue par l'approche systémique (Harland C.M., 1996), identifient quatre niveaux de système dans les chaînes logistiques (Figure 7) :

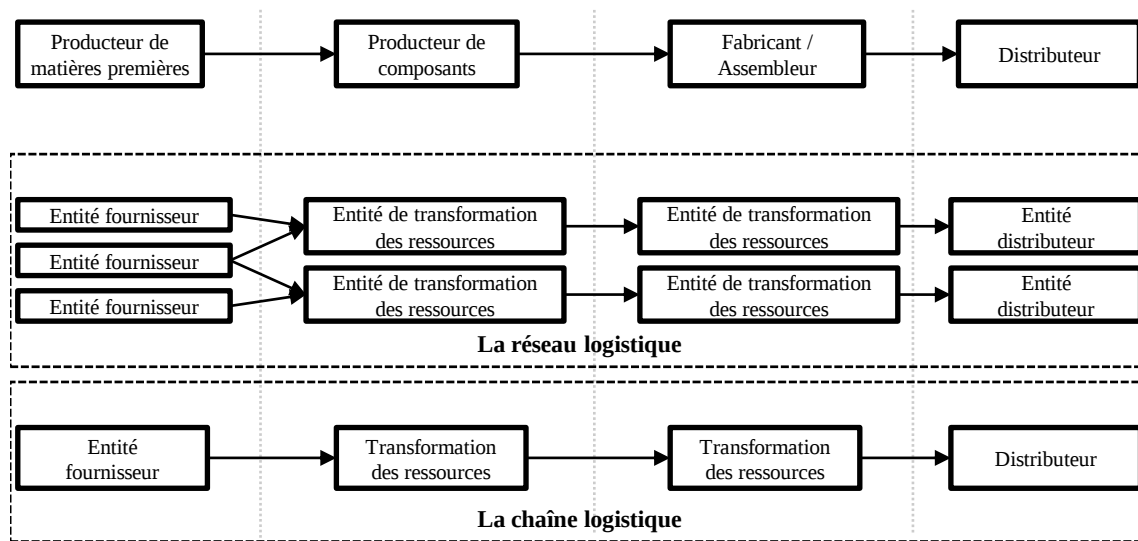


Figure 7 : Configurations structurelles de la chaîne logistique

1. Chaîne interne, qui intègre les fonctions d'affaires impliquées dans les flux de matériels et d'informations de l'intérieur vers l'extérieur ;
2. Relations dyadiques, gestion de relations bipartites avec fournisseurs intermédiaires ;
3. Chaîne logistique externe, gestion d'une chaîne d'affaire incluant un fournisseur, le fournisseur du fournisseur, un client et le client du client ;
4. Réseau inter-affaires, gestion d'un réseau d'affaires interconnectées qui sont impliquées dans la fourniture finale d'un produit ou d'un service à un client.

Alors que les deux premiers niveaux ont pour but d'intégrer et gérer les flux de matériels et d'informations dans une ou deux compagnies respectivement, les deux derniers se concentrent sur une chaîne d'affaires entière. Nous supposons ici que ces quatre niveaux suffisent à représenter l'entière des relations organisationnelles dans une chaîne logistique.

II.5.3.1. Ontologies pour l'intégration processus

(Ku K.C. *et al.*, 2008) fournissent un Framework pour la collaboration d'entreprise ou encore des entreprises en joint-venture⁷, basé sur une gestion de la connaissance ontologique. Ce Framework confirme l'utilité d'un artefact ontologique central entre les différents partenaires. Il identifie les phases pour l'intégration ontologique comme étant : la Conception, la

⁷ Ou coentreprise : entreprise commune créée et gérée à parts égales par plusieurs entreprises.

Construction, la Liaison et le Lancement. Durant les phases de Conception et Construction les experts domaines des entreprises partenaires produisent, par communication mutuelle, une ontologie qui identifie le domaine de collaboration. Puis, durant la phase de Liaison, cette ontologie identifie un vocabulaire commun qui est utilisé plus tard pour le partage d'information (Figure 8).

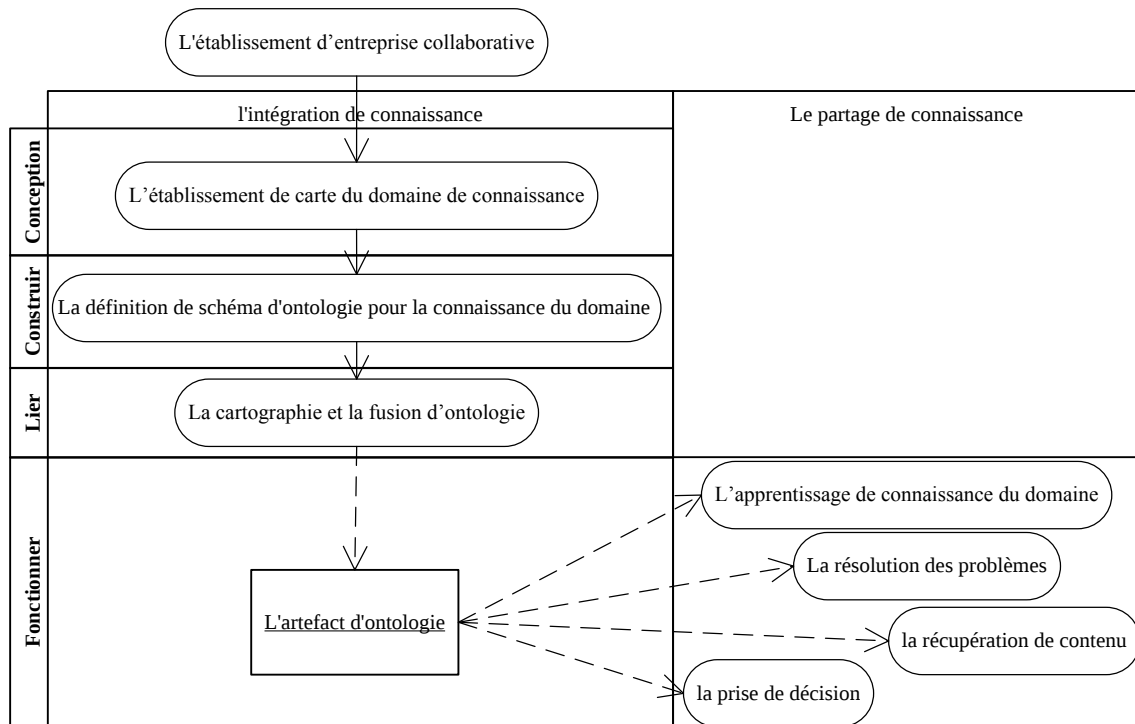


Figure 8 : Cadre de gestion des connaissances basée sur l'ontologie de l'entreprise collaborative

Les domaines pour lesquels une collaboration peut être initiée via ce Framework sont les domaines technologiques, financiers, commerciaux, de production et de ressources humaines. Ce Framework, déjà utile, est applicable au niveau stratégique dans les entreprises collaboratives, mais pas aux niveaux tactique ou opérationnel.

L'architecture de l'entreprise explicite les composants de l'entreprise systématiquement. En effet, à travers son architecture, une entreprise peut implémenter l'intégration de manière à faire face aux changements dynamiques. (Kang D. *et al.*, 2010) présentent une ontologie d'architecture d'entreprise qui facilite la compréhension universelle de la sémantique de l'architecture tant pour les humains que pour les machines, aidant ainsi l'entreprise à implémenter les mécanismes d'intégration et de collaboration. Le célèbre Framework d'architecture d'entreprise Zachman (Zachman, 1987) est utilisé dans cette implémentation

ontologique. (Brown, 2004) définit six valeurs communes de l'architecture d'entreprise qui sont :

1. La documentation facilement accessible de l'entreprise ;
2. Possibilité d'unifier et d'intégrer les processus d'affaires dans toute l'entreprise ;
3. Possibilité d'unifier et d'intégrer les données dans l'entreprise et d'établir des liens avec les partenaires extérieurs ;
4. Augmentation de l'agilité en réduisant les obstacles de la complexité ;
5. Réduction des temps d'obtention de solutions et les coûts de développement en maximisant la réutilisation du modèle d'entreprise ;
6. Possibilité de créer et de maintenir une vision commune du futur partagée par les communautés d'affaires et IT, conduisant à un alignement permanent des deux.

(Lin H. and Harding J., 2007) proposent une approche sémantique permettant une autonomie de l'information entre les entreprises collaborant en faisant correspondre les concepts similaires des ontologies des différents partenaires dans une ontologie centrale. Cette correspondance fournit une fonction de traduction arbitraire ontologique pour les applications de productions interentreprises. De manière simple, lorsque deux entreprises entrent dans une coentreprise, elles font correspondre leurs ontologies individuelles dans une ontologie EMS (Enterprise Manufacturing System) unifiée. Cette correspondance demande des experts domaines pour identifier les relations entre la terminologie individuelle des concepts avec l'ontologie EMS. Le concept d'une ontologie unique modératrice est aussi présent dans cette application, mais au même titre, et pour les mêmes raisons, que la complétude des raisonnements ontologiques ou la décidabilité des applications d'ontologies modulaires distribuées, cela n'est pas une option viable dans un contexte industriel.

Les organisations de fabrication doivent de plus en plus se concentrer sur leurs compétences de base et se désintégrer dans le cadre d'une entreprise pour éviter d'être enfermées dans des technologies inappropriées. Cette tendance à la désintégration verticale augmente flexibilité de la réponse aux besoins des clients. Cette souplesse dans la capacité de réponse engendre une collaboration plus étroite et soutient la création d'une entreprise virtuelle. L'entreprise de fabrication virtuelle dans un environnement distribué doit fournir un processus pour la coordination tant au niveau inter qu'intra-entreprise avec des technologies basées sur des services web, comme une infrastructure pour l'intégration. Dans ce modèle particulier de chaîne logistique, les entreprises impliquées visent une collaboration sur un pied d'égalité où

aucune ne doit prendre l'ascendant sur les autres. Cela implique une recherche d'optimisation globale des processus, plutôt que strictement locale, de manière à viser l'amélioration des performances de la chaîne dans son entier et non d'un partenaire en particulier. Des fonctions de coordination et de partage efficient de l'information ne peuvent donc qu'être un vrai plus dans la gestion de telles structures (Hardwick et Bolton, 1997).

(Yan J. *et al.*, 2010) présentent un Framework de service qui utilise les ontologies sémantiques pour aider les clients à personnaliser leurs tâches de production de biens en accord avec leurs besoins. Simplement en utilisant l'ontologie centrale pour représenter les besoins du processus de production et augmenter le niveau de sensibilisation pour le client. Cependant, cette approche manque d'applications ontologiques standardisées ce qui, avec l'augmentation d'ontologies processus individuelles, crée des conflits de communication entre l'agent courtier et les ontologies processus.

(Rajsiri V. *et al.*, 2010) abordent le problème d'une approche ontologique pour la conception d'un modèle de Processus Collaboratif d'Affaire (PCA). Ce PCA est la spécification de besoins pour construire un système d'information collaboratif pour un réseau d'organisations. (Jason J., 2009) fournit un alignement ontologique sémantique des processus d'affaires ce qui permet une intégration étroite des partenaires (Figure 9).

Enfin, (Chi, 2010) implémente les capacités d'intégration de la base ontologique de connaissances. L'objectif de cette implémentation étant de créer la connaissance conceptuelle des partenaires de la chaîne logistique et d'utiliser les capacités d'inférence ajoutées sur la base pour surveiller la capacité des partenaires vis-à-vis de la planification d'une production future. Les processus adoptés pour cette implémentation sont premièrement, la détermination du cadre de l'ontologie. Deuxièmement, la définition des entités d'expertises dans l'artéfact ontologique comme par exemple définir les concepts ontologiques pour les partenaires de la chaîne logistique, les dépendances correspondantes entre les concepts et les propriétés des objets associés. Troisièmement, la création d'un modèle de connaissances et son implémentations qui fournit des capacités supplémentaires pour inférer au coordinateur central des informations additionnelles sur les partenaires, fournissant ainsi un support central. Cependant, cette implémentation manque de standards de représentation des processus produits existants.

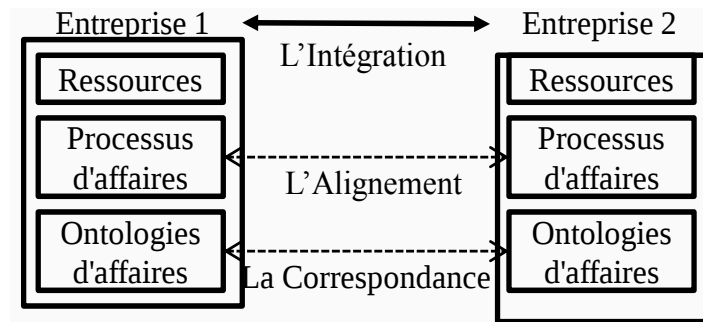


Figure 9 : Alignement des processus d'affaires

Une approche intéressante pour intégrer les partenaires d'une entreprise distribuée consisterait en l'utilisation de standards déjà usités dans les entreprises individuelles et l'application d'un outil d'inférence automatique après identification des points d'interaction entre les partenaires en respectant les limitations du partage d'information entre eux. C'est dans cette direction que nous nous dirigerons au Chapitre III.

II.5.3.2. Ontologies pour la représentation produit

Généralement, l'information produit est dispersée dans différents ERP (Enterprise Resource Planning), PDM (Product Data Management System) ou, plus récemment, dans des PLM (Product Life cycle Management). Cette distribution de l'information peut provoquer certaines redondances de données et des inconsistances. Dans le contexte de l'entreprise collaborative, ces problèmes peuvent être contrôlés par l'identification d'un modèle de données central tel que (Giménez D.M. *et al.*, 2008) l'identifie. Ce modèle sépare le modèle de données objet en deux hiérarchies distinctes : le modèle Hiérarchique Structurel organise la connaissance relative à l'information structurelle sur le produit alors que le modèle Hiérarchique Abstrait (AH) organise le produit en fonction des différents niveaux de spécification (du produit de substitution aux articles individuels). Un exemple typique d'un système qui gère une Hiérarchie Structurelle est le MRP (Matériel Resources Planning) dans lequel cette information est représentée par la nomenclature qui identifie les sous-composants et leurs quantités nécessaires. En plus de ces modèles, (Giménez D.M. *et al.*, 2008) incorporent aussi le concept de propriété pour manipuler différentes valeurs de données. Cet article s'intéresse aux besoins de modélisation, mais ne fournit pas de possibilités fonctionnelles et reste une « simple » approche de modélisation.

(Bock C. *et al.*, 2010) et (Verdouw C.N. *et al.*, 2010) proposent des approches de modélisation basées sur les ontologies et identifient les modèles de produits les plus

prolifiques utilisés en entreprise. Les approches sémantiques pour la représentation des contraintes de conception d'un produit deviennent de plus en plus populaires. (Bock C. *et al.*, 2010) et (Kim K.Y. *et al.*, 2006) identifient ainsi tous les deux les ontologies DL comme un moyen approprié de représenter les contraintes de conception d'un produit.

(Yang D. *et al.*, 2009) et (Yang D. *et al.*, 2008) introduisent une approche web sémantique pour représenter le modèle produit avec une ontologie OWL-DL et utilisent une base de connaissance ontologique pour identifier les différentes configurations produit. Cette approche bien que très utile pour les réseaux dans lesquels de nombreuses entreprises peuvent représenter leur modèle produit commun et les contraintes de configurations correspondantes, ne fournit cependant pas de modèle objet standard pour représenter les contraintes d'assemblage.

(Tursi A. *et al.*, 2009) et (Dassisti M. *et al.*, 2006) s'intéressent à une approche ontologique pour la représentation d'un modèle produit qui peut être très utile pour l'intégration d'un système d'information distribué. Cette approche utilise un modèle d'ontologie central comme artefact d'intégration pour gérer les différents processus de production et fonctionne bien dans un réseau de production complètement intégrée où les partenaires choisissent une intégration complète et totale. Dans le contexte de l'entreprise virtuelle, cette approche peut être limitée puisque les entreprises ne sont pas totalement intégrées et conservent leur autonomie, au moins partiellement. De plus, cette approche demande un régime d'implantation rigoureux qui peut mener à une limitation financière aussi bien qu'à une allocation de ressources substantielle à des capacités spécifiques. Le modèle de représentation standard de produit et processus est utilisé dans cette approche et les besoins de correspondances entre les différents modèles de concepts sont identifiés, mais aucun accent n'est porté sur le niveau de capacité de service des ontologies (tels que les capacités d'inférence) ou sur le formalisme de modélisation. La correspondance est définie en logique de premier ordre qui présente des défauts en termes de services de raisonnement, d'inférence et de satisfaction de requêtes, comme expliqué précédemment.

Finalement, (Matsokis A. and Kiritsis D., 2010) introduisent l'utilisation des approches web sémantique et utilisent OWL-DL pour représenter les informations de cycle de vie du produit. Ils basent leur approche sur des modèles standards qui sont largement acceptables par les entreprises, mais n'explorent pas les capacités d'inférence que les bases de connaissance ontologiques peuvent fournir.

II.6 Le model ISA-95

ISA-95 est une norme d'intégration industrielle, ayant pour but d'identifier les entités et leurs modèles de données au sein de l'entreprise. ISA-95 identifie donc les entités de l'entreprise, telles que les entités d'équipement, les entités matérielles, les entités de produits fabriqués, les entités de production ainsi que les entités de représentation des capacités de production. Cette standardisation des modèles de données facilite l'intégration et l'échange de données entre les entreprises partenaires. Dans cette section, nous allons présenter quelques-unes des représentations de la classe de modèle ISA-95 qui servira de base pour notre ontologie OWL-DL.

II.6.1 Classes représentant des ressources d'équipement

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 10.

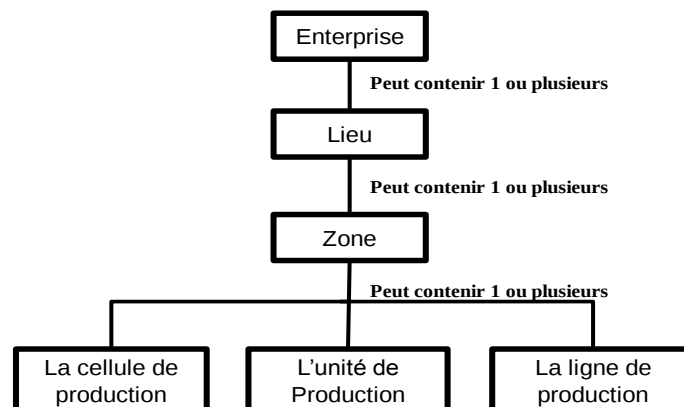


Figure 10 : Hiérarchie de l'équipement d'entreprise

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. La Classe Équipement est un moyen de décrire un groupement d'équipement, avec des caractéristiques similaires, pour l'ordonnancement et la planification. Toute entité équipement peut être un membre de zéro ou plus catégorie(s) d'équipements (Figure 11). La Classe Équipement contient la liste d'identification de tous les équipements appartenant à la classe et la

spécification de test correspondant pour la catégorie d'équipement. L'identification des ressources, selon la norme ANSI – ISA-95, se fait soit par l'ID de classe, soit par l'ID d'instance, soit en les définissant par l'ID de classe et un attribut. Un segment de processus peut exiger un certain nombre de machines à fraiser, d'une Classe Équipements spécifiée. Un autre segment peut exiger un sous-ensemble des fraiseuses avec l'attribut « Fine ». Ici, la valeur « Fine » de l'attribut des fraiseuses définit les ressources nécessaires. Alternativement, une ressource spécifique peut être précisée pour un planning de production, par exemple en spécifiant une fraiseuse particulière avec son ID.

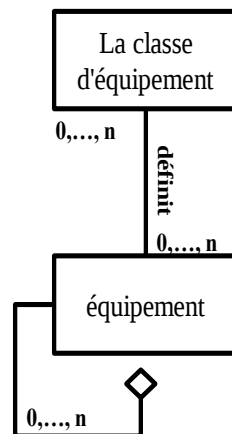


Figure 11 : Modèle objet de la classe d'équipement

II.6.2 Classes représentant les entités de produit

Définition Produit est l'objet conteneur pour l'échange d'informations concernant la famille de produits et comprend donc la référence à la famille de produits, ses règles de production, sa nomenclature et sa macro-gamme (Figure 12). Les règles de production précisent l'information qui est utilisée au sein des entités de fabrication pour la réalisation du produit, telles que, par exemple, les informations de montage. Définition Produit comporte les segments entités produits, tels que les différentes familles de produits qui peuvent être classées dans différentes définitions de produits, alors que les variantes du produit dans sa famille peuvent être classées comme des segments de produit de la même définition.

L'information associée au segment du produit définit les ressources de fabrication, le personnel, l'équipement ou le matériel qui sont nécessaires pour la production du segment de produit considéré, pour une quantité déterminée (lot standard ou la taille du lot). Pour cela, sont définies les catégories de ressources ou, dans certains cas, l'instance exacte des besoins en ressources, par exemple un segment de montage nécessite 1 assembleur pour 2 heures et 1

machine d'assemblage pendant 2 heures. Un segment de produit définit également les paramètres qui peuvent être précisés lorsque le segment est exécuté.

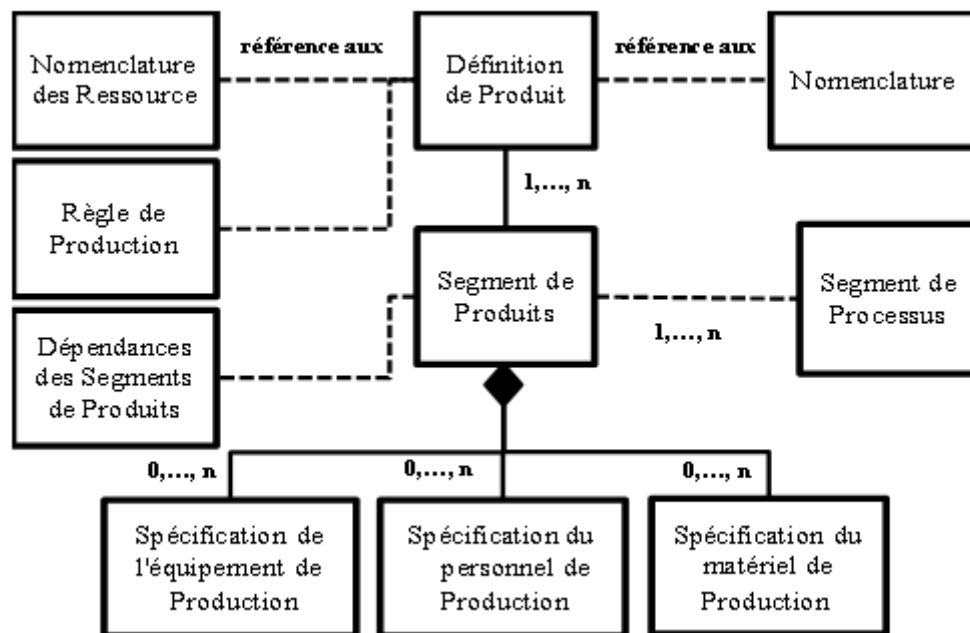


Figure 12 : Spécifications objet de la définition de produit

Les éléments de spécification d'équipement définissent les ressources matérielles, soit par une classe, soit par une instance, nécessaires pour la production d'un segment de produit. De même, les éléments de spécification des matières identifient les ressources matérielles, par un identificateur de classe ou par une ID d'instance, nécessaire pour la production d'un segment de produit. Le segment de produits comprend la quantité de ressources nécessaires pour le segment de produit, l'identification du segment processus, les paramètres associés à un segment de produit et les dépendances entre les segments de produits. Les matériaux peuvent être identifiés par la classe ou la définition du matériau.

II.6.3 Classes représentant les entités processus

L'entité segment processus définit le regroupement logique des ressources en personnels, en équipements, en ressources et en matériel requis pour effectuer une étape de la production (Figure 13). Elle définit également les ressources spécifiques, basées sur l'identification des ressources élaborées dans la représentation de la classe précédente pour les équipements. Les ressources peuvent être identifiées soit par l'identification de classe soit par la représentation individuelle d'entité.

Identifier les segments de processus nécessite une compréhension des processus d'affaires au sein de l'entreprise et de la structure générale des procédés de fabrication. Tous les segments de processus doivent se rapporter à l'étape de production, il ya au moins trois types généraux de segments de processus.

- Segments de production - sont ceux qui représentent la transformation de matières premières ou intermédiaires en matières intermédiaires ou en produits finaux.
- Segments de déplacement - sont ceux qui représentent le déplacement de matériel, l'emplacement du produit et permettent d'en garder une trace.
- Segments d'inspection - sont ceux qui représentent les tests et la pertinence des matériaux et des produits.

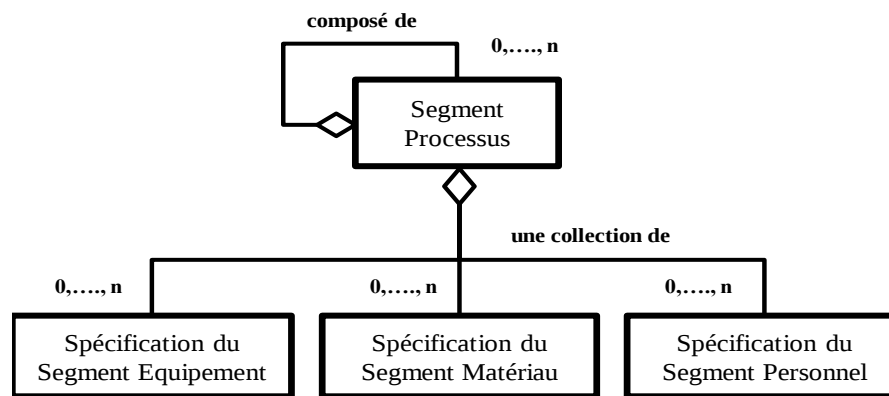


Figure 13 : Spécifications objet du segment de processus

Le segment de processus contient des segments de processus impliqués dans la fabrication du produit, par exemple un segment de l'assemblage du produit peut être constitué de segments de montage, d'essai et d'emballage. Chacun peut être défini comme un élément séparé, avec spécifications séparées basées sur la compréhension des processus d'affaires.

La spécification du segment équipements contient les spécifications sur les ressources en matériel, qui sont nécessaires à l'exécution du segment processus. Elle définit la quantité de la catégorie d'équipement ou de l'équipement nécessaire. De même, la spécification du segment matériel contient des informations sur les ressources matérielles nécessaires à l'exécution du segment processus.

II.6.4 Classes représentant les entités matériaux

La Classe Matériaux est un moyen permettant la définition du regroupement de matériaux, pour une utilisation dans un planning de production. Par exemple, une Classe Matériaux « Acier » peut contenir une définition de différents alliages d'acier. Un matériau peut appartenir à zéro ou une Classe Matériau (Figure 14). Il peut contenir la liste des définitions des matériaux appartenant à la classe et la liste des spécifications de tests d'assurance, de qualité associée à l'entité Classe Matériaux.

La définition du matériau est un moyen de décrire des produits présentant des caractéristiques similaires aux fins d'ordonnancement et de planification. Il peut contenir la liste des lots de matière appartenant à la définition et la liste des spécifications de test d'assurance qualité associé à la définition des propriétés matérielles.

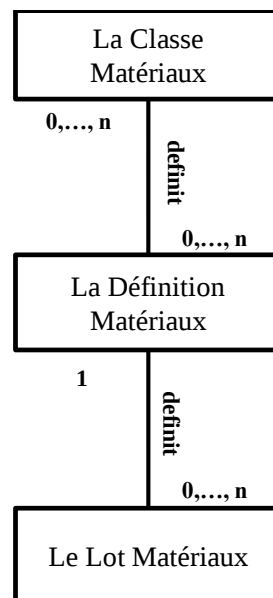


Figure 14 : Modèle Matériaux

L'objet Lot Matériaux, identifie de manière unique une quantité spécifique de matériaux, dénombrable ou pondérable. Il décrit la quantité ou la qualité de matériaux disponibles, leurs états actuels et leurs valeurs de propriétés spécifiques. Les éléments de Lot Matériaux peuvent être utilisés pour contenir des informations sur des matériaux spécifiques et sous-ensembles de la matière stockée en tant que sous-lots. Il peut également inclure la liste des définitions de matériaux et la liste des spécifications de tests d'assurance qualité associées à des propriétés.

II.6.5 Classes représentant les entités de la commande de production

La classe Planning De Production est constituée d'un ensemble d'une ou de plusieurs demandes de production (Figure 15). Le planning de production contient également l'information qui définit le contexte de la planification, tel que l'heure de début et de fin, l'emplacement et la date de fabrication. La demande de production définit la demande pour la production d'un seul produit identifié par le segment de produit.

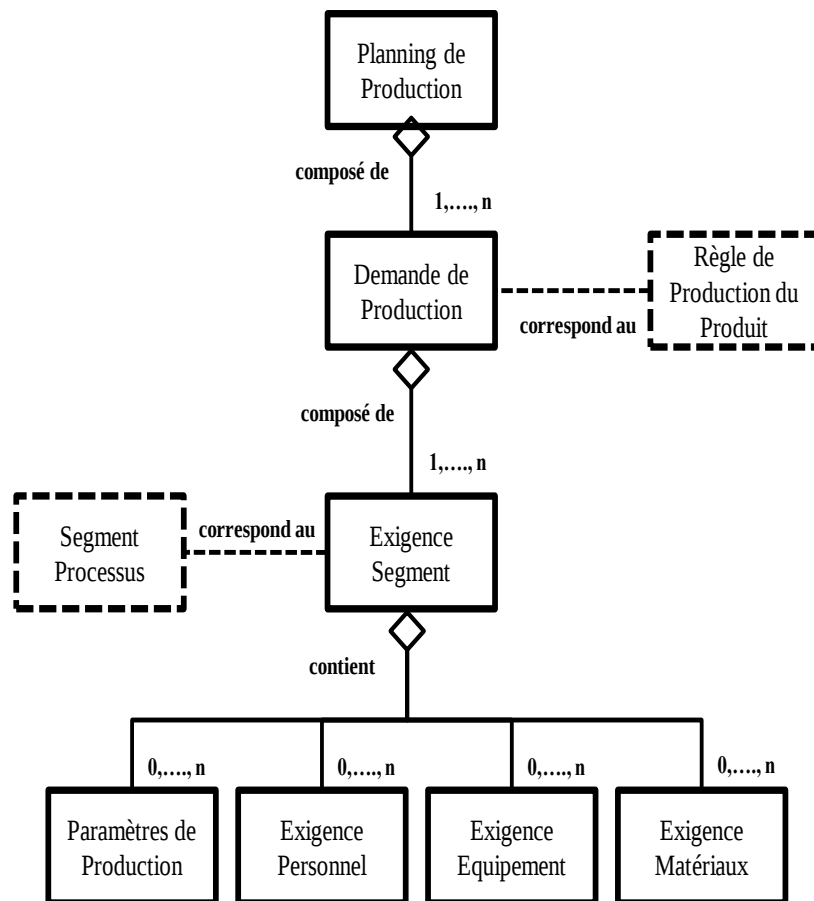


Figure 15 : Diagramme composant du planning de production

Une demande de production contient les informations requises par les entités de fabrication, pour obtenir le planning de production. Une demande de production comporte les exigences de production pour un segment de produit, l'intervalle de temps pour la production et la priorité de la demande.

Une demande de production est constituée d'une ou plusieurs exigences de segment. Les exigences de segment référencent la capacité de segment dans laquelle sont associés l'équipement, la matière et les paramètres de production. Une exigence concernant le segment peut correspondre soit à un segment de produit soit à un segment de processus. Une demande

segment devra définir les ressources et les paramètres déjà définis dans le segment processus. L'exigence concernant le segment contient la définition du planning pour un segment spécifique de la production, y compris l'identification du produit associé ou d'un segment processus, la plage de temps de la demande, la durée prévue de la demande, les paramètres de production pour le segment et la définition de l'équipement, du produit et de la matière utilisée dans la production.

La demande de production peut inclure une ou plusieurs exigences pour l'équipement que l'installation doit utiliser dans le processus de production, cette exigence peut atteindre une précision allant jusqu'à la pièce d'équipement. Chacune de ces exigences est définie dans un élément d'exigence équipement.

Une exigence matériau définit une exigence pour le matériau à être consommés. Une condition matériau peut comprendre la quantité totale du matériau devant être produite ou consommée et l'unité de mesure. Le matériau peut être défini comme une Classe Matériaux, une définition du matériau, un matériau ou un sous-lot.

II.7 Conclusion

Dans ce chapitre, nous avons parcouru les principaux thèmes d'intégration de l'information par artefacts ontologiques centraux, ainsi que les limites à leurs utilisations telles qu'elles sont décrites dans la littérature. En conclusion, il est clair que les ontologies peuvent être utilisées comme des artefacts d'intégration basiques. Cependant, les capacités de raisonnements entre des ontologies modulaires multiples sont limitées et la recherche concernant leurs utilités pour les entreprises collaboratives distribuées est assez pauvre en termes de quantité. Cela pourrait néanmoins changer drastiquement lorsque des services de raisonnement acceptables seront introduits. Dans toutes les utilisations pratiques, des ontologies OWL-DL constituent des artefacts centraux, comme les services de raisonnement, de la vérification de la consistance des ontologies, la satisfiabilité des concepts étant uniquement possible dans de telles configurations centrales.

Notre travail, ici, correspond aux besoins opérationnels pour intégrer les ontologies avec les artefacts d'information dans le cadre de la collaboration d'entreprise (Zurawski R., 2006). Sous cet aspect, nous devons identifier les composants architecturaux basiques, tout en gardant à l'esprit les besoins suivants :

1. Création collaborative d'ontologie et processus de maturation associé ;
2. Réduction de l'écart sémantique entre l'ontologie et l'artéfact informationnel, ce qui impose un processus d'annotation d'artéfacts ontologiques et informationnels ;
3. Robustesse de l'approche d'intégration qui doit aussi être capable de supporter les outils distribués nécessaires à la création/gestion d'artéfact logique (ontologies) et informationnelle (rédaction).

Finalement, le processus d'utilisation d'ontologies centrales comme un artéfact d'intégration est utile, mais ce n'est pas le seul avantage que les ontologies amènent. En effet, des services additionnels de configuration ne sont pas complètement traités dans le contexte industriel. Dans le prochain chapitre, nous allons aborder notre proposition d'utilisation des ontologies comme un artefact de configuration dans le contexte de l'entreprise virtuelle.

Chapitre III
Modélisation

Chapitre 3 : Modélisation

III.1 Introduction

Dans le chapitre précédent, le thème de base a été l'identification de l'utilité d'ontologies de web sémantique dans les différents domaines d'application ainsi que la base théorique de la Logique Descriptive. Autre thème qui a été identifié dans le chapitre précédent était la structure actuelle de l'artefact ontologique. Nous avons donc vu que l'application de l'ontologie de web sémantique nécessite d'abord l'identification de la structure de l'artefact ontologie : soit un artefact ontologique central, soit un artefact ontologique modulaire et distribué. Le problème d'identification de la structure la mieux adaptée a été contourné en notant que les services actuellement existants de raisonnement et d'inférences ne sont pas suffisamment matures pour être efficacement utilisés dans le cadre d'un artefact ontologique modulaire et distribué, bien que la notion de modélisation d'un domaine à travers un tel type d'artefact soit un domaine en cours de développement.

Le deuxième obstacle rencontré concerne la procédure de modélisation réelle de l'artefact ontologique. Pour le surmonter, des processus de modélisation sont étudiés et identifiés. Les pistes étudiées sont, soit un concepteur unique et central de l'ontologie, soit une équipe de concepteurs répartis entre diverses entreprises. En cas de concepteurs multiples, il est important qu'un processus particulier permette de gérer ces acteurs distribués.

Après ces deux questions fondamentales de recherche sur l'ontologie du web sémantique, nous identifions dans ce chapitre le thème de cette thèse : ***l'utilité de l'ontologie du web sémantique comme étant la conception d'un artefact d'intégration pour les informations et les ressources physiques distribuées de la chaîne logistique***. Ce thème proposé englobe deux idées de base, la première est liée à l'intégration des artefacts informationnels générés dans l'entreprise distribuée, la seconde est la gestion de l'utilité des ressources physiques de fabrication de l'entreprise distribuée.

Pour le premier thème concernant l'intégration de l'information, dans la revue de la littérature qui est présentée dans les sections II.5.1 et II.5.2, nous avons discuté des exigences de l'intégration de l'ontologie et du contenu. Celles-ci concernent, d'une part l'identification du modèle du contenu ou de l'artefact d'information généré au sein de l'entreprise distribuée et d'autre part, l'identification d'un processus collaboratif permettant d'intégrer l'ontologie modélisée avec l'artefact d'information généré. Pour cela, nous proposons une architecture

conceptuelle avec des composants spécifiques qui peuvent gérer cette ontologie ainsi que l'intégration des artefacts d'informations. Dans la section III.6, nous présentons les fonctionnalités de ces composants, ainsi que la manière dont ils interagissent pour atteindre cet objectif d'intégration.

Pour le deuxième thème concernant la gestion de l'utilisation des ressources physiques de fabrication de l'entreprise distribuée à l'aide d'ontologie du web sémantique, nous discuterons d'abord, en section III.2, de la faiblesse actuelle de la flexibilité des systèmes logistiques pour répondre aux exigences particulières de la chaîne logistique. L'importance de cette partie est qu'elle identifie les besoins actuels de la chaîne logistique ainsi qu'une alternative aux pratiques conventionnelles. Cette alternative peut être utilisée pour l'intégration des ressources distribuées dans les petites et moyennes entreprises. La section III.3 de ce chapitre traite de la représentation logique des ressources physiques de la chaîne logistique au sein de l'artefact ontologique dans le cadre de l'approche proposée. Ce parallèle entre les représentations physiques et logiques des ressources est très important, car cela permet la configuration des ressources réseau. Dans la section III.4, nous proposons un modèle de représentation des entités de la chaîne logistique. Dans la section III.5, un modèle de système de gestion des connaissances est présenté, afin de fournir des bases solides pour notre approche sur l'utilisation des artefacts ontologiques comme des artefacts de configuration pour les ressources distribuées de la chaîne logistique. Enfin, il est nécessaire de modéliser des ressources et des entités de processus selon la sémantique comprise par la norme ISA-95 pour qu'elle soit réelle. Pour cela nous proposons un modèle UML basé sur la norme ISA-95 qui est discuté dans la section III.7.

III.2 Entreprise virtuelle et réseau logistique

Un réseau logistique se compose d'entreprises ayant des relations et où les partenaires ont généralement un accès limité à chacune des informations. De ce fait, il peut aider à la planification et à l'ordonnancement d'une production collaborative ou au développement de produits.

L'entreprise virtuelle (EV) est similaire aux concepts de chaîne/réseau logistique, mais avec un ensemble de relations très courtes dans la durée, car elles sont établies dans un but ou un projet particulier. Cette relation se termine avec l'achèvement du contrat.

L'établissement d'une EV demande l'identification des capacités commerciales fournies par les entreprises partenaires, au sein d'une fenêtre temporelle. Constituer une EV permet d'exploiter des opportunités de marché pourtant généralement limitées. La modélisation des opérations d'entreprise basée sur le processus est un moyen très efficace pour aider à l'intégration des entreprises. Ces modèles de processus décrivent à la fois la fonctionnalité et la dynamique de l'entreprise et permettent d'identifier toutes les informations produites (par exemple : les commandes, les pièces, les produits, les ressources, les aspects opérationnels, etc.). Entre autres, les modèles de processus d'affaires décrivent les fonctionnalités et les comportements opérationnels (dynamiques) de l'entreprise. Ils permettent d'identifier les activités opérationnelles, ainsi que leurs informations telles que les produits, les commandes, les processus, les ressources et enfin leurs relations avec l'environnement interne et externe.

Aujourd'hui, l'intégration des informations et des activités entre les différents fournisseurs et clients est devenue nécessaire, car les chaînes logistiques modernes sont constituées d'un nombre croissant d'entreprises (Akkermans H.A. *et al.*, 2003). De plus, l'évolution rapide des exigences des clients impose une modification drastique de la topologie de la chaîne logistique pour répondre, au mieux, aux désirs du client.

Il est important d'établir une distinction entre la chaîne logistique et le réseau logistique. La chaîne d'approvisionnement classique se compose de plusieurs entreprises impliquées dans la fabrication, l'assemblage et l'emballage de produits qui sont fabriqués en son sein. L'hypothèse commune au sujet des membres de la chaîne logistique est que chaque partenaire a une capacité opérationnelle suffisante pour réaliser la partie du processus général qui lui incombe en un lieu de production dédié. *A contrario*, dans le cas du réseau logistique, également constitué de multiples entreprises, les capacités opérationnelles individuelles ne sont parfois pas suffisantes pour réaliser la partie du processus général qui leur incombe. Par conséquent, ces entreprises doivent collaborer et partager leurs capacités de production à plusieurs endroits et moments du processus de fabrication

Il existe de nombreux articles traitants de la flexibilité, mais la plus grande partie concerne la flexibilité des systèmes de production. Ce concept de flexibilité dans le réseau logistique varie de la flexibilité simple de niveau opérationnel, à une flexibilité de niveau plus stratégique. De plus, une seule entreprise peut avoir de multiples typologies de relations avec ses partenaires, telles que le CPFR (Collaborative Planning, Forecasting and Replenishment), le VMI (Vendor Managed Inventory) ou une relation plus classique entre vendeur et acheteur. Ainsi, la

capacité à s'engager et à se désengager des interactions d'une collaboration est d'une importance critique pour la gestion du réseau logistique.

Les processus d'interaction de l'entreprise avec, par exemple, les produits, les pièces, les documents et les informations et leurs procédés nécessitent des ressources qui doivent être contrôlées par des règles de gestion et des contraintes organisationnelles. Notre approche se concentre sur l'utilité des ontologies sémantiques web pour représenter les ressources physiques dans un réseau distribué d'entreprises et fournit un mécanisme pour l'intégration des ressources de production et d'approvisionnement. Elle permet ainsi une aide à la décision pour le management de ces ressources distribuées dans une entreprise virtuelle. Elle sert donc de support décisionnel dans la gestion des ressources distribuées et permet aussi d'accéder à des ressources sous-utilisées de l'entreprise virtuelle. Notre approche de la flexibilité du réseau se concentre sur la flexibilité de l'utilisation des ressources physiques et matérielles. Un accent est mis sur la flexibilité du réseau et constituera une incitation pour les PME à intégrer leurs ressources distribuées afin d'améliorer la flexibilité de livraison du réseau logistique.

III.3 Réseau logistique dans une perspective système

L'environnement en constante évolution et de plus grandes exigences de flexibilité ont obligé les entreprises à relever un défi sans précédent : remplacer leur structure hiérarchique traditionnelle d'organisation avec un contrôle central par une structure plus souple et décentralisée afin d'ajuster leur degré d'autonomie (la reconfiguration est un arrangement de parties ou d'éléments qui confèrent à l'ensemble sa forme et ses caractéristiques). Les systèmes reconfigurables sont des systèmes spécialisés représentant les produits, les processus et les structures organisationnelles d'une entreprise qui se combinent pour donner un système de systèmes. Celui-ci remplit la fonction d'un système intégré pour l'ensemble du produit et de son cycle de vie. Le réseau logistique représente un système intégré qui se compose d'entités telles que les fournisseurs, les fabricants, les entrepôts, les distributeurs et les détaillants qui, à travers des plans coordonnés et d'activités, développent des produits par la conversion de matières premières en produits finaux. Les activités de coordination définissent la relation entre ces membres parmi lesquels les flux physiques et/ou informationnels sont partagés sur la base d'objectifs communs.

Le terme « Supply Chain Management » (SCM) définit (Harland C.M., 1996) : premièrement la chaîne logistique interne qui intègre des fonctions impliquées dans les flux de matières et

d'informations du début à la fin du processus de l'entreprise ; deuxièmement, la relation dyadique entre deux entreprises en relation directe ; troisièmement la gestion d'une chaîne d'entreprises, comprenant un fournisseur, le fournisseur d'un fournisseur, un client et le client d'un client et ainsi de suite ; et enfin la gestion d'un réseau d'entreprises interconnectées impliquées dans la fourniture ultime de lots de produits et/ou de services requis par l'utilisateur final. La Figure 16 représente le modèle de l'offre générique de la chaîne.

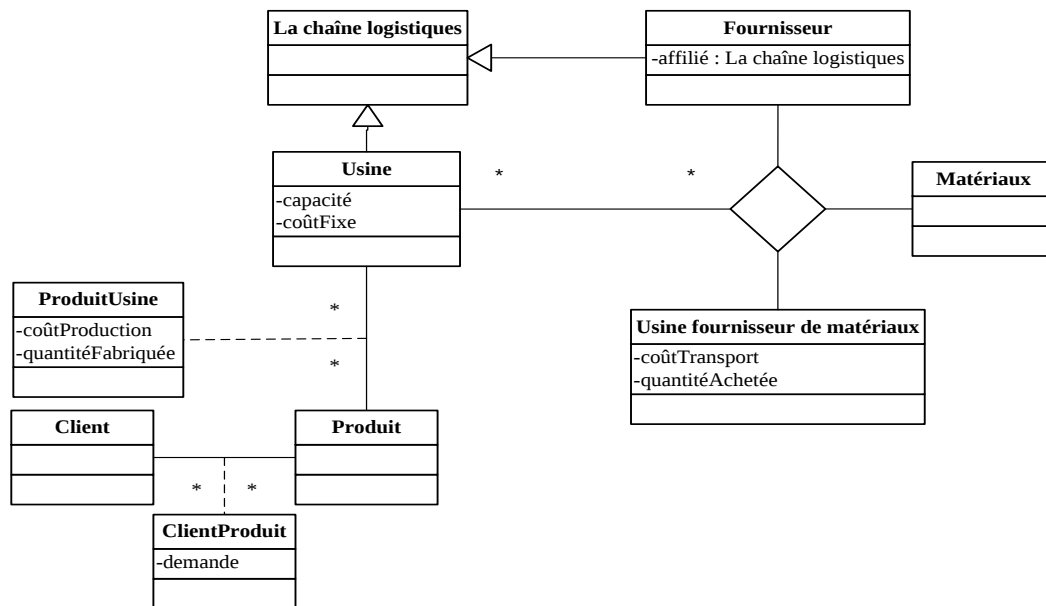


Figure 16 : Modèle objet générique de la chaîne logistique

III.3.1 Réseau logistique comme un système configurable

Formellement, le système (ensemble) peut être défini comme un assemblage de sous-systèmes (les parties), de mécanismes de production et de collaboration et de ressources destinées à remplir une série de tâches pour satisfaire aux exigences fonctionnelles. Dans les systèmes configurables, les parties définissent les systèmes physiques et logiques et l'ensemble donne sa forme, sa structure, son organisation. Les relations sont définies entre les composants du système et peuvent être à la fois internes entre les éléments des systèmes et externes avec l'environnement du système.

On peut classer les modèles de configuration en trois types : le modèle macro qui décrit le comportement de l'ensemble du système en mettant l'accent sur la prise de décision stratégique. Les modèles au niveau micro qui sont conçus pour étudier le comportement des entités individuelles au sein du système. Ces modèles relèvent d'un domaine spécifique et visent à modéliser des problèmes spécifiques comme la configuration d'atelier. Le troisième

modèle est un modèle de coordination qui, généralement, vise à coordonner l'interaction entre les niveaux macro et micro.

La reconfiguration du système affecte les caractéristiques de sa structure ou de son organisation, caractéristiques principalement constituées par les produits, les systèmes, les processus et les ressources.

Nous en concluons qu'une intégration efficace de l'information et des processus est un préalable nécessaire à l'obtention de la configurabilité d'un réseau logistique. Mais l'incertitude de la demande client et le cycle de vie court du produit induit des obstacles à l'adaptation des pratiques d'intégration les plus efficaces. Ainsi, les techniques d'intégration flexibles sont une nécessité. Le développement de la technologie augmente la flexibilité des opérations de fabrication et de service et simplifie les aspects techniques de l'intégration de la chaîne logistique, ce qui est un catalyseur important de la collaboration efficace au sein de celle-ci. Cela permet aux entreprises de quitter plus facilement leurs partenaires actuels de la chaîne logistique et de poursuivre leur participation à d'autres chaînes logistiques plus lucratives.

III.3.2 Questions clés dans les systèmes configurables

Les principaux problèmes rencontrés dans les systèmes configurables sont les structures de réseau complexes. La structure d'un système configurable devient très complexe, surtout lorsque le sous-système (usines ou installations) a un niveau élevé d'interconnexion (par exemple, les flux de matières au travers de processus multiples interdépendants ce qui se produit au sein de l'usine ou de l'installation). La variation du système au cours du temps implique que la configuration peut aussi changer au cours du temps, car nous avons affaire à des systèmes dynamiques. Comme les produits sont conçus pour offrir de bonnes caractéristiques, il devient évident qu'une seule entreprise peut trouver des difficultés à les fournir seule. En conséquence, les entreprises sont obligées d'entrer dans une alliance stratégique (entreprise virtuelle). L'externalisation et l'approvisionnement sont les stratégies qui amènent des produits à moindre coût et augmentent la maîtrise des délais de livraison, mais un réalignement supplémentaire de la coordination ou de la commande est exigé pour l'assemblage du produit en temps opportun. Les technologies de l'information et le système d'aide à la décision aident à l'intégration des entreprises et permettent, ainsi, d'avoir une

meilleure politique décisionnelle par un alignement des décisions entre les différentes entités du système.

III.3.3 Les Réseaux Logistiques configurables

Le réseau logistique configurable est donc un réseau d'entités entreprise indépendantes qui possèdent la souplesse nécessaire pour modifier leur structure avec une relative facilité, sans perdre leur efficacité opérationnelle. Les réseaux logistiques configurables basent leur prise de décision sur un système de décision unique et partagé.

Les principaux aspects qui sont les préoccupations majeures dans la prise de décision sur la configuration du réseau logistique sont : sa surveillance et sa mise à jour ainsi que son système physique et son infrastructure d'information. L'ensemble permet d'avoir une intégration transparente et rapide des flux de produits dans les processus décisionnels en relevant les informations au sein de la chaîne logistique.

L'avantage offert par la configurabilité du réseau logistique est sa plus grande souplesse face aux exigences changeantes des clients grâce à la sélection de fournisseurs moins chers et/ou plus rapides dans les circuits d'approvisionnement avec des volumes de produits précis.

Un des obstacles dans les réseaux logistiques configurables est le manque de confiance entre les partenaires de la chaîne logistique, car la période de collaboration est limitée et n'est pas un partenariat stratégique à long terme. Ainsi, la standardisation des produits et des approches de base de connaissances ne fournissant que des informations utiles et pertinentes sont, parmi d'autres, des solutions viables pour les périodes de collaboration à court et moyen terme.

La question de la gestion des processus dans le réseau logistique est l'intégration d'informations entre ses entités. Pour atteindre cet objectif, les ressources d'information de la chaîne logistique doivent être organisées efficacement et partagées. L'intégration de l'information offre des canaux qui transmettent des informations d'un constituant du réseau logistique à un autre. Une autre forme d'intégration est de travailler collectivement sur un problème commun, en partageant la compréhension du problème et en appliquant les meilleures pratiques.

III.3.4 Approche pour l'intégration des ressources distribuées du réseau logistique

La principale proposition de cette thèse est la modélisation et l'intégration subséquente du réseau logistique via les ontologies du web sémantique. Cela permettra d'obtenir une architecture web capable d'aider à la décision au niveau de la configuration du réseau logistique. Cette approche est déclinée en huit étapes qui sont détaillées ci-dessous. Notons que ces étapes ne sont pas obligatoirement séquentielles, mais qu'elles peuvent se dérouler en parallèle pour certaines d'entre elles :

- La première étape est l'identification de l'architecture appropriée de l'artefact ontologique, c'est-à-dire un artefact ontologique central pour tous les partenaires distribués. Pour répondre à cela, les capacités de raisonnement et d'inférence de la langue ontologique ont été étudiées et déjà présentées dans le Chapitre II. Les différentes approches architecturales pour le raisonnement ontologique sont identifiées dans la littérature scientifique. Basé sur le raisonnement et les capacités d'inférence, nous proposons un artefact ontologique central et unique, qui est créé par la coordination des différents partenaires collaborant. Le raisonnement distribué et les services d'inférence correspondants ne sont pas suffisamment matures. Donc l'hypothèse de recherche d'une inférence ontologique locale et de la transmission des résultats correspondants aux agents distribués n'est pas encore possible dans une mise en œuvre pratique.
- La deuxième étape est l'identification du processus correspondant qui peut intégrer la connaissance des entités distribuées dans un artefact central. Cette décision est influencée par les pratiques actuelles en modélisation d'ontologie dans un environnement collaboratif. Ce processus engendre des activités de toutes les entités collaboratrices. Pour résoudre ce problème, les activités du processus et les éléments d'architecture correspondants sont discutés en III.6. Ces composants architecturaux aident à la rationalisation de la procédure de modélisation de l'ontologie entre les différents partenaires lors de l'initialisation de l'entreprise virtuelle. Ce concept de processus collaboratif permet une meilleure représentation sémantique par rapport à l'algorithme automatique pour l'intégration d'ontologie discuté dans le Chapitre II
- La troisième étape est la modélisation des ressources du réseau logistique dans un artefact ontologique selon la sémantique généralement comprise dans le domaine de l'industrie. Pour cela, le choix d'une norme industrielle qui représente les ressources

de l'entreprise est nécessaire, ISA-95 précurseur de la CEI 62264 est choisie et a été discutée dans II-6. Cette norme d'intégration industrielle identifie les modèles de données et les attributs appropriés pour les entités systèmes industrielles, leur intégration et leur contrôle. Cette approche, de l'utilité d'un modèle de données standardisées, soulage l'inconfort de l'intégration entre les partenaires distribués qui forment l'entreprise virtuelle pour des projets spécifiques, et réduit ainsi le temps de mise en œuvre et le coût de la démarche d'intégration. Ici, il faut noter que le processus présenté à l'étape 2 (pour la modélisation de l'ontologie) ne rentre pas en conflit avec cette étape. En effet ce processus identifie l'importance de la collaboration au niveau de la phase initiale de l'établissement de l'entreprise virtuelle. Si une entreprise partenaire ne se conforme pas à la norme d'intégration industrielle, les différences peuvent être atténuées durant cette phase initiale de modélisation.

- La quatrième étape est l'identification du modèle du réseau logistique, qui représente les points d'intégration entre les différentes entités. Dans le réseau logistique, le point d'intégration est toujours le produit qui est fabriqué. Cette étape nécessite une compréhension complète et un examen de la norme industrielle qui représente les ressources de l'entreprise (ici ISA-95). Après avoir examiné les exigences de la flexibilité, la configurabilité et les contraintes de temps de mise en œuvre, nous proposons un modèle de réseau logistique qui est discuté dans la section III.4.
- La cinquième étape est l'identification d'un modèle général de système de gestion des connaissances. Comme l'artefact ontologique est un artefact logique et que des règles de gestion sont appliquées sur celui-ci pour la configuration du réseau logistique, cela nécessite l'identification d'un modèle général pour la base de connaissances. Cela est nécessaire pour maintenir la cohérence de cette approche avec les méthodes actuelles acceptées au sein de la littérature. Ce modèle de système de gestion des connaissances est abordé en III.5.
- La sixième étape de l'intégration des ressources de l'entreprise virtuelle basée sur l'artefact ontologique est le choix d'un langage d'ontologie qui a les capacités logiques appropriées de représentation, ainsi qu'un soutien suffisant de raisonnement et d'inférence. Dans les sections IV.3 et Annexe - 4 sont présentées les différentes capacités logiques et de raisonnement de la représentation du langage d'ontologie du web (OWL-DL).

- La septième étape est l'identification du modèle d'intégration pour les ressources de l'entreprise virtuelle dans le serveur central de l'artéfact ontologique. La contrainte de base de ce modèle d'intégration est la possibilité d'opérer avec des protocoles web.
- La huitième étape est l'identification d'un ensemble d'outils permettant de gérer l'intégration des ressources proposée.

Enfin, pour valider cette approche d'intégration, une étude de cas est traitée afin d'identifier les difficultés dans la procédure de mise en œuvre ainsi que ses limites. L'utilité finale d'une telle approche n'est pas seulement de fournir une intégration des artéfacts informationnels dans le cadre de l'entreprise virtuelle, mais aussi d'aider à l'exploitation de la flexibilité du réseau en termes de matériels et ressources de production. Cette approche ontologique peut aussi faciliter des approches de simulations du réseau d'approvisionnement puisque les techniques actuelles nécessitent aussi une identification de modèles de données complets, ce qui est déjà réalisé dans l'artéfact ontologique. Il est possible de trouver plus de détail sur la simulation des réseaux logistiques dans (Stefanovic D. *et al.*, 2009).

III.3.5 Conclusion

Sur cette base de discussion, il est clair que les outils, actuellement disponibles pour la collaboration dans l'entreprise virtuelle, ne sont pas suffisants. La collaboration n'est pas seulement limitée à l'intégration des processus, mais devrait également inclure l'intégration de l'information. L'exigence d'intégration de l'information dans l'entreprise virtuelle est plus complexe que dans la collaboration à long terme conventionnelle, en raison de la limitation du partenariat et du partage d'informations à court terme. La période de la coordination entre ces entreprises étant courte, de nouvelles approches doivent permettre l'intégration des informations et des processus entre des partenaires devant collaborer rapidement. Ces approches sont alors beaucoup plus appropriées. Nous proposons donc une intégration d'entreprise basée sur l'ontologie pour l'intégration des processus et l'intégration informationnelle.

La plupart des travaux en cours sur l'intégration d'entreprise, basée sur une ontologie, ne fournissent pas une vue d'ensemble des différentes capacités qui peuvent être fournies par l'ontologie. Ils n'ont pas répondu à la question sur l'exigence structurelle de l'ontologie et le processus correspondant nécessaire pour mettre en œuvre une certaine capacité opérationnelle.

Du point de vue du web sémantique, les ontologies ont été utilisées comme un outil de gestion de la connaissance qui ajoute une vue logique à une vue informationnelle, mais aucun travail n'est effectué dans la perspective de l'entreprise. Aucune étude n'est réalisée pour répondre à la question de l'ontologie centrale, distribuée ou modulaire (perspective du web sémantique).

III.4 Modélisation du réseau logistique

III.4.1 Considérations générales

Le réseau logistique offre aux entreprises une nouvelle manière de planifier, organiser, gérer, mesurer la production et livrer des biens ou des services. Dans ce nouveau paradigme industriel, les entreprises agissent comme parties du réseau et non plus comme des entités isolées. La clé pour une gestion efficace du réseau logistique réside dans le comblement de l'écart entre les changements au jour le jour arrivant dans chacune des entités et l'habilité à les identifier et y répondre. Actuellement, l'idée du réseau logistique remplace de plus en plus le concept de la chaîne logistique (Harland C.M., 1996), car les chaînes logistiques sont devenues de plus en plus complexes et entrelacées, par exemple, les processus de production sont devenus de plus en plus dépendant de fournisseurs, de chaînes d'assemblages multiples, etc. Considérant les choses de manière moins linéaire et de façon plus « toile », les réseaux logistiques regroupent les activités de la matière première jusqu'à la vente de détail en passant par la production, le stockage, la distribution et le transport qui impliquent de nombreux participants. À chaque nœud (entité) du réseau, différents types de processus sont réalisés (achat, production, assemblage, stockage, ventes, etc.), mais ces processus sont interdépendants et interconnectés les uns aux autres via les flux d'informations, de matériels et financiers.

Beaucoup des approches de modélisation multi-agents des chaînes logistiques sont basées sur des objets abstraits (agents ou modules) qui représentent les entités industrielles (machines, humains, entreprises, etc.) et qui réalisent des fonctions calquées sur les stratégies industrielles. Cependant, ces approches ne peuvent inclure toutes les entités industrielles ni toutes les politiques associées qui existent dans les réseaux logistiques réels (principalement en raison des difficultés de modélisation et de la très grande variété des modèles réels existants) (Puigjaner and Laínez, 2008). *A contrario*, les approches processus sont plus génériques, car elles utilisent les processus industriels comme éléments de base. De cette façon, les processus inter-reliés du réseau logistique peuvent être modélisés plus facilement et plus précisément (De la Fuente M. *et al.*, 2010).

Les approches processus sont donc les approches dominantes pour la modélisation du réseau logistique. Ces approches permettent une décomposition de processus complexes en des processus plus simples avec des unités logiques appropriées qui explicitent leurs opérations internes, leurs entrées, leurs sorties, leurs attributs et leurs relations avec les autres processus adjacents. Ces approches sont aussi utilisées par la communauté du management des risques, car cela permet de relier les risques aux processus. Cela permet aussi d'identifier pour chaque opération de processus les travaux à remplir et les critères de leur évaluation, les ressources nécessaires ainsi que les contraintes potentielles à respecter. Comme dans un réseau logistique, les différents processus sont fortement reliés, une modélisation axée processus apparaît donc comme bien adaptée à la modélisation des réseaux logistiques.

L'approche pour modéliser un réseau logistique est, premièrement, de modéliser le nombre d'entités (fournisseurs, fabricants, clients) puis deuxièmement, de modéliser les processus réalisés dans chaque nœud du réseau et, finalement, les dépendances d'interconnexions entre les nœuds (Márque C., 2010).

Les modèles SCOR (Supply Chain Operation Reference) représentent une approche universelle du management des chaînes logistiques qui peut être appliquée à divers domaines industriels. Ce standard met en avant un couplage faible entre les acteurs du réseau logistique, servant ainsi de squelette à une modélisation processus. SCOR contient trois niveaux de détails de processus. Le niveau supérieur définit le cadre et le contenu de la chaîne logistique, le deuxième contient les intersections entre processus et définit les catégories de processus, le troisième enfin décrit les éléments de processus courant et les flux associés. Chaque élément de processus dispose d'attributs et d'identification et agit comme une entité granulaire. Ce modèle représente donc le réseau logistique (via ses nœuds), modélise les processus et les relations entre les ressources. L'activité de modélisation des processus implique à chaque nœud une sélection des processus par rapport aux types de processus, une méthode de réalisation et des métriques, puis la définition des relations entre les processus sélectionnés. Cette approche permet donc la modélisation des relations à la fois entre les processus au sein d'un même nœud, mais aussi entre différents nœuds. Dans un environnement industriel, chaque entité du réseau logistique est exprimée en terme de ressources disponibles, telles que matérielles, humaines, machines, de transport, de production ou de stockage. De même, chaque processus dans le réseau logistique a une capacité qui est exprimée en termes de quantité d'unités produit produites par période temporelle ou de lots produits par unité. Et

finally, each process is supposed to consume a certain quantity of additional materials, like energy. The industrial environment model gives two important parameters: the costs and durations of the processes. The constraints represent a set of maximum available resources for each node, this maximum being defined in a particular way for each given network.

III.4.2 Modèle

From III.4.1, we observe that to model a logistics network, it is imperative to take into account the products manufactured, the resources and the processes of the different actors and the relations of interdependence between each entity, without forgetting the different constraints linked to the processes and resources. We thus define the meta-model of the logistics network $\mathfrak{M}$ which can be defined as follows :

$$\mathfrak{M} = f [\mathfrak{L}_{S,i}, \mathfrak{L}_{M,i}, \mathfrak{P}_i, \mathfrak{R}_i, R_{\mathfrak{P}}^{\mathfrak{R}}, R_{\mathfrak{Q}}^{\mathfrak{R}}, \mathfrak{C}_{P_i}, \mathfrak{C}_{R_i}]$$

$\mathfrak{L}_{S,i}$ = Resources supplier of the network.

$\mathfrak{L}_{M,i}$ = Resources of production of the network.

$\mathfrak{P}_i$ = Types and attributes of the processes of the network.

$\mathfrak{R}_i$ = Types of products of the network.

$R_{\mathfrak{P}}^{\mathfrak{R}}$ = Relations between the products and the processes of the network.

$R_{\mathfrak{Q}}^{\mathfrak{R}}$ = Relations between the products and the resources of the network.

$\mathfrak{C}_{P_i}$ = Constraints processes.

$\mathfrak{C}_{R_i}$ = Constraints resources.

i = Index (1, 2, ..., n).

III.5 Le modèle référence du système pour la gestion des connaissances

In this section, we present a reference model for the management of knowledge. The term knowledge is used in the literature when a domain of interest is characterized and formalized by an explicit representation within an artifact of ontology. Thus, it becomes important to characterize a general model for the management of knowledge.

connaissances, en vue d'assurer la cohérence de la mise en œuvre de notre approche avec les pratiques acceptées.

La composante ontologique enrichit le modèle du domaine avec des constructions. Elle transforme ainsi une représentation abstraite du problème en un modèle de connaissance, et ce, en formulant des axiomes entre les notions appropriées dans le domaine du problème. Ces constructions sont les suivantes : (1) les axiomes, qui définissent des règles spécifiques pour le domaine du problème ; (2) les algorithmes qui fournissent, étape par étape, les procédures, pour aborder le domaine du problème et le résoudre (algorithmes tableau, par exemple) ; (3) l'engagement de modèle, qui relie les axiomes ontologiques aux valeurs de données. Les deux premières composantes sont modélisées à travers une analyse complète du domaine, c'est-à-dire l'identification des exigences du problème et la sélection du formalisme logique approprié. Une ontologie est en soi un vocabulaire avec des règles appliquées sur son utilisation. L'application au monde réel nécessite donc l'utilisation de plages de données sur cette ontologie. Les engagements ontologiques fournissent ces plages de données.

Le modèle de référence pour la construction du système de connaissance est divisé en trois parties : la première est la représentation du système (taxonomie système), la seconde sont les exigences de représentation fonctionnelles (taxonomie des problèmes) et la troisième est la représentation formelle de la connaissance (l'ontologie).

L'ontologie présente l'état du système à la fois d'un point de vue statique et dynamique ; l'ontologie présente également la réponse (ou réaction) du système. Une chaîne logistique peut être représentée comme une collection de toutes les instances possibles d'un système générique avec les relations correspondantes.

$$S = (T, W, R)$$

S = Le système

T = Ce qui symbolise les éléments du système

R = La relation entre les éléments du système

W = Les instances de classes du système (S)

La notion symbolisant les éléments du système est une notion abstraite et pour enquêter sur un élément du système, nous sommes obligés de la séparer du monde extérieur et de l'expliquer comme un objet (O_b).

$$O_b = (\{a_i, A_i\} \mid i \in N_n\}, \{(b_j, B_j) \mid j \in N_m\})$$

$N_n = \{1, 2, \dots, n\}$ est le nombre des attributs que l'objet O_b possède.

$N_m = \{1, 2, \dots, m\}$ est le nombre de la chaîne d'observation où les attributs sont examinés et recueillis.

$a_i = L$ 'attribut

$A_i =$ Ensemble d'instances d'attributs (a_i)

$b_j =$ La chaîne d'observation

$B_j =$ Ensemble des états possibles de la chaîne d'observation (b_j)

$i = L$ 'indexation des attributs

Les chaînes d'observation sont des situations, des circonstances où d'autres sources ou le problème nécessitent une recherche. Les variables sont utilisées pour une représentation opérationnelle d'un des attributs. Chaque attribut a un nom, qui est pris à partir d'un ensemble de valeurs possible A_i .

Les attributs définissent deux types de variables : générales et spécifiques, pour une utilisation dans des modèles généraux et spécifiques, respectivement. Les variables générales et spécifiques sont des éléments de trois systèmes initiaux : le système - objet, le système - problème spécifique, et le système - problème général. Les deux dernières représentations initiales du système relient les attributs de domaine à des variables observées du monde réel, qui sont classées comme des engagements ontologiques. La séparation d'objets génériques et spécifiques est comparative. Dans certaines situations, un seul modèle de problème est nécessaire, tandis que dans d'autres cas deux ou plusieurs modèles de problèmes sont nécessaires pour pallier à la complexité, en séparant le domaine des problèmes dans des modèles d'information avec différents niveaux d'abstraction. En conséquence, les modèles d'abstraction sont multiples pour l'ontologie obtenue et un mécanisme supplémentaire pour la résolution de l'hétérogénéité entre ces niveaux d'abstraction est nécessaire, par exemple une cartographie ontologique.

$$G_P = (\{vv_i, VV_i\} \mid i \in N_n\}, \{ww_j, WW_j\} \mid j \in N_m\})$$

$$S_P = (\{v_i, V_i\} \mid i \in N_n\}, \{(w_j, W_j) \mid j \in N_m\})$$

$G_P =$ Modèle de problème général

$i, j =$ L'indexation des attributs et des chaînes d'observation respectivement

$vv_i =$ Variable qui peut être attribuée aux attributs

$VV_i =$ Ensemble des valeurs possibles que la variable vv_i possède

$ww_j =$ L'état du système générique pour vv_i respectivement

$WW_j =$ Ensemble des états possibles de ww_j

$S_p =$ Modèle de problème spécifique

$v_i =$ Les variables qui peuvent être assignées à des attributs (a_i) pour le problème spécifique

$V_i =$ Ensemble des valeurs possibles que v_i peuvent posséder

$w_j =$ Les états spécifiques du système

$W_j =$ Ensemble d'états possibles w_j

La notion symbolisant les éléments du système sur un problème particulier peut être formulée dans un artefact ontologique central :

$$T = (O_b, G_p, S_p)$$

L'engagement ontologique I est l'application d'axiomes logiques conçus pour tenir compte du sens voulu du vocabulaire et l'affectation des contraintes sur les variables du système. Le modèle d'ontologie résultant peut être représenté comme :

$$O = (M, A, H)$$

$O =$ Modèle d'ontologie

$M =$ Modèle de données pour le domaine de chaîne logistique

$A =$ Ensemble d'axiomes

$H =$ Algorithmes ou heuristiques

Du point de vue de la mise en œuvre de la base de connaissances, les caractéristiques des attributs et le modèle de données sont fixés, sur la base de l'expressivité dans le langage utilisé pour représenter le modèle ontologique. Le moteur d'inférence est en outre responsable de l'application de l'algorithme correspondant ou des heuristiques pour l'affectation, l'inférence et la satisfaction de la requête. Ces systèmes d'inférence diffèrent en fonction des langages de représentation logiques.

III.5.1 Développement de composants pour les systèmes de gestion des connaissances

Le développement d'une ontologie et la mise en œuvre du modèle de référence mentionné ci-dessus comprennent la définition de l'ontologie et la portée de son utilisation. Enfin, il reste à l'assembler en utilisant un langage de calcul et à la stocker dans un environnement qui faciliterait sa diffusion et son utilisation, les 3 phases de ce processus sont :

- **La Capture** : ce qui implique l'identification des concepts clés et de leurs relations dans le domaine étudié.
- **L'Assemblage** : qui est une représentation explicite, dans un langage formel, de la conceptualisation faite lors de la phase de capture.
- **Le Stockage et l'utilisation** : il s'agit du troisième volet de l'implémentation du système fondée sur la connaissance. Cela correspond au stockage de la base de connaissances, qui est capturée et ensuite modélisée dans les deux étapes précédentes (capture et assemblage ontologique). Le stockage de la base de connaissances est important, car elle s'appuie sur le contexte de raisonnement et les capacités d'inférence que l'ontologie fournit.

III.5.2 Conclusion

L'ontologie peut être utilisée de différentes manières, mais essentiellement dans la perspective de l'entreprise, elle est utilisée comme une partie à intégrer parmi les différentes ressources ou un support décisionnel pour les différentes activités. La gestion de la chaîne logistique est un domaine où les interactions entre les sciences de la décision et l'ingénierie des systèmes d'information sont les plus profondes. L'implémentation de la décision n'est pas possible sans le support d'un système d'information et un système d'information seul, sans composants d'aide à la décision, n'est plus suffisant pour maintenir un avantage compétitif dans le contexte actuel de la chaîne logistique.

Pour le problème de la configuration de la chaîne logistique, les exigences clés pour le système d'information sont l'exécution des processus décisionnels et l'intégration des résultats de la prise de décision dans les opérations.

III.6 Approche ontologique pour la gestion de contenu

L'objectif de la gestion des connaissances dans une organisation est de favoriser le développement, la communication et la préservation des connaissances. Pour atteindre cet

objectif, une mémoire d'entreprise doit être créée. La mémoire institutionnelle est une représentation explicite et persistante des connaissances et des informations au sein d'une organisation. Une mémoire organisationnelle peut être construite selon différentes techniques. Elle peut être fondée sur la connaissance, basée sur les cas ou être distribuée entre les partenaires. Dans le cas de l'entreprise virtuelle, les approches qui facilitent l'intégration de la connaissance distribuée sont plus efficaces. « *Les ontologies sont une approche basée sur la connaissance pour la construction de la mémoire institutionnelle* » (Gómez-Pérez A. et al., 2004). Les approches de ce type sont donc le moyen le plus réalisable pour créer, maintenir et intégrer la mémoire institutionnelle dans le contexte de l'entreprise virtuelle distribuée, par exemple l'intégration du contenu des exigences opérationnelles appropriées entre les partenaires.

Comme déjà soulignées dans le Chapitre II, les ontologies peuvent être utilisées comme un outil central d'intégration pour le contenu. L'annotation de contenu basée sur l'ontologie nécessite trois composants de base. Le premier est le modèle d'ontologie d'un domaine particulier, le deuxième, les objets d'information qui sont annotés avec l'ontologie du domaine modélisée et enfin un mécanisme d'intégration entre l'objet d'information et des objets logiques (ontologie). Une instance du modèle d'objet des différentes ressources matérielles et informationnelles nécessaires au sein d'une entreprise virtuelle est montrée dans la Figure 17 et dans la section précédente.

Nous allons discuter ici de la représentation des différents segments ontologiques de l'entreprise. L'ontologie d'entreprise et les règles commerciales n'existent pas pour elles-mêmes, mais servent les processus de soutien aux entreprises. Dans des circonstances pratiques, les ontologies multiples ne fournissent pas suffisamment de capacités d'intégration. Elles ne permettent pas de distribuer le raisonnement et l'inférence pour la satisfaction de la requête qui ne peut donc être atteinte avec les ontologies distribuées ou modulaires.

III.6.1 Architecture proposée pour la création et la gestion de contenu

Comme discuté dans le Chapitre II, les exigences de collaboration pour la création de contenu sémantique ne sont pas limitées à la génération de contenu (authoring), mais concernent la restauration de la modélisation d'ontologie de domaine de collaboration (maturation d'ontologie) et le mécanisme d'intégration correspondant (annotation de contenu de l'information avec le contenu ontologique). La section ci-dessous fournit l'analyse des besoins

pour les composants de l'architecture que nous proposons et de leurs besoins en flux de données qui s'adressent à la création de contenus collaboratifs et à leur annotation avec l'artefact ontologique au sein de l'entreprise virtuelle.

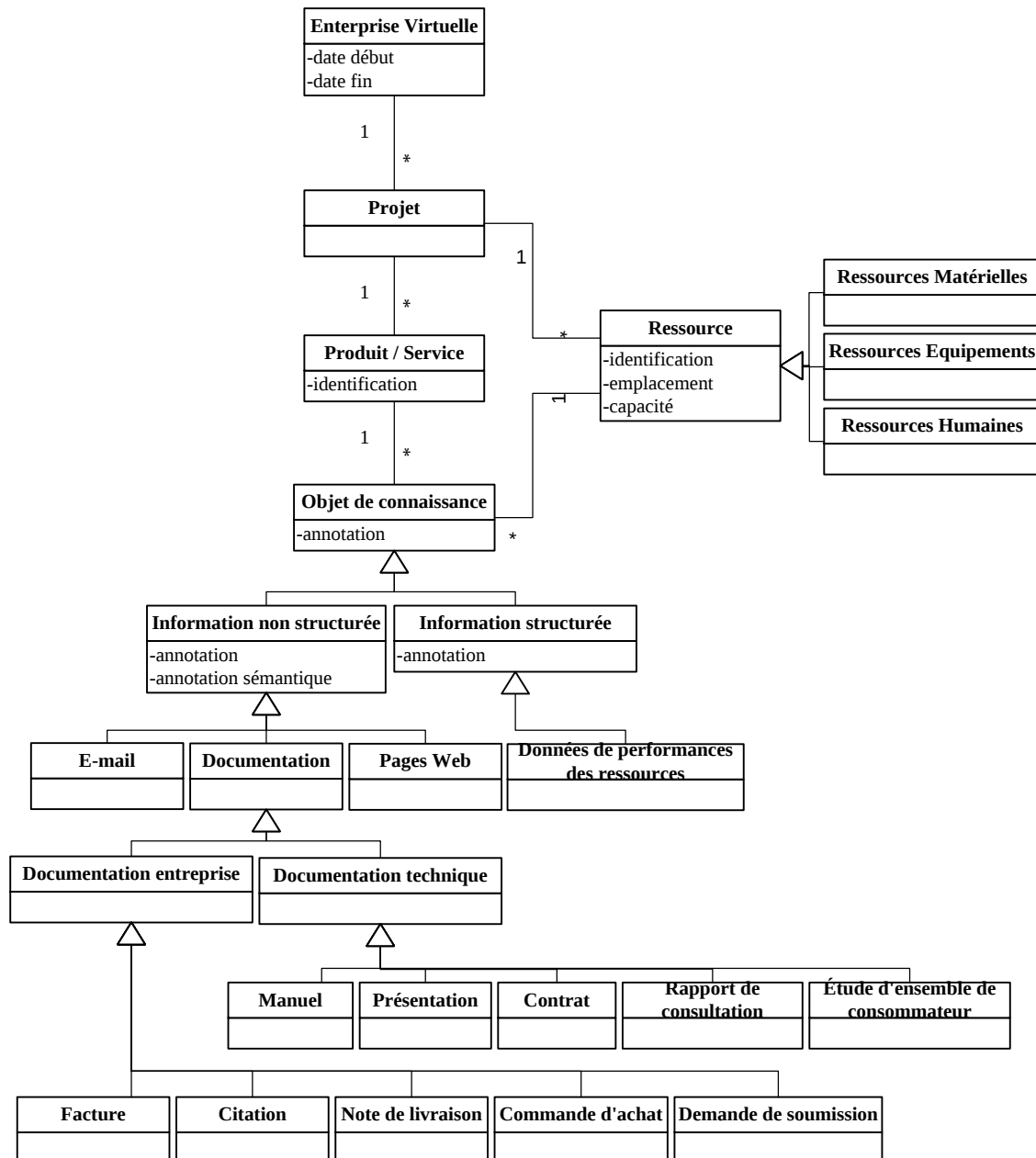


Figure 17 : Modèle objet pour l'entreprise virtuelle

L'analyse des besoins pour l'intégration du contenu et de l'artefact ontologique au sein de l'entreprise virtuelle englobe trois phases. La première est la délimitation du modèle de représentation pour la documentation ou le contenu créé au sein d'une entreprise virtuelle, comme indiqué dans la Figure 17. La deuxième phase consiste à identifier les variations des différents flux d'information, nécessaires à l'intégration des artefacts ontologiques avec le

contenu. La troisième phase consiste à identifier les éléments architecturaux nécessaires, ainsi que leurs interactions pour l'intégration ci-dessus.

III.6.2 Analyse des exigences pour l'intégration du contenu ontologique dans l'entreprise virtuelle

III.6.2.1. Le processus centré sur l'utilisateur

Du point de vue centré utilisateur, la génération d'un document sémantique et de son processus d'annotation sont généralisés dans la Figure 6 (§II.2.5). Ceux-ci pourraient être divisés en sous-processus :

1. La gestion du processus pour la génération de documents collaboratifs et leurs annotations correspondantes avec une ontologie de domaine mature (Figure 18) ;
2. La gestion du processus pour la génération de documents collaboratifs et la maturation collaborative de l'ontologie de domaine ;
3. La gestion du processus pour l'ontologie mature et le document déjà existant ; c'est une approche statique de combiner les ontologies avec le contenu. (Eriksson H., 2007).

L'objectif de base de nos recherches porte sur les systèmes de gestion des processus décrits ci-dessus, car c'est le besoin opérationnel que le concepteur rencontre lors de l'intégration du contenu. Aux étapes initiales de la mise en place de la collaboration entre les entreprises et de ce processus, il faut ajouter un mécanisme pour classer les objets d'information tandis que la collaboration est toujours en cours. Les activités du cycle de vie nécessaires à l'ontologie et à l'intégration de contenu au sein de l'entreprise virtuelle sont représentées dans la Figure 18. Les activités correspondantes nécessaires à l'intégration de contenu d'ontologies dans une entreprise virtuelle sont les suivantes :

1. Processus pour la maturation d'ontologie de domaine ;
2. Processus pour la production de nouveaux contenus ;
3. Processus pour la catégorisation du contenu existant ;
4. Processus pour relier le contenu avec l'artefact ontologique de domaine.

Toutes ces exigences de processus et des activités associées liées à l'intégration du contenu et de l'ontologie sont représentées dans la Figure 18.

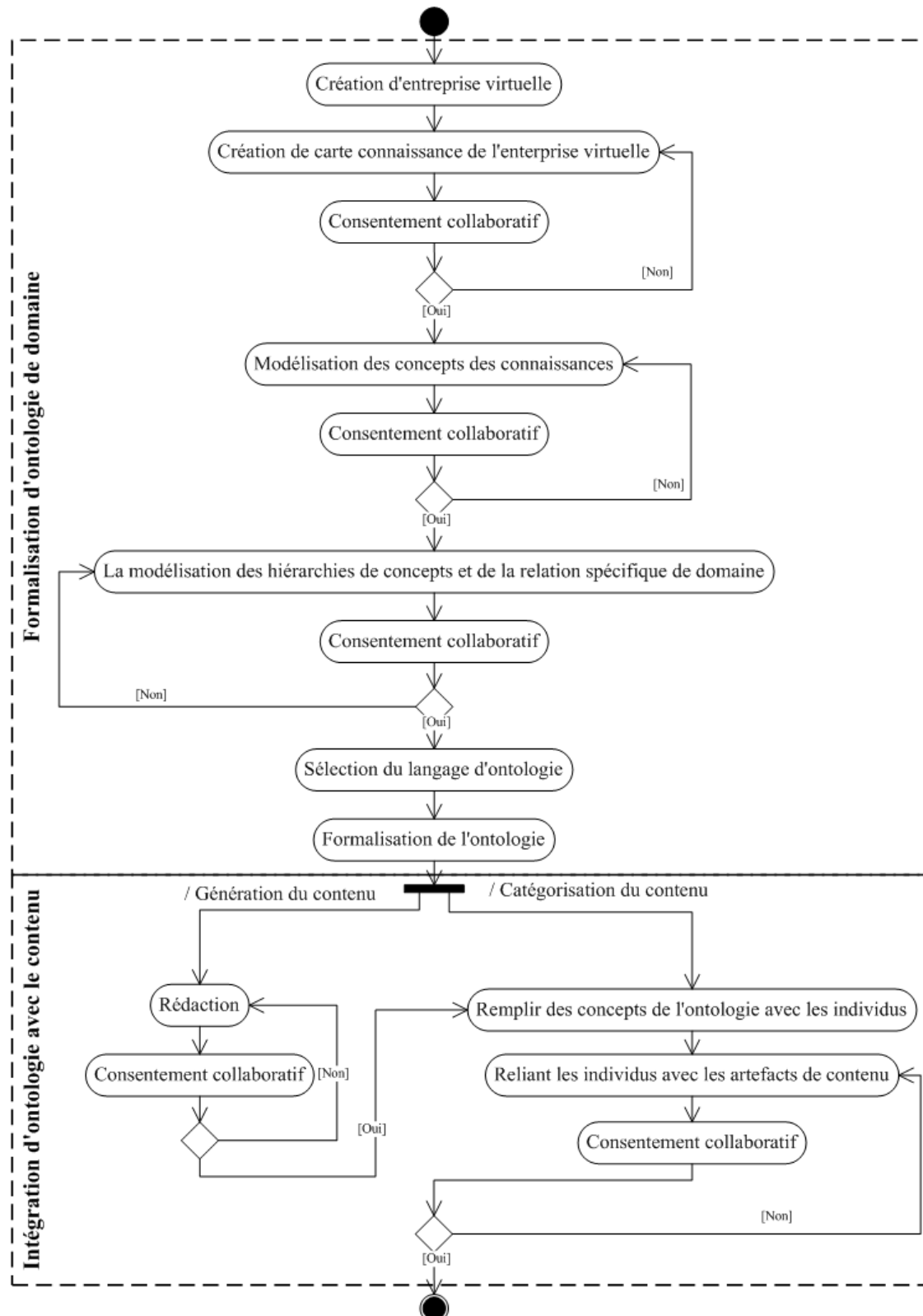


Figure 18 : Diagramme d'activité du processus pour l'ontologie de domaine mature

III.6.2.2. Les exigences d'intégration de contenu et ontologie centrée sur des documents

(Uren V. *et al.*, 2006) identifient l'exigence de l'annotation sémantique pour les gestions de contenu. Le résumé de ces besoins opérationnels en ce qui concerne l'entreprise virtuelle est

discuté ci-dessous. L'exigence mentionnée dans la référence ci-dessus est à l'identification des besoins dans le contexte du web sémantique, mais il fournit des indications précieuses sur les modèles d'intégration d'ontologies et de contenu :

1. Utiliser un format standardisé pour représenter l'ontologie réduit les problèmes d'intégration et facilite également la mise en œuvre rapide. Le format standard pour la représentation du langage d'ontologie est OWL (Ontology Web Language) (McGuinness D.L. and Van Harmelen F., 2004) et des normes pour la représentation de l'annotation se trouvent dans le schéma d'annotation RDF (Kahan J. *et al.*, 2002).
2. L'architecture de gestion des connaissances doit être un point unique de l'interface d'entrée capable de supporter les activités de collaboration puisque l'ontologie est une conceptualisation partagée d'un domaine.
3. La prise en charge de l'évolution de l'ontologie, c'est à dire de la gestion de version d'ontologie, ainsi le modèle d'intégration ontologique de contenu devrait s'adresser également à l'évolution ontologique.
4. L'évolution du document, l'obligation de maintenir la cohérence du document en constante évolution avec l'artefact ontologique au sein de l'entreprise virtuelle qui ne changent pas fréquemment tout au long de la collaboration, impose que les artefacts de contenus associés soient en constante évolution ou mis à jour.
5. De multiples approches pour l'intégration de l'artefact document avec les artefacts ontologiques sont identifiées dans le Chapitre II (Eriksson H., 2007). Ces modèles d'intégration sont : premièrement, l'incorporation d'objets actifs dans le document, ce qui entraîne un document avec un élément de représentation des connaissances comme une partie de l'objet de contenu. Deuxièmement, l'intégration des annotations des artefacts documents avec les individus concepts de l'ontologie, ce qui est mis en œuvre dans un PDFTab (cf. § II.5.2) intégrant le fichier PDF (Portable Document Format) avec l'ontologie OWL à travers l'éditeur d'ontologie Protégé. Troisièmement, le lien entre la structure du document et l'artefact ontologique, ce qui a l'avantage d'une stratégie d'intégration claire reliant les individus ontologiques et les concepts avec l'URI (Universal Resource Identifier) du document correspondant au sein du serveur d'ontologies. Ce modèle d'intégration étant, à notre avis, le plus adapté à l'entreprise virtuelle via une base de connaissances mise en œuvre par le serveur central d'ontologies, nous avons donc choisi de nous y référer pour construire notre propre architecture serveur, laquelle est détaillée en Figure 19.

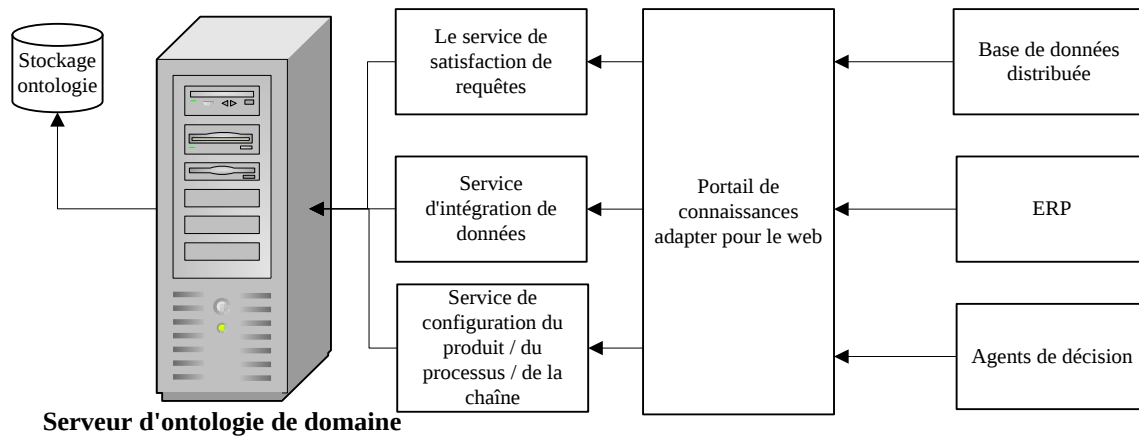


Figure 19 : Architecture du serveur d'ontologie

III.6.2.3. Analyse des besoins d'architecture

Sur la base des exigences définies ci-dessus de l'intégration ontologique des contenus et des processus associés, nous proposons les composants de notre architecture. Chacun de ces composants a une fonctionnalité spécifique. La littérature présentée précédemment identifie seulement le processus approprié pour une entreprise seule, mais dans le contexte d'une entreprise virtuelle, où des entreprises diverses doivent synchroniser leurs activités dans l'intégration ontologique des contenus, il n'y a pas encore de propositions. Ceci est la valeur ajoutée de nos travaux. Voici les caractéristiques de cette architecture et dans la section suivante ses fonctionnalités seront présentées.

1. Cette architecture devra permettre l'intégration d'une carte des connaissances descendante pour aider l'utilisateur dans la définition et la structuration de son modèle conceptuel du projet ;
2. Cette architecture devra permettre l'existence de toute une gamme d'expertises et de points de vue pour la génération de documents sémantiques et pour le sous-processus d'annotation associé ;
3. Cette architecture devra fournir un processus d'examen intégré ;
4. Cette architecture devra favoriser l'appropriation primaire de l'utilisateur de domaine et non pas les annotations de tierces parties ;
5. Cette architecture devra fournir, pour l'annotation sémantique, une interface d'entrée unique capable de fournir des fonctionnalités de collaboration.

III.6.2.4. Architecture proposée et processus collaboratif d'intégration d'ontologies et de contenus

Dans (Fahd A. *et al.*, 2009), (Fahd A. *et al.*, 2010) l'interface d'entrée unique (SPEI) comporte quatre composants principaux pour l'intégration de contenus et de l'ontologie. Les entités, comme indiqué dans la Figure 20, sont :

- Un composant pour la « conscience de groupe » (Group Awareness) (GA) qui fournit aux auteurs distribués et aux experts du domaine une « sensibilisation » sur l'état de la collaboration (collaboration awareness) sur les projets d'intégration précédents et actuels de contenus ontologiques au sein de l'entreprise virtuelle. L'objectif de ce composant est donc un champ multidimensionnel et très spécifique qui dépend du type de sensibilisation désiré dans le processus de collaboration. Nous n'en présenterons donc pas le fonctionnement interne puisque celui-ci est totalement dépendant du contexte.
- Le Référentiel des Auteurs et Experts (AER) du domaine gère les informations relatives à la communication asynchrone/synchrone parmi les créateurs de contenus distribués (auteurs) et les experts du domaine (ingénieurs de la connaissance).

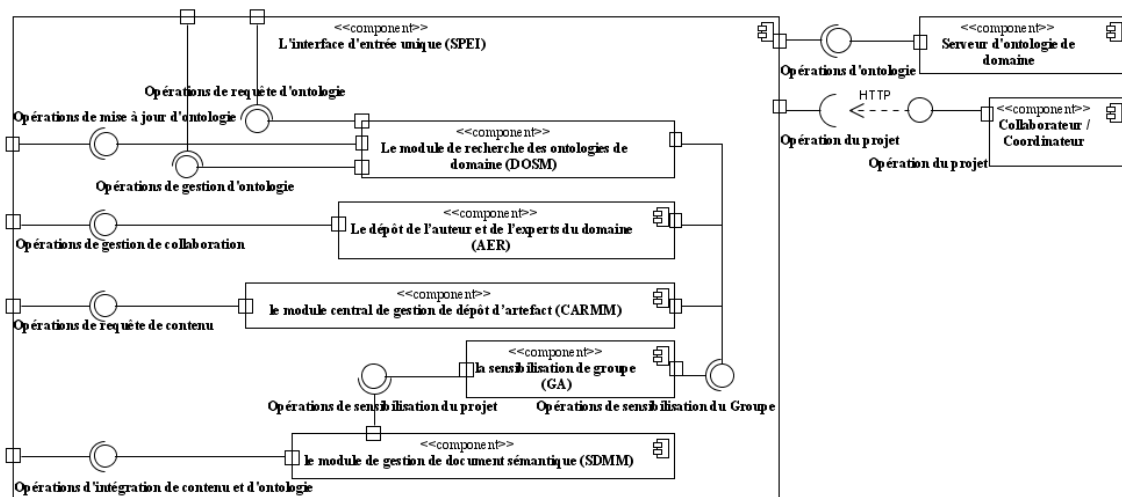


Figure 20 : Interface d'entré unique avec composants correspondants pour intégration de contenu et d'ontologie

- Le Module de Recherche Ontologies de Domaine (DOSM) fournit une interaction avec l'artefact ontologique. Cette interaction est réalisée par le biais de requêtes sur l'information représentée dans l'artefact ontologique, ainsi que par la gestion des mises à jour de l'artefact, c'est à dire le remplissage des classes définies de l'artefact ontologique avec des données d'instance. Pour cette composante, l'identification d'une

API qui gère ces opérations de mise à jour de l'ontologie, ainsi que des opérations de requête sur l'ontologie sont nécessaires. DOSM utilise les fonctionnalités fournies par l'API OWL- Protégé, qui est une bibliothèque java « open source » pour le Langage Ontologique Web (OWL) et RDF(S). Cette API fournit des classes et des méthodes pour charger et enregistrer des fichiers OWL, pour interroger et manipuler des modèles de données OWL et pour effectuer un raisonnement. En résumé, il fournit un méta niveau d'accès et des modèles de données. Le diagramme statechart en Figure 21 représente les deux processus fondamentaux de DOSM.

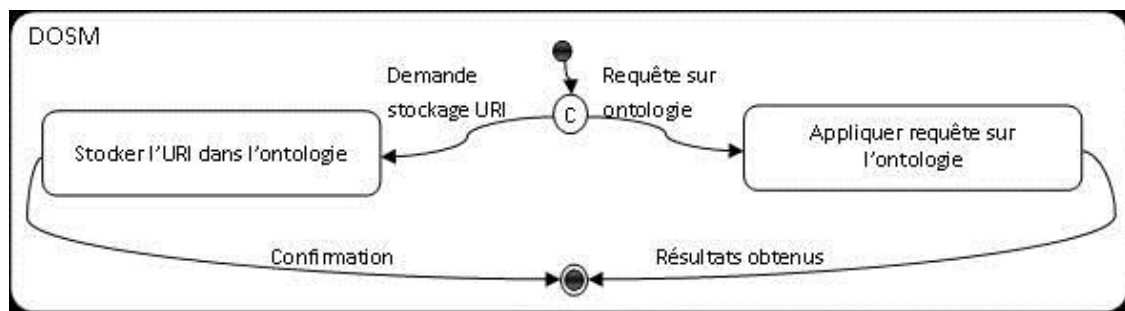


Figure 21 : Représentation Statechart des fonctionnalités de base de DOSM

- Le Module Central de Gestion de Dépôt d'Artefacts (CARMM) gère l'artefact contenu à un emplacement central, sur un serveur ontologique et gère l'URI (Universal Resource Identifier) des artefacts contenus qui sont utilisés pour le remplissage de l'artefact ontologique avec les individus de concept et avec les URI d'objets informationnels. Comme ce contenu est stocké dans un serveur central, CARMM doit donc gérer à la fois les processus de mise à jour et d'interaction avec ce contenu. Ces deux processus sont représentés dans le diagramme statechart de la Figure 24.

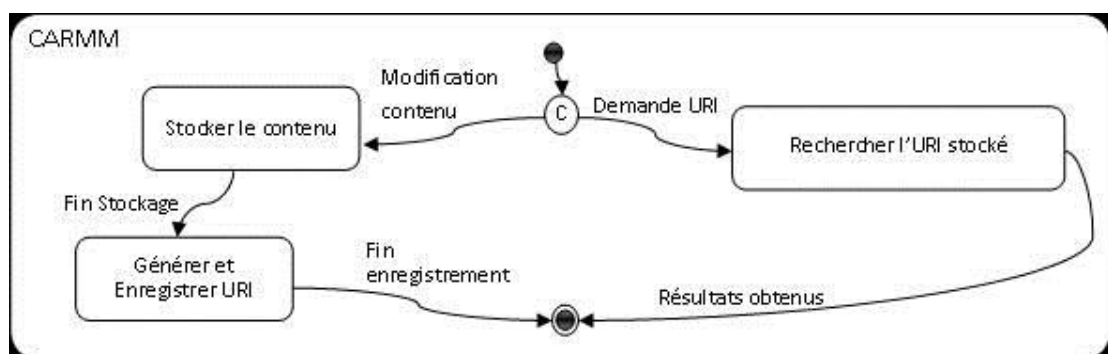


Figure 22 : Représentation Statechart des fonctionnalités de base de CARMM

- Le dernier volet de l'intégration de contenu et de l'ontologie est le Module de Gestion de Document Sémantique (SDMM) qui gère les besoins en processus spécifiques du

projet pour l'intégration de contenu ontologique de collaboration au sein de l'entreprise virtuelle. Il gère donc l'interaction entre les collaborateurs du projet grâce au composant AER. Le diagramme statechart Figure 23 identifie la fonction de base du composant SDMM.

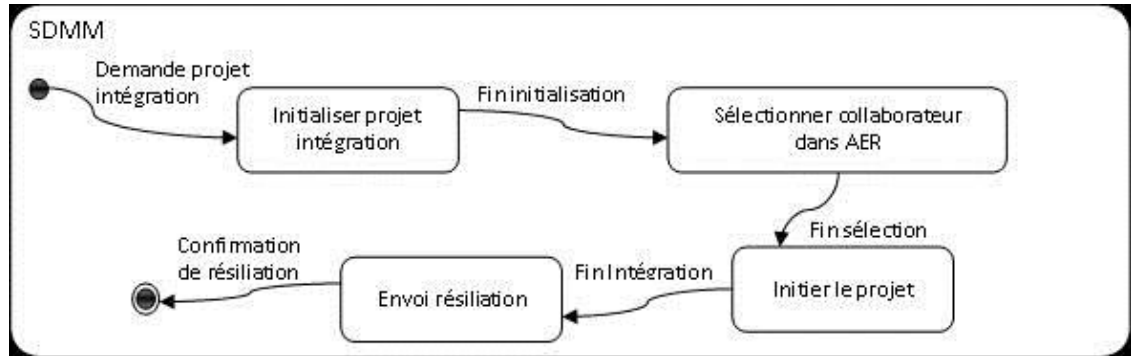


Figure 23 : Représentation Statechart des fonctionnalités de base de SDMM

La philosophie de conception de l'architecture proposée est basée sur une interface à un seul point d'entrée. Chaque objet contenu produit dans une entreprise de l'EV est transféré au dépôt d'artefact central. Le coordonnateur/initiateur du flux d'intégration de contenu ontologique collaboratif sélectionne d'abord l'artefact contenu généré récemment, cherche ensuite l'ontologie de domaine (notion d'artefact carte) et décide de la répartition préliminaire du contenu au concept de l'ontologie. Enfin, il invite les collaborateurs répartis dans l'ensemble des entreprises à formuler un consentement collaboratif grâce à une communication asynchrone facilitée par AER. Les responsabilités du coordonnateur sont :

1. **Sélection du thème** : l'ontologie concept ou sélection individuelle d'ontologie ;
2. **Sélection des collaborateurs** : pour former un consentement collaboratif. La sélection des collaborateurs est basée sur la provenance du contenu généré.

Le diagramme de séquence complet, pour initialiser un nouveau projet au sein de cette architecture est montré dans la Figure 24. Le diagramme montre également l'implication des éléments architecturaux dans l'ensemble du processus et le sous-processus correspondant à l'intégration collaborative de contenu et d'ontologie :

Séquence 1-3 : Le coordinateur se connecte dans le SPEI et cherche le contenu qui vient d'être généré au sein de l'entreprise virtuelle, puisque chacun des artefacts de contenu générés est mis à jour et est ensuite stocké dans un serveur de stockage central appelé le module central de gestion de dépôt d'artefacts (CARMM). À la suite de la connexion du

SPEI et de l'indication des objets de contenu potentiels qui nécessitent une intégration à l'artefact ontologique dans une requête simultanée à la CARMM et au serveur d'ontologie OWL, il peut identifier la différence entre les deux résultats de la requête. L'ontologie OWL permet le stockage des concepts des individus avec un format de données de type chaîne.

Séquence 3-4 : Après avoir fourni au coordinateur tous les artefacts de contenu disponibles qui exigent une intégration ontologique, le coordonnateur peut sélectionner l'artefact contenu approprié en fonction de son expertise de domaine. Cette sélection permet à la SPEI de récupérer l'URI de l'artefact de contenu sélectionné.

Séquence 5-6 : À la suite de la sélection de l'artefact contenu, le coordinateur récupère en outre les concepts ontologiques appropriés par l'intermédiaire du DOSM. L'artefact ontologique de domaine est statique et est produit dans les premières étapes de la création d'une entreprise virtuelle. Les concepts de l'objet de connaissance et leurs propriétés de type de données correspondantes sont sélectionnés et formalisés au sein de l'ontologie OWL avant que tout processus de génération de contenu des artefacts soit engagé, comme indiqué sur la Figure 24. Le DOSM utilise l'API OWL Protégé pour donner accès à des concepts formalisés du domaine ontologique, et ce, de manière individuelle.

Séquence 6-7-8 : Après la sélection des objets de contenu et par la suite du concept ontologique de domaine, le coordonnateur doit obtenir le consentement collaboratif auprès des auteurs de contenus distribués, des experts du domaine présents au sein de l'entreprise virtuelle. AER conserve les informations de communication de ces experts et SPEI envoie la requête de collaboration à ces experts par l'AER. Après une réponse positive des collaborateurs, il est possible d'initier l'intégration de contenu ontologique ou la création de documents sémantique du projet.

Séquence 9-10 : À ce moment, SPEI conserve les informations concernant le coordinateur, les collaborateurs disponibles, l'URI d'artefact contenu ainsi que le concept ontologique sélectionné par le coordonnateur pour l'attribution du concept de l'ontologie de domaine. Cette information est ensuite transmise à la SDMM qui facilite la communication synchrone et asynchrone entre le coordonnateur et ses collaborateurs par l'AER et permet d'atteindre un accord sur la provenance de l'artefact contenu dans les concepts de l'ontologie de domaine.

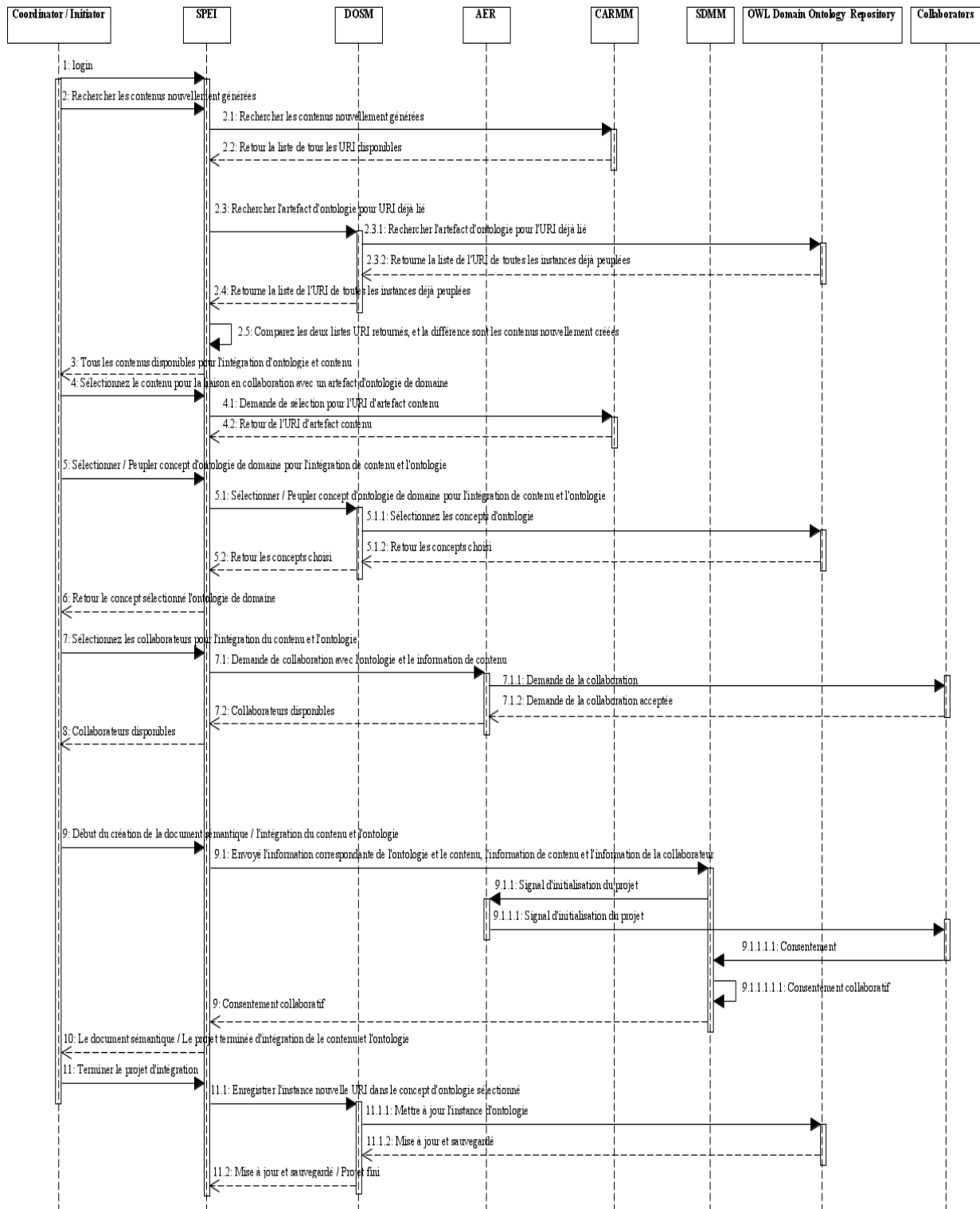


Figure 24 : Processus et diagramme séquence d'interaction des éléments architecturaux pour l'intégration de contenu et d'ontologie

Séquence 11 : Une fois le consentement collaboratif atteint par les ingénieurs de connaissances distribuées, le SDMM rend le contrôle à SPEI et communique la fin de la

collaboration. Ensuite, le coordonnateur peut permettre à SPEI de sauver dans les artefacts ontologiques de domaine l'URI des artefacts de contenu et de l'individu à l'aide de DOSM.

Un des composants supplémentaires, non représentés sur le diagramme de séquence, est la conscience de groupe (GA). La responsabilité principale de ce composant est de surveiller tout le processus d'intégration simultanée de l'ontologie dans le SPEI et de fournir à chaque coordonnateur des capacités de sensibilisation/informations supplémentaires, telles que l'histoire de chaque collaboration, de chaque collaborateur ou la sensibilisation de processus, etc.

III.6.2.5. Conclusions et discussion

Dans cette section, nous avons présenté une approche pour l'intégration de contenu avec l'artefact ontologique dans l'entreprise virtuelle, en prenant en considération des exigences collaboratives. À cet égard, nous avons présenté le modèle d'objet contenu dans la Figure 17 et les exigences de processus pour l'intégration de contenu dans l'ontologie de l'entreprise virtuelle, comme le montre la Figure 18. Ensuite, nous avons présenté le diagramme d'activité, les éléments architecturaux proposés qui aideront à réaliser le processus de collaboration et l'élaboration des similitudes. Le diagramme d'activités proposées permettra la formalisation ontologique de domaine lors de l'étape initiale de la création d'une entreprise virtuelle et la dernière partie de ce diagramme répond au processus d'intégration de contenu ontologique. Nous avons également défini le processus de la mise en place d'un serveur ontologique central. La Figure 19 montre comment, à partir de la recherche initiale, trouver les limites pratiques du raisonnement ontologique. Dans la Figure 25, nous présentons l'articulation entre les travaux présentés précédemment, les objets correspondants et les technologies pertinentes pour ces objets et le processus dont l'extrait est un artefact ontologique OWL-DL qui peut être utilisé comme un artefact d'intégration pour le contenu généré à l'intérieur de l'entreprise virtuelle.

Jusqu'à ce point, nous avons présenté nos travaux sur l'intégration de contenus ontologiques, qui sont représentés dans la Figure 17, comme étant des « objets de connaissances » pour un projet dans l'entreprise virtuelle. Dans la suite, nous présenterons l'utilisation du langage OWL-DL pour permettre la configuration des ressources d'entreprise virtuelle ou d'un artefact d'intégration. Les technologies utilisées sont présentées dans la Figure 25.

La suite du travail, basée sur ISA-95, fournira les fondements d'un système d'intégration industriel standard et ses modèles de données correspondants permettront de représenter les ressources industrielles. Nous traiterons dans le Chapitre IV une étude de cas pour valider cette architecture.

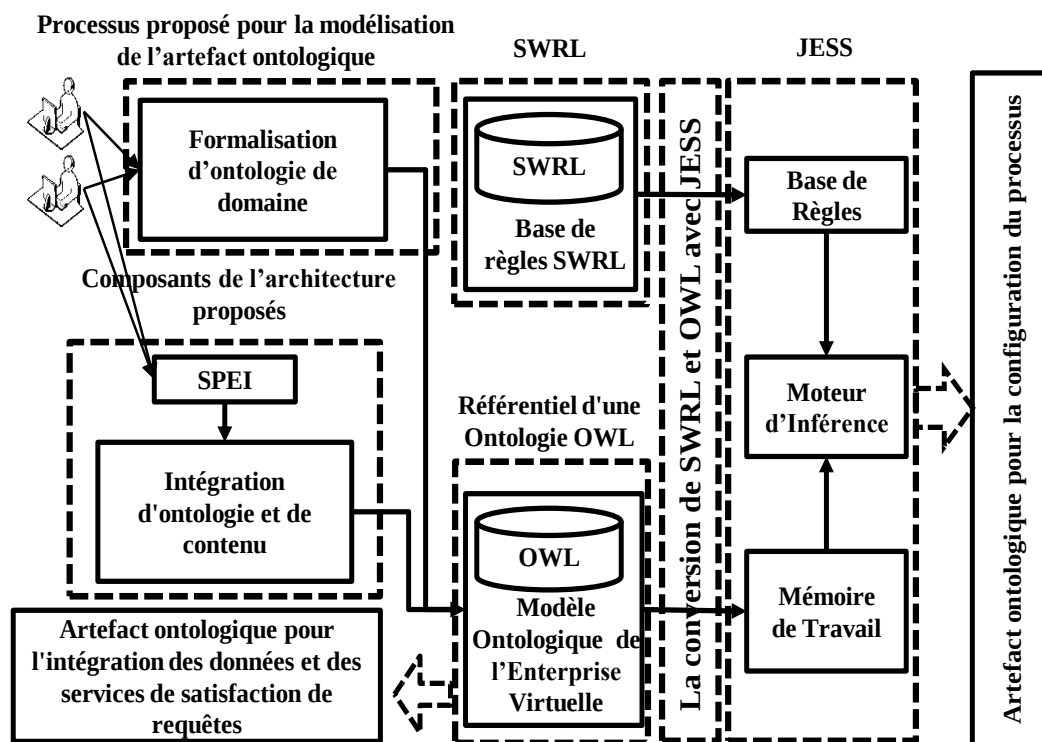


Figure 25 : Architecture de mise en œuvre et limites technologiques

III.7 Modèle UML proposé

En se basant sur la présentation du modèle d'intégration d'entreprise ISA-95 abordée dans la section II.6, les entités basiques de l'entreprise sont classifiées et un modèle UML correspondant a été construit basé sur le modèle d'un réseau logistique. L'utilité principale de ce modèle présentée en Figure 26 est de fournir un modèle de données hiérarchique unifié sur lequel l'artéfact ontologique pourra être fondé. Cette idée d'utiliser les modèles de données industriels existants peut réduire la durée d'implémentation et facilite aussi l'intégration du système de support ontologique dans un système industriel existant comme une application web autonome avec des entrées standardisées.

La relation entre les entités du modèle est basée sur le modèle du réseau logistique pour l'entreprise virtuelle qui est présentée dans la section III.4. L'importance de cette approche est déjà mentionnée dans la section III.2, c'est-à-dire la facilité et la rapidité de mise en œuvre auprès de l'entreprise distribuée. Également basées sur les exigences de l'intégration

d'informations de l'entreprise virtuelle et la configuration du réseau, des classes supplémentaires, par exemple la classe « Location » ainsi que classe « Knowledge Object », ont été ajoutés. Ces classes ne sont, en effet, pas explicitement abordées dans la norme ISA-95. En outre, nous établissons et définissons aussi la relation entre ces nouvelles entités.

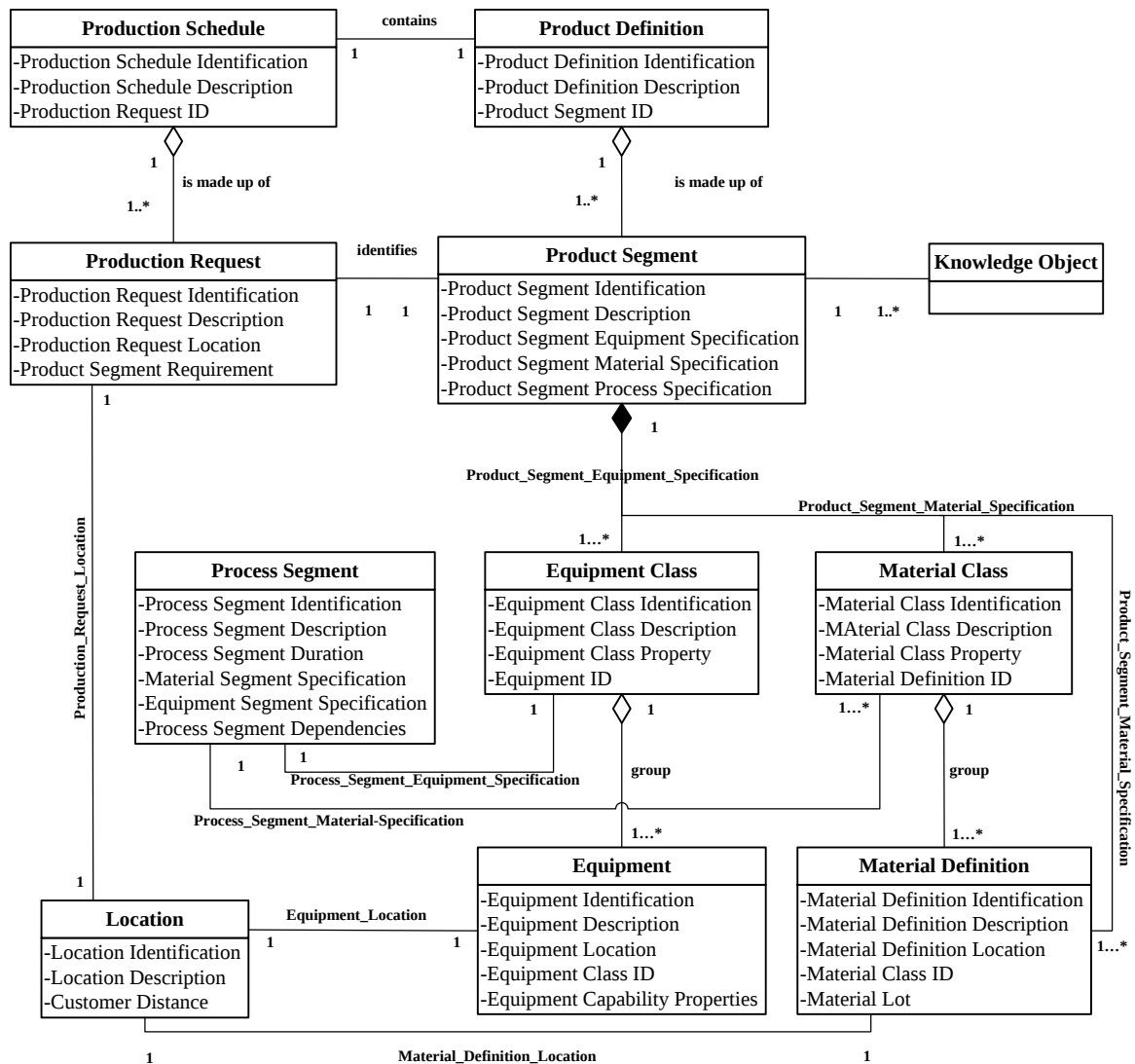


Figure 26 : Modèle UML basé sur ISA-95

Dans le modèle ISA-95, des classes sont définies et correspondent aux différentes entités opérationnelles dans une entreprise. Pour faciliter la collaboration *inter* entreprise, c'est-à-dire convertir le modèle mono entreprise en un modèle multi entreprise, un modèle unifié est construit qui peut identifier les hiérarchies ainsi que les attributs de ces entités. Ce modèle unifié est construit sur l'interprétation sémantique des modèles de classes d'ISA-95 déjà décrits en II.6. Ce modèle UML va fournir une base pour la construction de l'artéfact

ontologique écrit en OWL-DL et peut constituer un point d'extension pour de futurs objets de connaissances créés dans l'entreprise virtuelle.

Les entités de base qui représentent les ressources matérielles dans le modèle de données ISA-95 sont sous-classifiées en « Matériel Définition » et « Classe Matériel ». La *Matériel Définition* est définie comme l'entité particulière qui représente les éléments matériels individuels, par exemple un alliage d'acier peut être défini comme une entité *Matériel Définition* de la *Classe Matériel* « Acier ».

Chaque entité *Matériel Définition* contient des attributs par exemple : « Matériel Définition Identification » qui est utilisé comme une identification unique de ressources entre les différents partenaires, « Matériel Définition Description » qui décrit (ou nomme) l'entité *Matériel Définition*, « Matériel Définition Location » qui identifie la localisation de l'entité fournisseur de ce matériel dans le réseau logistique. L'attribut principal : « Classe Matériel ID » identifie les relations d'un groupe d'entités individuelles *Définition Matériel* avec une *Classe Matériel*.

En effet, chaque entité *Classe Matériel* fournit une agrégation des entités *Matériel Définition* similaires dans l'entreprise virtuelle. L'attribut « Lot Matériel » identifie une relation entre les entités *Définition Matériel* et les entités quantifiables de lot qui sont transportés entre les différents partenaires. Les attributs « Classe Matériel Identification » et « Classe Matériel Description » sont très similaires aux attributs correspondants de *Matériel Définition* et « Classe Matériel Propriété » indique les propriétés collectives de la classe telles que les spécifications de test des matériels ou de qualité au sein de l'entreprise collaborative.

De manière identique aux entités de ressources matérielles, les ressources équipements peuvent être représentées via la classe « Ressource Équipement ». Cette entité de base est subdivisée de façon similaire à l'entité « Ressource Matériel ». La *Classe Équipement* est utilisée pour représenter les ressources multiples et distribuées fournies par les différents partenaires. Chaque entité *Équipement* est identifiée via un attribut « Équipement Identification » qui associe un numéro d'identification unique à chaque entité d'*Équipement*. L'attribut « Classe Equipment ID » relie la classe *Équipement* à celui à qui l'équipement appartient. De plus, l'attribut « Équipement Localisation » identifie la localisation de l'équipement dans l'entreprise virtuelle. La *Classe Équipement* dans le modèle UML agrège les différentes entités *Équipement* en fonction de leurs attributs. Les autres attributs de la *Classe Équipement* sont les mêmes que pour la *Classe Matériel*.

La classe « Segment Processus » est utilisée pour représenter les entités processus fournies par les différentes parties. Chaque entité de cette classe est identifiée via les attributs « Identification Segment Processus » et « Description Segment Processus ». L'attribut « Durée Segment Processus » qui peut être décrit comme une contrainte ($\mathcal{C}_{Pi}$), représente le temps standard que chaque segment de processus prend pour traiter un lot standard de produits fabriqués. Cette standardisation dans le processus ($\mathcal{P}_i$) au sein de l'entreprise collaborative est importante puisque les caractéristiques standardisées du processus peuvent être aisément représentées avec les artéfacts ontologiques. Chaque segment processus relie les besoins matériel et équipement pour le processus considéré avec les attributs « Segment Matériel Spécification » et « Segment Équipement Spécification » comme montrés dans le diagramme de classe. Enfin, l'attribut « Segment Processus Dépendances » peut être utilisé pour représenter les similarités entre les entités processus caractérisées dans cette classe.

Le point central de l'intégration dans une entreprise virtuelle est, bien sur, le produit fabriqué qui est la matérialisation de la collaboration et de la coordination de leurs matériels et ressources. Conformément à la représentation des produits dans ISA-95 (cf. Figure 12, II.6.2), les familles de produit sont représentées sous la classe « Définition Produit », les multiples variantes des produits sont représentées sous la classe « Segment Produit » ou comme des instances de la classe « Définition Produit ». Chaque *Segment Produit* est composé d'informations relatives aux ressources matérielles, équipements et processus nécessaires pour la fabrication de l'entité produit. Ces informations sur le matériel, le processus et l'équipement du produit sont reliées via les attributs « Segment Produit Spécification Équipement », « Segment Produit Spécification Matériel » et « Segment Produit Spécification Processus » respectivement. Ces attributs seront transformés plus tard en relations logiques transitives dans l'artéfact ontologique.

Le *Segment Produit* peut aussi servir de point d'intégration pour les objets de connaissances (Knowledge Objects) relatifs au produit (cf. Figure 17, III.6.1). Les différentes possibilités concernant la représentation des *Segments Produit* et des *Définitions Produit* peuvent être considérées et dépendent du type de collaboration dans laquelle les entreprises s'engagent. Nous posons comme hypothèse que le produit qui est fabriqué dans l'entreprise virtuelle appartient à une seule famille de produit et que les ressources distribuées sont capables de produire les différentes variantes d'une même famille de produits. Les produits à familles multiples ne sont pas considérés dans ces travaux, car dans cette situation chaque famille de produit a ses propres besoins matériels, processus et équipement ce qui amène à un artéfact

ontologique plus détaillé ainsi qu'à des moteurs d'inférence de plus grande rapidité pour la prise de décision.

La dernière entité dans le schéma d'ISA-95 est celle qui représente la production dans l'entreprise. Cette entité est aussi représentée ici avec deux classes : « Production Planification » et « Production Requête ». La *Production Planification* agrège la *Production Requête*. Chaque *Production Requête* identifie un besoin de lot standard de *Segment Produit* exprimé par le client. Chaque entité de *Production Requête* exige un seul *Segment Produit* et la *Production Requête* est considérée comme une demande par période unitaire d'un lot de *Segment Produit*.

III.8 Conclusion

Dans ce chapitre, nous avons souligné les exigences d'intégration d'information de l'entreprise virtuelle et nous avons proposé une méthodologie d'intégration fondée sur l'ontologie du web. Les approches conventionnelles pour l'intégration d'entreprise traitent de l'intégration complète du système, ce qui a des conséquences sur le temps de mise en œuvre et les ressources nécessaires pour une telle intégration. De plus, des complications peuvent découler de l'échange d'informations avec des personnes non contractuelles à court terme entre les partenaires. Avec le flux d'information défini ci-dessus pour la création d'artefact ontologique, le serveur d'ontologies de domaine d'architecture centrale correspondant et le processus proposé pour l'intégration de contenu ontologique, l'entreprise distribuée peut partager les ressources d'informations pertinentes pour sa collaboration. Cette approche peut aider les petites et moyennes entreprises qui entreraient en collaboration, dans la gestion de l'intégration de leur artefact d'informations, tout en continuant à avoir la possibilité de changer de partenaires rapidement. La dernière partie du chapitre explique les technologies correspondantes permettant d'exploiter la capacité de l'artefact ontologie comme un artefact de configuration du système. Cette capacité de l'ontologie sera expliquée dans le Chapitre IV, dans lequel nous avons modélisé une ontologie OWL, basée sur les normes industrielles pour les modèles d'intégration de données d'entreprise de l'ISA-95, présentée en II.6. Dans le Chapitre IV, une étude de cas sera traitée pour la validation et de brèves introductions au langage d'ontologie OWL-DL, à SWRL (Semantic Web Language règle) ainsi qu'au moteur d'inférence JESS (Java Expert System Shell) sont fournies.

Chapitre IV
Application

Chapitre 4 : Application

IV.1 Introduction

Dans le Chapitre II, nous présentons la nomenclature des ontologies et les bases de l'ontologie de web sémantique (OWL) ainsi que la Logique Descriptive. Par la suite, nous discutons les différents processus existants en termes de modélisation d'ontologie. Ces processus serviront de base pour la modélisation du processus ontologique collaboratif, ainsi que pour l'intégration ultérieure de l'information produite.

Les éléments architecturaux proposés et leurs interactions ont pour but d'aider à gérer l'activité de l'intégration des artefacts de contenu avec l'artefact ontologie. Nous proposons également dans cette thèse d'identifier les multiples approches d'annotation utilisées pour l'intégration de contenus et de l'ontologie (cf. II.5.2), c'est-à-dire soit stocker l'annotation dans le contenu, soit au sein de l'artefact ontologique. Pour le scénario de l'entreprise virtuelle, nous proposons le stockage de l'annotation à l'intérieur de l'artefact ontologique.

Dans le Chapitre III, nous évoquons l'absence de travaux relatifs à la gestion des ressources dans les réseaux logistiques actuels. Ainsi, pour pouvoir répondre à cette flexibilité et à ce manque de souplesse, nous proposons une approche pour l'utilisation d'ontologies issues du web sémantique comme un artefact de configuration. Le résultat de cette approche est un système d'aide à la décision qui peut être utilisé pour la configuration des ressources dans les réseaux logistiques. Nous avons identifié les exigences architecturales pour la mise en œuvre ontologique. Nous avons aussi discuté de cette distinction entre un artefact distribué et modulaire ou un artefact ontologique central. Cette réflexion nous a amenés, dans le cadre d'une entreprise collaborative, à choisir un artefact central ayant des capacités de raisonnement et de déduction fournies par l'artefact ontologique. La plupart des travaux déjà présentés dans la littérature n'ont pas identifié les différents modèles ontologiques existants et n'ont pas souligné l'importance de l'utilisation des modèles de données standardisés en matière de modélisation ontologique. En partie, en raison de l'absence de modèles de données standardisés ou encore en raison de l'existence de plusieurs modèles de données dans lesquels les techniques d'intégration d'ontologies prennent de l'importance.

Dans le Chapitre III, l'idée de représenter les entités composant le réseau logistique comme des unités logiques d'un système global a été présentée et, ainsi, nous avons pu identifier une représentation ontologique adéquate des entités structurelles du réseau logistique. L'approche

de base est l'identification des différentes ressources présentes au sein de l'entreprise individuelle. Après cette identification des ressources, l'interaction entre les processus de production et les entités matérielles est identifiée (cf. III.6.2). Pour cela, un cadre pour la mise en œuvre ontologique au sein de l'entreprise distribuée est discuté en II.5.3.1. De plus, nous avons présenté un diagramme d'activité pour l'intégration du contenu généré au sein de l'entreprise virtuelle (Figure 18). L'importance de ce diagramme est de fournir aux entités réparties au sein de l'entreprise virtuelle, un raisonnement, une démarche, permettant de résoudre les divers problèmes qu'ils seraient amenés à rencontrer dans la mise en œuvre du système basé sur l'ontologie.

Ensuite, la décision concernant le choix d'un artefact central d'ontologie pour la base de connaissances basé sur une ontologie représentant les entités de la chaîne logistique et leurs artefacts d'information est argumentée. À la fin, une représentation complète et schématique des différents éléments intervenant dans le flux informationnel ainsi que la configuration du réseau logistique est délimitée.

Cette chaîne logistique intégrée est la dernière étape dans l'évolution de la chaîne logistique. L'intégration n'est plus limitée aux frontières de l'interaction simple entre deux entreprises, mais s'étend à l'ensemble de la chaîne logistique et, de plus, les limites de l'interaction des entités du réseau logistique peuvent changer rapidement en raison de l'engagement et du désengagement des entités constituant le réseau. Par conséquent, un facteur d'utilisation élevé du réseau, ainsi qu'une meilleure réponse aux changements topologiques, ne peut être atteint que grâce à une meilleure intégration des ressources en son sein. Ainsi, une approche permettant d'intégrer des artefacts informationnels ainsi que de gérer la configuration du réseau pour la production est une évolution significative par rapport aux travaux antérieurs.

L'intégration de l'entreprise distribuée nécessite aussi une méthode normalisée pour la modélisation des données et des processus ainsi qu'une identification d'outils automatisés qui peuvent s'appliquer à des modèles métiers. Nous avons, dans cette thèse, délimité ce cadre pour la conversion du modèle d'intégration d'entreprise ISA-95 dans l'ontologie. Nous avons ensuite appliqué des règles métier compatibles avec cette ontologie de Logique Descriptive. Cela fournit une base de connaissances dans un système central, capable d'intégrer les interactions de l'entreprise distribuée, qui agit comme une aide à la décision pour la configuration des réseaux logistiques. L'ontologie et les règles de gestions développées ont la portée du produit, du procédé, de la logistique et du matériel, mais d'autres règles logiques

métiers portant sur les objets de connaissance peuvent aussi être ajoutés par la méthode systémique présentée ci-dessous.

Les principaux problèmes rencontrés dans l'identification de ce cadre sont :

- Premièrement, l'identification de la norme internationale qui définit les ressources physiques présentes dans l'entreprise, dans ce but, nous proposons ISA-95 (cf. II.6) ;
- Deuxièmement, l'utilisation de cette norme d'intégration mono entreprise dans un modèle d'intégration multi entreprise. Cela demande l'identification de processus collaboratifs pour la modélisation des ressources de l'entreprise au cours de la phase initiale de la création de l'entreprise virtuelle discutée dans la Figure 18 ;
- Troisièmement, l'identification du modèle de réseau logistique permettant d'identifier la façon, dont les ressources de fabrication de l'entreprise utilisent les processus et la manière dont ils interagissent avec les ressources matérielles. Ce modèle est discuté dans la section III.4 ;
- Quatrièmement, la modélisation réelle de ces ressources et des entités de processus selon la sémantique comprise par la norme ISA-95. Pour cela nous proposons un modèle UML basé sur la norme ISA-95 qui est discuté dans la section III.7 ;
- Cinquièmement, la transformation du modèle UML en un artefact ontologique OWL-DL (Fahd A. *et al.*, 2011) (cf. IV.2.2) ;
- Sixièmement, l'identification de l'ensemble des outils et des bibliothèques correspondantes qui nous aideront dans la gestion de l'artefact ontologique ainsi que du système compatible avec l'artefact ontologique OWL-DL.

IV.2 Présentation du cas d'étude

Ce cas d'étude est utilisé pour expliquer l'approche proposée d'intégration des ressources dans une entreprise distribuée, comme discuté dans la section 1 de ce chapitre. Cette approche définit un cadre en partant de :

- Premièrement de l'étude des capacités logiques d'un langage de représentation ontologique ;
- Deuxièmement des besoins architecturaux de l'implémentation ontologique ;
- Troisièmement des besoins du processus correspondant entre les partenaires ;
- Quatrièmement de la sélection de standards industriels qui identifient les hiérarchies et les modèles de données correspondants pour les ressources de l'entreprise ;

- Cinquièmement de la phase d'implémentation et des diagnostics d'erreurs.

Nous avons abordé les contraintes de modélisation du réseau logistique dans la section III.4 et identifié les différentes entités importantes et nécessaires pour cette modélisation. Les principales entités dans le cadre d'un réseau logistique collaboratif sont les entités de produits/services qui servent de point de collaboration entre les différentes entreprises qui collaborent. Ces entreprises agrègent leurs ressources et leurs expertises pour fournir le produit voulu. D'autres entités importantes sont les entités équipements, matériels et de relation entre produits et processus ainsi qu'entre produits et ressources.

Pour l'intégration des ressources des entreprises distribuées, un système d'aide à la décision basé sur une ontologie centrale pour la configuration du réseau est implémenté. Le réseau d'entreprise contient de nombreux fournisseurs qui s'engagent dans la collaboration avec les autres entités entreprise. La capacité des fournisseurs est considérée comme infinie et la collaboration est restreinte à une seule famille de produits avec différentes variations en termes de matériels, mais pas en termes de processus de fabrication. Ce cas d'étude est inspiré de la production d'une famille de produits de type robinet qui est fabriqué, assemblé et conditionné dans des centres de production distribués (Figure 27). Les ressources matérielles basiques nécessaires pour la fabrication des robinets sont agrégées dans trois classes matérielles :

- Une classe Matériel métal : BB ;
- Une classe Matériel joint : BA ;
- Une classe Matériel carton d'emballage : A.

Ces *classes Matériel* sont utilisées pour agréger les entités de *Définition Matériel* qui sont nécessaires aux *Segments Produit* ($\mathfrak{M}_i$), fabriqués au sein de la chaîne logistique. Par exemple, une *Classe Matériel* métal BB peut agréger tous les composants métalliques de différents individus de *Matériel Définition* qui sont utilisés dans différents *Segment Produit*. De même, une classe A de carton d'emballage agrège toutes les options différentes d'emballage pour différents *Segments Produit*.

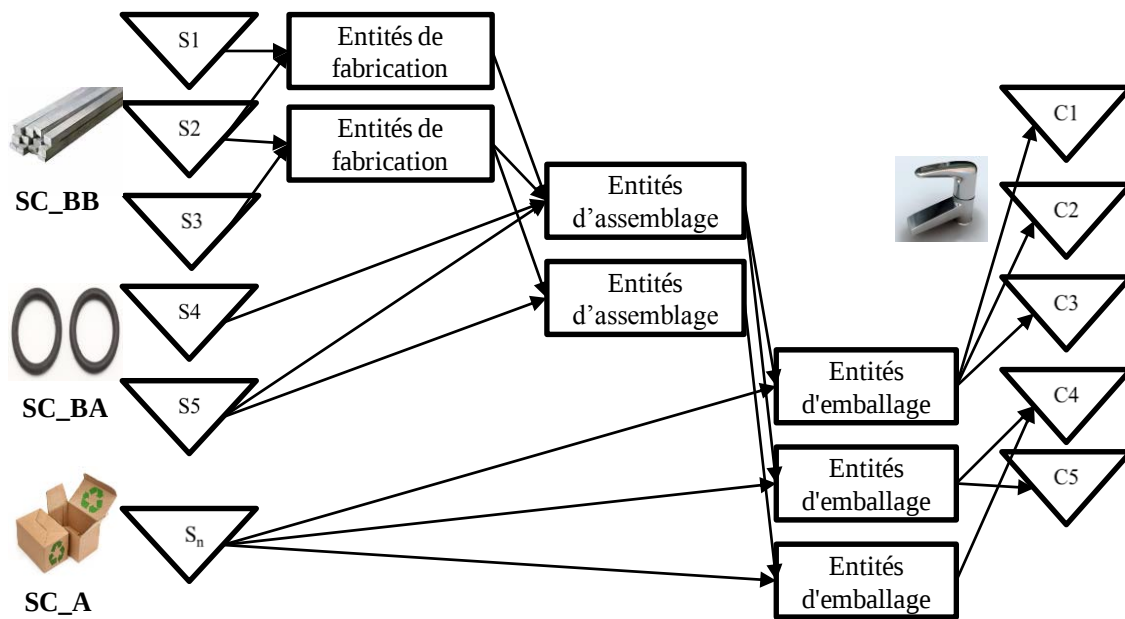


Figure 27 : Entités structurelles de l'étude de cas (Monteiro et al., 2007)

Ici, le processus industriel de fabrication d'un lot du *Segment Produit* demande trois *Segments Processus* de base ($\mathfrak{P}_i$). Le premier *Segment Processus* est la fabrication des corps supérieur et inférieur du robinet. Ce processus demande le transport de la *Classe Matériel* BB depuis différents fournisseurs ($\mathfrak{L}_{S,i}$) jusqu'à l'entité équipement disponible ($\mathfrak{L}_{M,i}$). Le deuxième processus est un processus d'assemblage et c'est l'agrégation des ressources transportées depuis le résultat produit par le premier processus de fabrication et du transport correspondant depuis le fournisseur de la *Classe Matériel* BA jusqu'aux ressources d'équipement d'assemblage disponibles dans l'entreprise collaborative. Le dernier *Segment Processus* est le transport du résultat produit par le précédent *Segment Processus* et par la troisième *Classe Matériel* A depuis les fournisseurs disponibles.

Nous avons ainsi clarifié les différents modèles de réseau logistique, expliqué le diagramme de classe, le contexte sémantique et dans la suite du manuscrit, nous allons fournir une introduction aux capacités de représentation sémantique d'OWL ainsi que la procédure de conversion complète pour l'utilisation de ce langage web ontologique comme un médium de configuration pour les ressources des réseaux d'entreprise. Cette conversion incorporera les contraintes de modélisation restantes, abordées dans la section 1.1 de ce chapitre. La nécessité de la section suivante est de clarifier les contraintes du langage de modalisation OWL, comme déjà discuté dans section 1.3. de ce chapitre.

IV.3 Langage Ontologique Web (OWL)

OWL, langage de représentation logique web, est capable de déclarer des domaines et des propriétés, de les organiser dans une hiérarchie et d'en fixer l'étendue. Les domaines des propriétés OWL sont les classes OWL et les étendues de ces propriétés peuvent être soit des classes OWL aussi, soit être des types de données prédéfinis tels que « string », « integer », etc. OWL peut définir une propriété comme transitive, symétrique, fonctionnelle et inversement et ses classes peuvent avoir une propriété particulière restreinte ainsi toutes les valeurs de la propriété sont des instances de la classe (Annexe - 4).

Voyons plus en détail les constructeurs de représentation logiques d'OWL.

IV.3.1 Constructeurs de langage OWL

IV.3.1.1 Description des Classes dans OWL

IV.3.1.1.1 Description de classes simples

Le concept de classe fournit un mécanisme abstrait pour regrouper les ressources ayant des caractéristiques similaires. Chaque classe est associée à un ensemble d'individus, nommés les instances de la classe. Les classes OWL sont définies de six manières possibles:

1. Un identifiant de classe (référence Universal Resources Identifier - URI) qui définit syntaxiquement une classe dans l'artéfact ontologique OWL ;
2. Une énumération exhaustive des individus qui, ensemble, définissent une classe ;
3. Une restriction de propriété sur un artéfact ontologique pour définir une classe ;
4. L'intersection de deux ou plus descriptions de classes ;
5. L'union de deux ou plus descriptions de classes ;
6. Le complément d'une description de classe.

Le premier type de description de classe définit un nom alors que les cinq autres donnent des classes anonymes en plaçant des restrictions sur les individus. Une description plus complexe des classes est possible via une description emboîtée des classes.

Il existe deux identifiants de classe prédéfinis dans toutes les ontologies OWL : **Owl:Thing**, qui contient tous les ensembles possibles d'individus et **Owl:Nothing** qui est l'ensemble vide.

IV.3.1.1.2 Description de classes complexes

IV.3.1.1.2.1 Description de classe par énumération

Cette description de classe identifie explicitement une liste d'individus qui sont les instances de la classe OWL, syntaxiquement via la propriété **owl:oneOf**.

IV.3.1.1.1.1 Description de classe par restriction

Cette description de classe décrit une classe anonyme de tous les individus qui satisfont à une certaine restriction définie dans l'artéfact ontologique. OWL distingue deux types de restrictions qui sont des contraintes de valeurs ou de cardinalité. Ces restrictions peuvent être appliquées soit sur les types de données pour lesquels la valeur de restriction est un nom de type, soit sur les propriétés objet où la valeur de restriction est un individu.

IV.3.1.1.2.3 Description de classe par intersection, union et complément.

Ces trois descriptions de classe représentent les constructeurs les plus avancés. **owl:intersectionOf** décrit une classe qui contient exactement les individus qui sont membres de l'ensemble des descriptions classes de la liste d'intersection. **owl:intersectionOf** peut être vu comme analogue à une conjonction logique. Au contraire, **owl:unionOf** est analogue à une disjonction logique. La description de classe **owl:complementOf** contient les individus qui ne sont pas dans les classes de la liste de complément. Elle est équivalente à une négation logique.

IV.3.1.2 Axiome de classe

Ces différentes définitions de classes sont les briques de construction élémentaires pour définir des classes dans les ontologies OWL, mais pour représenter les interrelations entre les classes déclarées, OWL fournit le concept d'axiomes de classe. La forme la plus simple de description de classe est la référence URI, mais des axiomes additionnels peuvent être utilisés pour définir des composants qui fixent les caractéristiques nécessaires et suffisantes de la classe OWL. OWL contient trois constructeurs à associer avec les descriptions de classe. Syntaxiquement, ces constructeurs sont des propriétés qui ont les descriptions de classe à la fois comme domaine et comme étendue.

1. **rdfs:subClassOf** permet de faire en sorte qu'une extension de classe (individus) d'une description de classe (concept) est un sous-ensemble d'une autre description de classe.

2. **owl:equivalentClass** permet de faire en sorte qu'une description de classe à la même extension de classe que celle d'une description équivalente. L'axiome d'équivalence fixe l'équivalence seulement en termes d'extensions de classe (individus) fournissant ainsi une relation entre les deux concepts (description de classes) mais n'établit pas d'égalité entre les concepts.
3. **owl:disjointWith** permet de faire en sorte que les extensions des descriptions de classe n'ont aucun membre en commun en tant que description de classe disjointe. Un axiome de classe peut contenir plusieurs occurrences d'**owl:disjointWith**.

IV.3.1.3 Individus OWL

Les individus sont définis avec les axiomes ou les faits individuels⁸. Les axiomes individuels sont principalement liés à deux secteurs :

1. Axiomes sur l'identité individuelle,
2. Axiomes sur l'appartenance à une classe et les valeurs de propriétés des individus.

Dans les différentes situations d'implémentation d'ontologies, différentes hypothèses sur l'identité des individus peuvent être posées. Avec des ontologies distribuées et multiples, l'Hypothèse de Nom Unique (UNA – Unique Naming Assumption) ne peut être appliquée puisque maintenir la consistance sur des ontologies distribuées demande un processus de coordination central de développement. UNA définit différents noms uniques pour référer à différentes choses distinctes dans la description du monde.

OWL fournit trois constructeurs pour fixer les choses au sujet de l'identité des individus :

1. **owl:sameAs** est utilisé pour établir que deux références URI pointent vers le même individu. Cette déclaration **owl:sameAs** lie les individus avec des individus et donc est souvent utilisée dans la définition de correspondances entre ontologies.
2. **owl:differentFrom** est utilisé pour établir que deux références URI pointent sur des individus différents.
3. **owl:AllDifferent** est utilisé pour établir que les individus d'une liste donnée sont distincts. Pour les ontologies dans lesquelles UNA est actif, cette déclaration est utilisée en opposition à **owl:differentFrom**, car celui-ci peut mener à un grand nombre de déclarations puisque chaque individu doit être déclaré comme disjoint par

⁸ Les faits sont également un terme pertinent utilisé dans le système essentiel et sera discuté en détail du point de vue de JESS et du moteur d'inférence.

paire. Ici, l'idée implicite est que tous les individus de la liste sont tous différents les uns des autres.

IV.3.1.4 Propriétés OWL

Nous avons donc présenté dans les sections précédentes différents scénarii à propos des descriptions de classe via les restrictions de propriétés et via les valeurs des propriétés des individus. Nous allons maintenant aborder la représentation formelle des propriétés OWL.

OWL offre deux catégories de propriétés qui peuvent être définies dans la construction d'une ontologie. La propriété objet **owl:ObjectProperty** lie les individus à des individus alors que la propriété type de données **owl:DatatypeProperty** lie des individus à des valeurs de données. Ces propriétés articulent les relations entre les différentes entités dans l'ontologie par leur représentation logique standardisée et en identifiant le sujet et l'objet de la propriété.

La hiérarchie entre les propriétés peut être représentée par l'axiome **rdfs:subPropertyOf** qui définit qu'une propriété est une sous-propriété d'une autre. Pour identifier le sujet d'une propriété, l'axiome **rdfs:domain** est utilisé. Cet axiome lie une propriété à une description de classe. Des axiomes domaines multiples peuvent être appliqués sur une unique propriété. Alternativement, pour représenter la propriété d'un objet, l'axiome **rdfs:range** est utilisé. Il lie la propriété soit à une description de classe, soit à une étendue de données.

Pour représenter les relations logiques entre les entités d'une classe, l'axiome **owl:TransitiveProperty** peut être utilisé. Généralement la relation transitive peut être utilisée pour représenter une relation « partie d'un tout » ou pour suggérer une relation directionnelle d'une entité domaine vers une entité étendue. À l'inverse, l'axiome **owl:SymmetricProperty** n'implique aucune direction dans la relation. Une propriété symétrique est une propriété qui indique que si la paire (x, y) est une instance de P, alors la paire (y, x) l'est aussi.

Pour exprimer la cardinalité sur le sujet et l'objet de la propriété objet ou la propriété type de données, **owl:FunctionalProperty** est utilisé. Une propriété fonctionnelle peut avoir seulement une valeur unique **y** pour chaque instance **x** (chaque sujet à un objet unique). L'axiome inverse, **owl:InverseFunctionalProperty** indique que l'objet d'une déclaration de propriété détermine uniquement le sujet (un individu) (une valeur **y** peut avoir seulement un **x** unique). Les propriétés objet et type de données sont disjointes, de fait une propriété fonctionnelle inverse ne peut être définie pour les types de données. Cela a une implication sur les capacités de raisonnement et d'inférence.

IV.3.1.5 Types de données

OWL utilise le même schéma de type de données que RDF, ce qui fournit un mécanisme de référencement du schéma de type de données XML (String, int, Float, Boolean, date, DateTime, Time) (<http://www.w3.org/2001/XMLSchema#>).

IV.3.2 Langage de règles sémantique web (Semantic Web Rule Language – SWRL)

SWRL est un langage de règles basé sur OWL qui permet à l'utilisateur d'écrire des règles qui peuvent être exprimées en termes de concepts OWL (par exemple description de classe, propriété objet et types de données, individus, etc.). SWRL a les mêmes fondations de Logique Descriptive. Les règles SWRL consistent en une partie antécédente nommée « head » et une suite nommée « body ». L'une et l'autre contiennent les conjonctions d'atomes. Les règles peuvent être interprétées comme : si les atomes dans l'antécédent sont vrais, alors les atomes de la suite le sont aussi. L'algorithme de correspondance de schémas utilisé par le moteur de règles est expliqué en Annexe - 5 **Erreur ! Source du renvoi introuvable.** et les limites technologiques sont présentées Figure 25.

IV.4 Représentation ontologique d'ISA-95

IV.4.1 Introduction

Toutes les classes des modèles sont représentées comme des sous-classes d'**owl:Thing**, toutes les classes modélisées dans l'éditeur d'ontologie Protégé sont représentées comme une sous-classe de cette classe. La plupart des relations dans le modèle UML sont remplacées par les propriétés des objets correspondants. Les détails du domaine et la gamme des propriétés de l'objet seront expliqués plus tard dans cette thèse. Les relations d'héritage sont converties en une combinaison d'objets et les propriétés des types de données avec la définition de leur hiérarchie.

Pour modéliser l'ontologie, le produit est considéré comme le point de référence parmi l'ensemble des entreprises collaborant. Chaque segment individuel du produit représente la correspondance entre le matériel et les ressources d'équipement ainsi que les exigences du processus.

IV.4.2 Nomenclature pour la représentation ontologie

Cette section décrit en détail l'approche de la transformation du modèle UML concerné dans un artefact ontologique OWL-DL unique. Le processus de description de la conversion du modèle UML vers le langage OWL-DL, se fait comme suit :

1. La propriété Description de type de données primitive: String (*) dans une classe est représenté en interne dans l'ontologie de web sémantique (OWL-DL) par **rdfs:label** pour chaque entité individuelle. Chaque individu créé au sein de l'ontologie OWL-DL possède cette propriété interne de **rdfs:label**. L'ontologie suit une hypothèse de nom unique. Donc, il ne peut pas avoir deux entités individuelles ayant les mêmes propriétés **rdfs:label** au sein de l'artefact d'ontologie.
2. La procédure de nommage des classes de l'ontologie a sa première lettre en majuscule et utilise la fonte gras cambria math (c'est-à-dire **Location, Equipment**, etc), la convention de nommage pour les propriétés d'objet de l'ontologie est : minuscules Courier New, suivie par la première lettre en majuscule Courier New (c'est-à-dire `has_Location`, `has_Material_Class`).
3. Les propriétés sont représentées de la même manière que les propriétés de l'objet, mais avec du Times New Roman (c'est-à-dire `has_Equipment_ID`, `has_Location_ID`), et les instances sont représentées par la première lettre en majuscule et Arial Narrow italique (c'est-à-dire *Production_Request_1*).

IV.4.2.1. Outils de mise en œuvre

L'outil utilisé dans la modélisation de l'ontologie est OWL-DL Protégé version 3.4.4 ; Jambalaya le module de l'extension de Protégé est utilisé pour la visualisation d'ontologie, pour la modélisation des contraintes sur l'artefact ontologique, on applique le langage de règles pour le web sémantique (SWRL⁹-Semantic Web Rule Language). Pour la déduction automatique sur l'ontologie modélisée et les règles correspondantes de SWRL¹⁰, le système essentiel basé sur java est utilisé (JESS – Java Expert System Shell). Pour la modélisation UML, nous avons utilisé la version 7 de Visual Paradigm. Plus tard, pour la génération automatique de la population et pour l'application séquentielle des règles, l'interface de programmation Protégé-OWL¹¹, qui est une bibliothèque java pour le langage ontologique web, est utilisée. Cette interface de programmation fournit des classes et des méthodes pour

⁹ <http://www.w3.org/Submission/SWRL/>

¹⁰ L'interaction entre l'ontologie différente et un composant du moteur de règles sont illustrés dans la Figure 25

¹¹ <http://protege.stanford.edu/plugins/owl/api/>

charger et enregistrer des fichiers OWL et pour les opérations générales de gestion de l'artefact ontologie OWL. Elle fournit aussi la capacité autonome d'exploiter les avantages de l'artefact ontologique sans l'interface utilisateur graphique (GUI) de Protégé.

L'interface de programmation Protégé-OWL fournit également le soutien aux requêtes en les groupant au sein l'interface de programmation SQWRL. Cette interface de programmation fournit la capacité d'extraire l'information explicite ou déduite sur l'artefact ontologique. SQWRL est fondée sur la langue de règles SWRL et utilise la fondation sémantique de SWRL. Le langage de requête qui en résulte fournit un petit, mais puissant, arsenal pour l'opérateur qui permet, ainsi, aux utilisateurs de construire des requêtes sur l'ontologie d'OWL. SQWRL contient également un ensemble d'opérateurs qui peuvent être utilisés pour effectuer les opérations de clôture, pour permettre des formes limitées de la négation comme un échec et pour faire des opérations de comptage et d'agrégation. L'interface de programmation Protégé-OWL fournit aussi les bibliothèques java concernant l'application et la mise en œuvre des règles SWRL.

L'interface de programmation SWRL Rule Engine fournit un mécanisme pour créer par programmation des entités du moteur de règles SWRL et permet d'interagir avec ces entités créées. L'interface de programmation SWRL Factory est une autre interface java qui prend en charge la création et l'importation des règles SWRL dans une ontologie d'OWL. L'interface de programmation SWRL Engine Bridge fournit l'infrastructure nécessaire pour intégrer les moteurs de règles dans Protégé-OWL et exécuter les règles de SWRL. Plus tard, pour l'interaction avec la feuille Excel en tant qu'une base de données dorsale, l'interface de programmation JXL est utilisée pour l'importation et par la suite pour écrire l'information déduite de l'artefact ontologique dans la feuille Excel. Pour les opérations de contrôle, l'environnement de développement intégré Eclipse pour Java est utilisé, en raison de la limitation de la disponibilité de ces bibliothèques dans le langage java.

IV.4.2.2. La conversion d'UML à l'OWL-DL (Fahd A. *et al.*, 2011)

La procédure pour l'utilisation de l'artefact ontologique OWL-DL pour l'intégration des ressources au sein d'une entreprise virtuelle distribuée suit les étapes suivantes. D'abord fondée sur l'exigence du réseau virtuel, un modèle UML est construit en gardant la cohérence de la sémantique du modèle d'intégration d'entreprise de l'ISA-95. Cette étape a déjà été discutée en détail précédemment (cf. III.7).

La deuxième étape est la conversion de ce modèle UML dans un artefact équivalent d'ontologie OWL-DL. Cela implique la collaboration entre les ingénieurs de la connaissance de l'entreprise distribuée pour identifier les classes pour l'inclusion, ainsi que leurs relations hiérarchiques correspondants au sein de l'artefact ontologique. Cette étape comporte aussi l'étendue des informations que les partenaires/collaborateurs sont prêts à partager au sein d'un artefact ontologique central, combiné et partagé.

La troisième étape est la modélisation de l'artefact ontologique avec le logiciel de modélisation d'ontologie (Protégé). Cette étape est constituée d'un processus collaboratif pour la modélisation de l'artefact ontologique. Cette étape consiste en la déclaration de classes, des propriétés de l'objet entre les classes ainsi que des propriétés qui représentent les attributs des classes.

En résumé, pour représenter le modèle équivalent à UML dans l'ontologie d'OWL-DL, un ensemble de trois types de diagrammes sont nécessaires. Le premier est le schéma hiérarchique qui représente la relation de généralisation parmi les classes modélisées de d'UML. La seconde instance est le schéma qui représente l'agrégation de chaque classe d'entité liaison entre les deux classes modélisées dans UML. Enfin le diagramme des propriétés d'objet qui représente l'association de composition entre les classes modélisées d'UML. En outre, les attributs de classe peuvent être représentés comme les propriétés de l'objet au sein des classes définies de l'ontologie OWL-DL.

IV.4.2.2.1 Représentation de la relation de généralisation en OWL-DL

Basée sur les exigences de l'étude de cas du modèle UML, les ressources matérielles et les équipements sont représentés dans la Figure 26. La Figure 28 représente la relation de généralisation entre les classes modélisées de l'ontologie avec la relation `has_subclass` entre la classe `owl:Thing`.

Dans l'artefact ontologique les classes définies :

- **Product_Definition** modélise les entités de produits fabriqués en collaboration au sein de l'entreprise virtuelle.
- **Knowledge_Object** modélise les contenus informationnels qui sont créés et maintenus pour fournir un soutien à l'intégration d'informations entre les partenaires.

- **Material_Definition** et **Material_Class** modélisent les entités qui représentent les ressources matérielles nécessaires à la fabrication du produit et pour l'agrégation des ressources matérielles, respectivement.
- **Equipment** et **Equipment_Class** modélisent les entités qui représentent les ressources d'équipement nécessaires à la fabrication du produit et pour l'agrégation des ressources d'équipement, respectivement.
- **Process_Segments** modélise les entités de traitement et leurs attributs associés impliqués dans la fabrication et la transformation des produits.
- **Production_Schedule** modélise l'entité qui représente les activités de production au sein de l'entreprise virtuelle et les ressources nécessaires en fonction des exigences des produits.
- **Location** modélise des entités qui représentent les attributs de localisation individuelle des ressources matérielles, des ressources d'équipement, et des clients finaux.

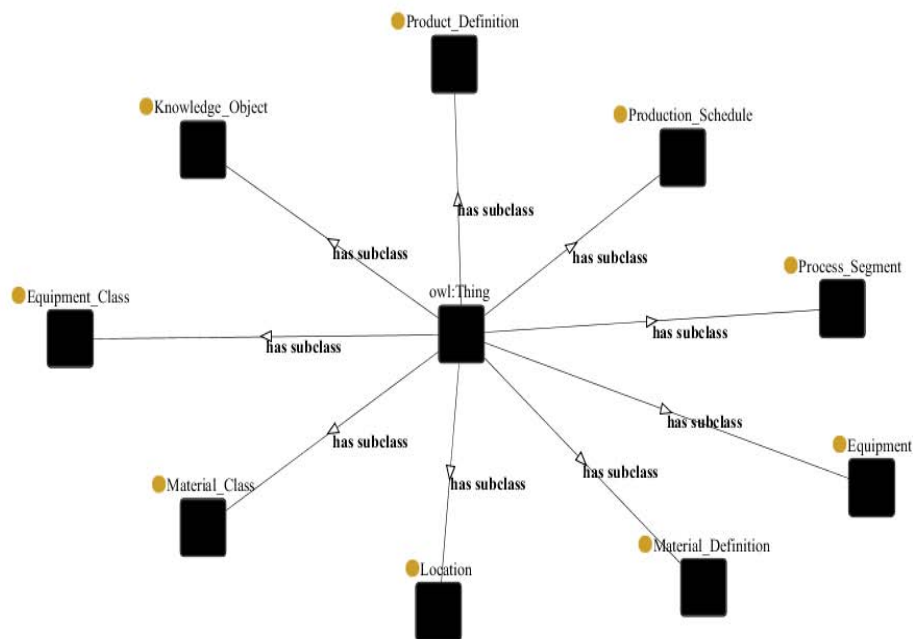


Figure 28 : Hiérarchie des classes d'ontologie

IV4.2.2.2. Représentation de la relation de la composition en OWL-DL

La section 4.2.2.1 précise les relations hiérarchiques entre les classes du modèle UML au sein de l'artefact d'ontologie, mais elle ne représente pas l'instanciation de chaque classe qui est

représentée par l'association d'agrégation d'UML entre les classes. Cette section représente cet aspect de l'ontologie.

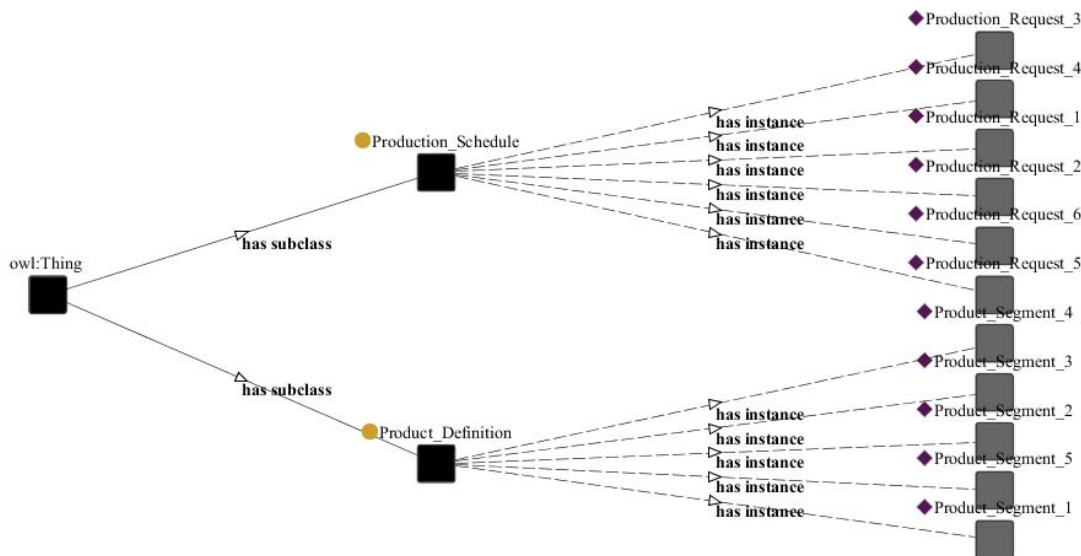


Figure 29 : Instanciation de la classe et production des produits de classe

La Figure 29 fournit l'instanciation des individus des classes **Production_Schedule** et **Product_Definition** dans l'ontologie. Selon ISA-95, la norme d'intégration d'entreprise, la classe **Production_Schedule** agrège la demande pour les cas individuels de la classe définie **Product_Definition**¹². La relation d'agrégation et la multiplicité de la relation de composition dans le modèle UML (Figure 26), entre les classes « Production Schedule » et « Production Request », est représentée au sein de l'artefact d'ontologie comme une propriété `has_instance`. Par exemple, la classe **Production_Schedule**, agrégats multiples de l'instance de la classe « Production Request » représentée dans le modèle UML, représente chacune des instances de la demande de l'instance individuelle de la classe « Product Segment ». Chaque **Production_Schedule** contient les engagements de production de l'entreprise virtuelle pour une famille de produits qui est définie comme des instances individuelles dans l'artefact ontologique. Chaque cas individuel *Production_Request* contient la demande d'une instance de la classe « Product Segment ». Ici, l'hypothèse de base, dans la modélisation de l'ontologie de l'artefact, est que chaque artefact ontologique est défini pour une seule famille de produits.

¹² Chaque instance individuelle de la classe définie **Product_Definition** représentent la variante de produit de la gamme de produits créée au sein de l'entreprise virtuelle

Le classe définie d'ontologie **Production_Schedule** contient des instances de la classe « Production Request » où l'attribut `Production_Request_Identification` est représenté au sein de l'artefact ontologique par la propriété fonctionnelle de datatype¹³ `has_Production_Request_ID`, de type `xsd:integer`. De même l'attribut `Production_Request_Description`, dans le modèle UML, est représenté en interne pour chaque entité individuelle d'ontologie par la propriété primitive datatype `rdfs:label`.

Basé sur l'analyse des besoins de l'étude de cas discutée plus tôt en IV.2 ainsi que sur les exigences de modélisation de la chaîne logistique présentées en III.4, les ressources matérielles sont représentées au sein de l'artefact ontologique dans la classe définie **Material_Class** avec les trois instances individuelles *SC_A*, *SC_BB*, *SC_BA*. Ces instances de la classe agrègent les instances des ressources matérielles en fonction des propriétés des matériaux pour l'assurance de la qualité, de leurs essais de qualité ou de leurs similarités dans les exigences de la connaissance des objets (Figure 30).

Basée sur l'exigence processus des produits, trois processus (fabrication, assemblage et emballage) sont modélisés dans l'artefact ontologique. Comme déjà discuté en III.6.2, la modélisation des processus nécessite la compréhension des processus d'affaires et c'est l'ingénieur d'ontologies de l'entreprise virtuelle collaborative qui doit spécifier la granularité des processus mis en cause pour la production de la famille de produits spécifiée. L'hypothèse d'une seule famille de produits est prise en se basant sur la similitude des processus de production. La seule différence entre les segments de produits est la différence de ressources matérielles utilisées dans différents segments de produits. Cette étape est prise pour plus de simplicité dans l'étude de cas, mais cela peut facilement être étendu à des scénarios plus complexes avec la définition de règles supplémentaires. Chacun des segments de processus identifie la classe de matériel et d'équipement nécessaire pour le segment processus individuel. Il identifie également les contraintes de durée sur chaque instance de la classe **Process_Segment**. Par exemple la contrainte de temps sur la *Manufacturing_Process_Segment* individuelle peut être définie dans l'artefact ontologique par la propriété type de données avec son domaine de **Process_Segment**. L'ontologie OWL-DL fournit des propriétés de données de type entier, chaîne, date, date et heure, temps, booléen, et réel.

¹³ R  f  rer    la section 3 du chapitre 4 pour la d  finition de la propri  t   fonctionnelle de datatype

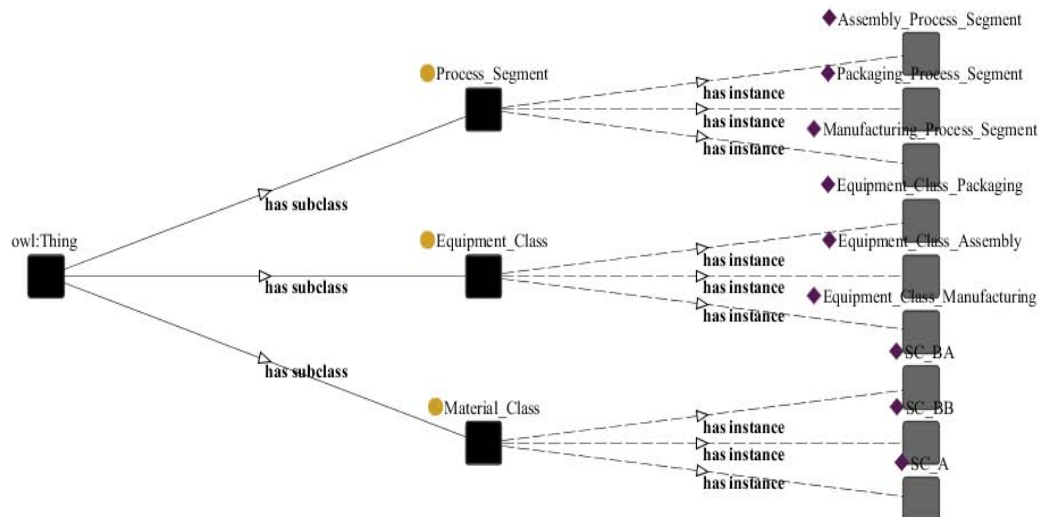


Figure 30 : Instanciation du classe processus, de la classe matérielle et de la classe équipement

Sur la base de l'exigence processus spécifiée par le produit, les partenaires distribués peuvent partager leurs ressources d'équipement, cela est évident dans l'instanciation de classe **Equipment_Class** (Figure 30). Dans toutes les classes d'OWL-DL ci-dessus, l'attribut UML de l'identification est représenté par une propriété fonctionnelle de type de données xsd: integer.

IV.4.2.2.3. Représentation des propriétés d'un objet dans l'ontologie OWL-DL

Dans les deux sous sections précédentes, l'ontologie équivalente au diagramme d'UML a été présentée. Les associations de généralisation des classes définies dans l'ontologie avec la super classe d'ontologie **owl:Thing** a été discutée. Puis l'association d'agrégation avec ses contraintes de multiplicité a été représentée dans les artefacts ontologiques, en même temps, les attributs de la classe définie sont également représentés par la propriété de type de données fonctionnelles. Maintenant, la relation entre les différentes classes définies dans l'artefact ontologique sera discutée sur la base de la représentation du modèle UML pour les ressources distribuées.

Comme les propriétés des objets sont utilisées pour rapporter les relations entre les classes définies au sein de l'artefact ontologique, les propriétés des objets qui sont définies sur ces classes d'ontologie peuvent également s'étendre à chacun des individus de la classe. L'ontologie OWL-DL peut représenter deux types de base de la représentation logique, c'est d'abord la relation transitive logique et en second la relation de similitude. Ces deux propriétés de l'objet nécessitent une classe de domaine et une classe gamme au sein de

l'artefact ontologique. Basé sur l'exigence de l'étude de cas et les entités de ressources réseau, la classe définie **Material_Definition** contient des instances de toutes les entités matérielles utilisées au sein de l'entreprise virtuelle et l'emplacement (lieu de fourniture) de chaque entité matériau peut être lié par la propriété transitive fonctionnelle `has_Location`. Comme chaque entité individu de matériel a un fournisseur unique, cette propriété transitive rapporte l'entité matériel avec son fournisseur via l'entité d'emplacement. De même, l'approche pour modéliser la ressource équipement disponible dans le réseau est également appliquée. Chaque entité d'équipement est liée à son entité emplacement par la propriété d'objet transitive et fonctionnelle `has_Location`. La Figure 31 est un instantané de l'artefact ontologique représentant les propriétés de l'objet entre les classes définies. Une représentation schématique similaire peut être faite pour les ressources d'équipement.

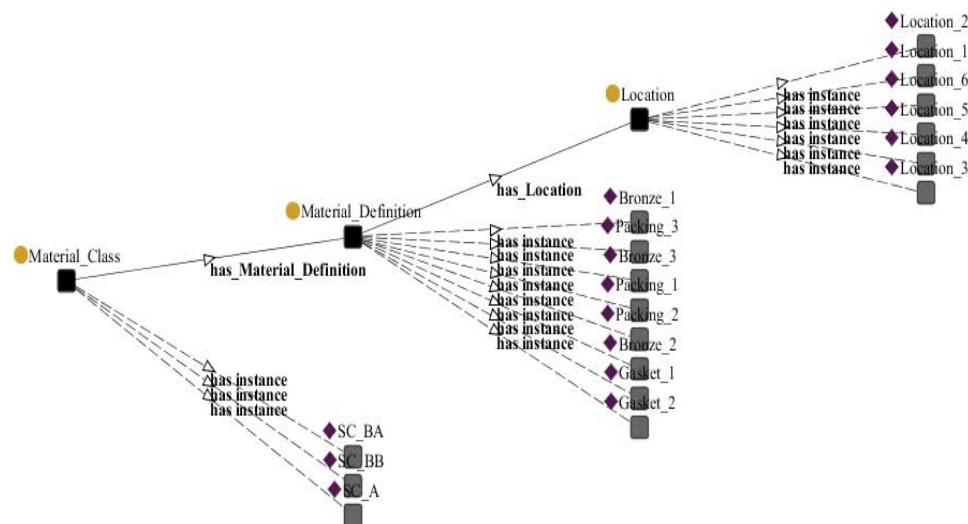


Figure 31 : Instanciation des ressources matérielles

Les propriétés de l'objet et les propriétés de type de données correspondantes définies dans l'artefact d'ontologie sont représentés Figure 32 et les caractéristiques restantes des classes définies sont individuellement examinées. Cette figure représente le domaine et l'étendue de chacune des propriétés de l'objet.

La classe définie **Process_Segment**, modélise l'agrégation de l'instance entière des processus impliqués dans la fabrication de la famille de produits. Cette classe comprend deux propriétés, `has_Segment_Material_Class_Specification` définissant l'objet transitif et l'autre `has_Segment_Equipment_Class_Specification` qui est la propriété de l'objet fonctionnelle transitive avec son domaine de **Process_Segment** et sa gamme **Equipment_Class**. L'utilisation de la propriété fonctionnelle transitive entre les classes définies

Process_Segment et **Equipment_Class** est due au fait que ce processus est pris en charge par les instances d'équipements regroupées sous la catégorie unique d'équipement. Et cela puisque chaque segment de processus a un ensemble d'équipements associé qui sont regroupés sous une catégorie d'équipements, comme par exemple la fabrication de la partie supérieure et inférieure du robinet qui nécessite le processus de moulage sous pression.

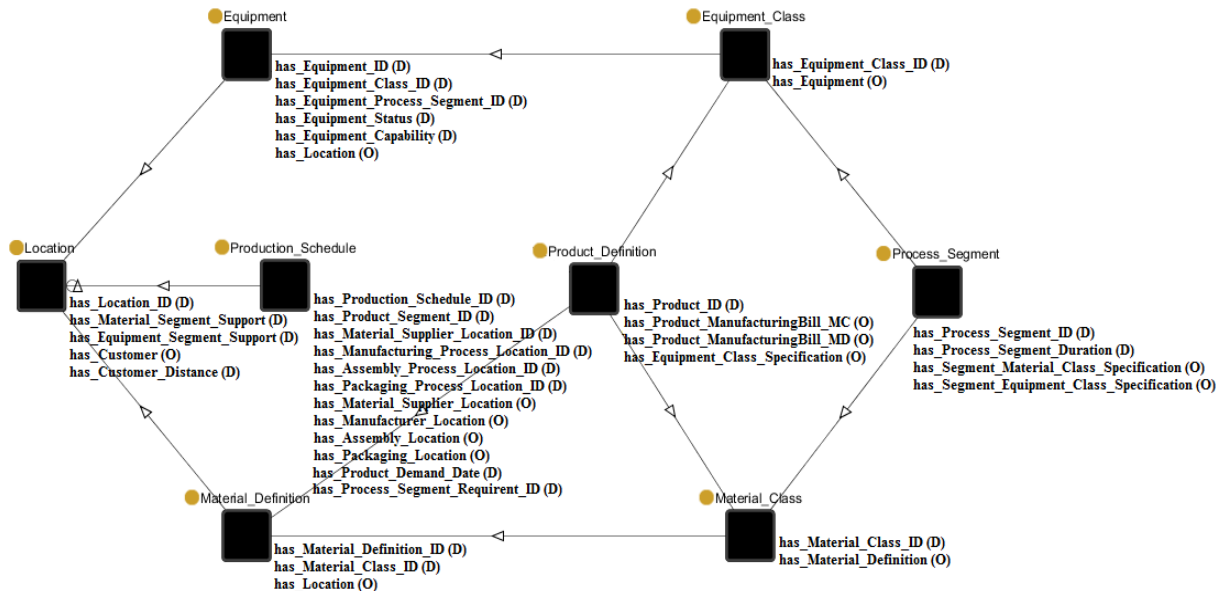


Figure 32 : Domaine et gamme des propriétés de l'objet et propriétés associées de type de données

La classe définie **Product_Definition** est un agrégat de tous les segments de produits ou des variantes de produits d'une seule famille de produits qui sont fabriqués au sein de l'entreprise virtuelle. Pour chaque segment de produits les spécifications matérielles sont représentées par deux propriétés d'objet transitives **has_Product_ManufacturingBill_MC** et **has_ProductManufacturingBill_MD** qui ont pour domaine **Product_Definition** et pour gammes **Material_Class** et **Material_Definition** respectivement.

De même, la spécification d'équipement est identifiée pour chaque segment de produits par les propriétés d'objet transitives **has_Product_Segment_Equipment_Specification** avec pour domaine **Product_Definition** et pour gamme **Equipment_Class**. Ainsi, chaque instance de produit individuel définit ses besoins en matières ainsi que les ressources de catégories d'équipements nécessaires à sa fabrication. L'approche de la télé-intégration des équipements à l'entreprise virtuelle est abordée dans la Figure 33.

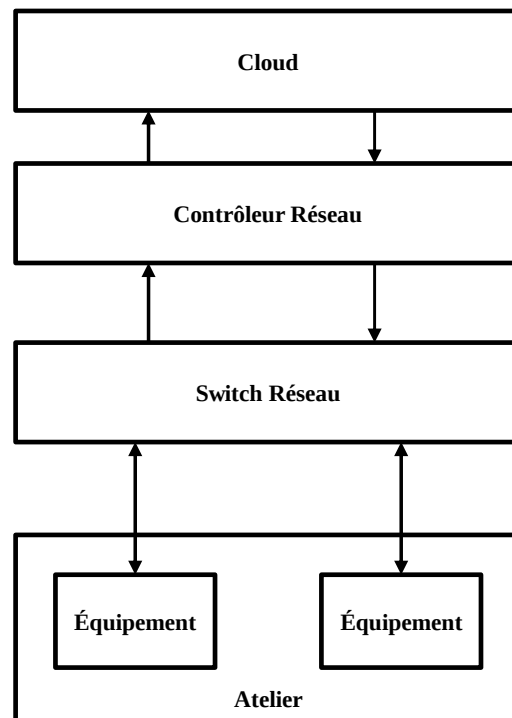


Figure 33 : Approche d'intégration des équipements (Khan W.A. et al., 2011)

La classe définie qui modélise la production au sein de l'entreprise est **Production_Schedule**. Cette classe agrégat définit la demande de produits de la clientèle comme une instance individuelle de *Production_Request*. Chaque *Production_Request* représente l'exigence de lot partie d'un segment de produit fabriqué au sein de l'entreprise virtuelle. Le modèle de données interne pour chaque individu *Production_Request* comporte la date et l'heure de la demande de production, qui est représentée comme une propriété de type de données xsd: date&heure. L'identification de la demande individuelle est représentée comme la propriété de type de données xsd:integer. Ce sont les valeurs affirmées des propriétés qui sont nécessaires pour le modèle ontologique.

Afin d'utiliser l'ontologie comme un artefact de configuration du procédé au sein de l'entreprise virtuelle, la demande pendant une période de temps discret est recueillie, puis est utilisée pour peupler les individus de la classe de l'artefact ontologique *Production_Request* avec les propriétés de type de données date, heure et segment de produits. Les propriétés supplémentaires de la Figure 34 représentent l'objet et les propriétés de type de données qui sont inférées automatiquement par l'application des règles SWRL sur l'artefact ontologique par le moteur de règles JESS. Les événements de séquences complètes pour les commandes et les informations correspondantes transférées à l'artefact ontologique seront représentés plus

tard en IV.5 et Figure 36. Dans la sous section subséquente, les règles qui sont définies dans SWRL seront discutées en détail.

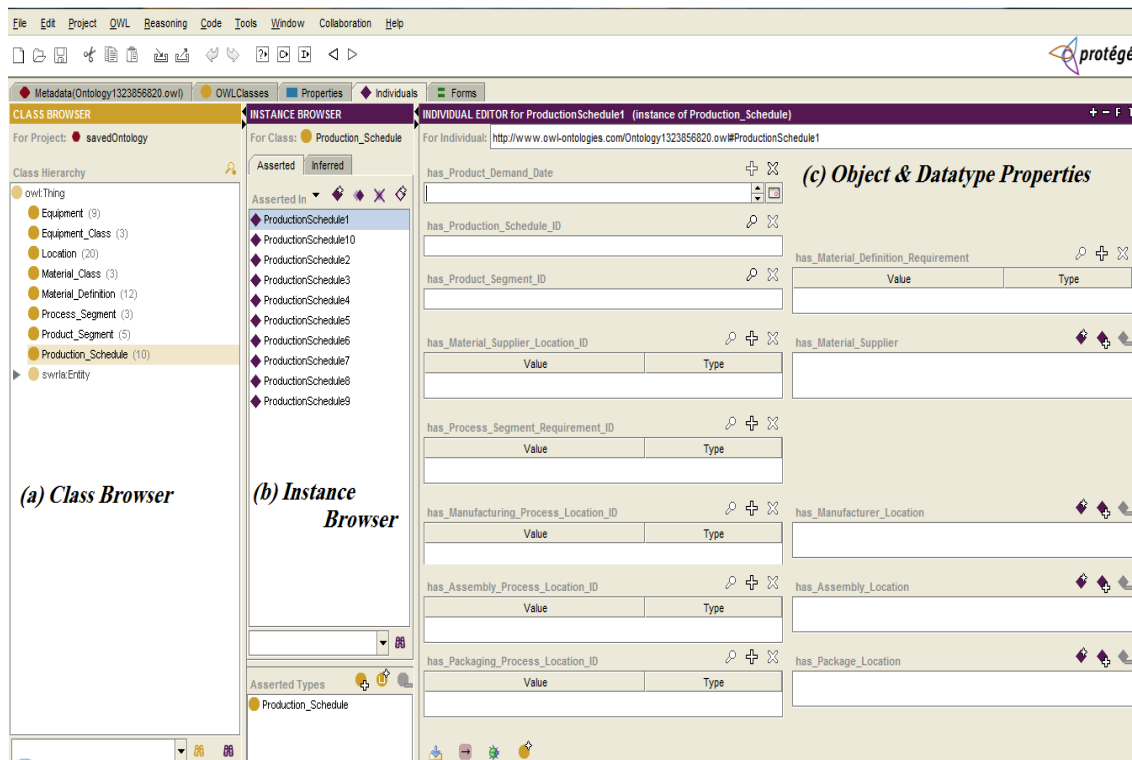


Figure 34 : Ontologie modélisée avec des données d'instance

La dernière classe définie dans l'artefact ontologique est la classe **Location**. Cette classe agrégats contient toutes les instances individuelles de localisation au sein de l'entreprise virtuelle et l'associe à l'emplacement des ressources matérielles, l'emplacement des ressources d'équipements et l'emplacement du client. En effet, chaque individu matériau à un fournisseur avec son entité emplacement correspondante. De même, le matériel partagé au sein de l'entreprise a une instance emplacement unique et chaque demande de production est également accompagnée de l'identification de l'emplacement du client.

Pour modéliser les interdépendances entre individus de la classe **Location**, une propriété transitive **has_Customer** est définie avec son domaine et sa gamme au sein de la classe **Location** et pour représenter la distance entre ces individus, une propriété **has_Customer_Distance** de type de données **xsd:String** est utilisée. Ici, la propriété est de type « chaîne » en raison de la contrainte d'expressivité d'OWL-DL. En effet, la relation de transitivité entre l'individu de la classe **Location** est représentée par la relation **has_Customer**, mais pour représenter la valeur de la distance entre les entités de localisation individuelles, cela nécessite l'expression de l'entité d'emplacement initiale,

l'entité d'emplacement finale et la distance entre les deux entités. Les interdépendances logiques entre les instances d'emplacement des individus sont représentées dans la Figure 35, où l'ontologie représente la propriété de type de données `has_Customer_Distance` avec son identification d'emplacement initial, son identification de l'emplacement final et la distance entre les deux instances de localisation individuelles.

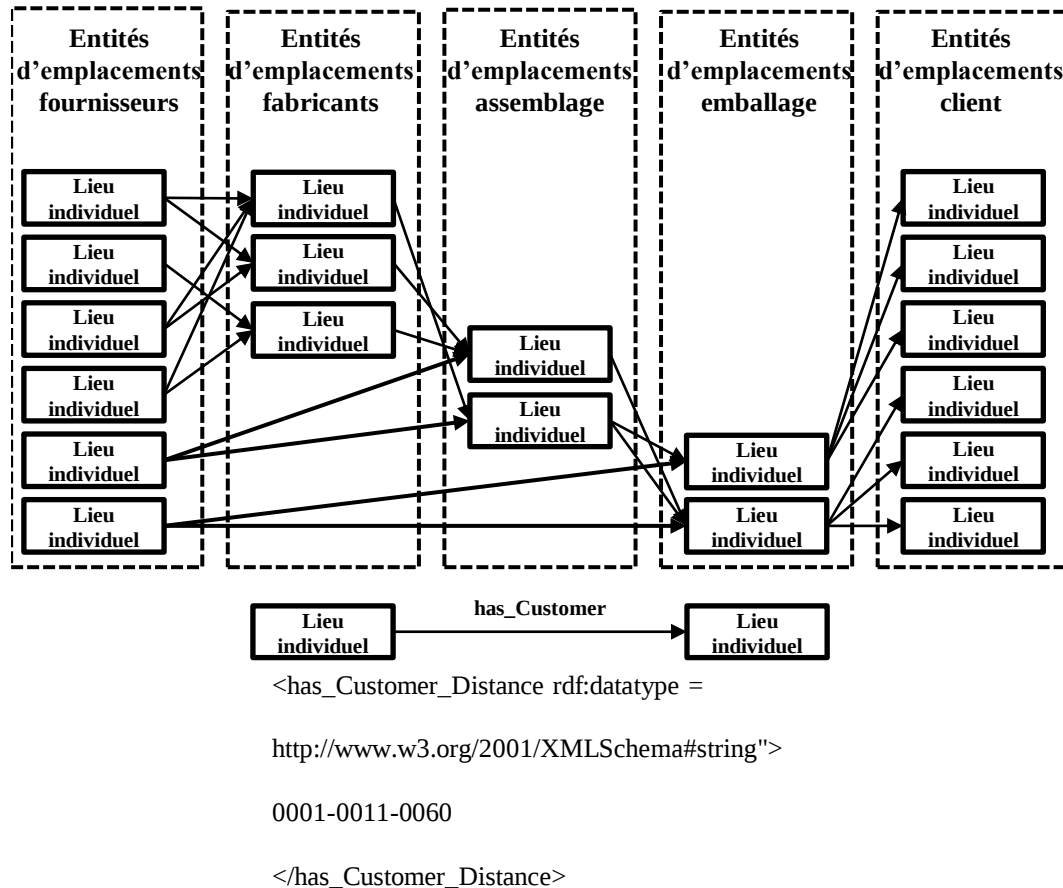


Figure 35 : Relations logiques entre les emplacements des individus

Dans la section précédente, les propriétés individuelles des objets qui sont équivalentes au modèle UML sont représentées et leurs représentations sémantiques et syntaxiques décrivent leur fonctionnement. L'association différente de la généralisation, de la composition et de l'agrégation avec leur multiplicité dans la classe du modèle UML est transformée dans l'artefact ontologique équivalent, mais pour l'utilité de cet artefact ontologique, comme une base de connaissances pour la configuration du procédé, des règles SWRL doivent lui être appliquées. Ces règles sont classées en séries multiples, en fonction de leurs caractéristiques fonctionnelles et seront discutées dans la section suivante. La propriété de l'objet complet de l'ontologie pour les entités de l'entreprise virtuelle est présentée dans la Figure 32.

IV.5 Application de la règle SWRL sur l'artefact ontologie modélisée

Le langage de règles du web sémantique (SWRL) peut être décliné. Par exemple, on peut citer SWRL:equal, SWRL:sameAs, etc. où ces opérateurs de comparaison peuvent être appliqués aux propriétés de types de données au sein de l'artefact ontologique. En outre, SWRL fournit également plusieurs opérateurs mathématiques pour les types de données de type entier, string, date, heure pour une référence complète des capacités de SWRL¹⁴.

Trois principaux types de règles sont définis sur l'artefact ontologique (Figure 36) :

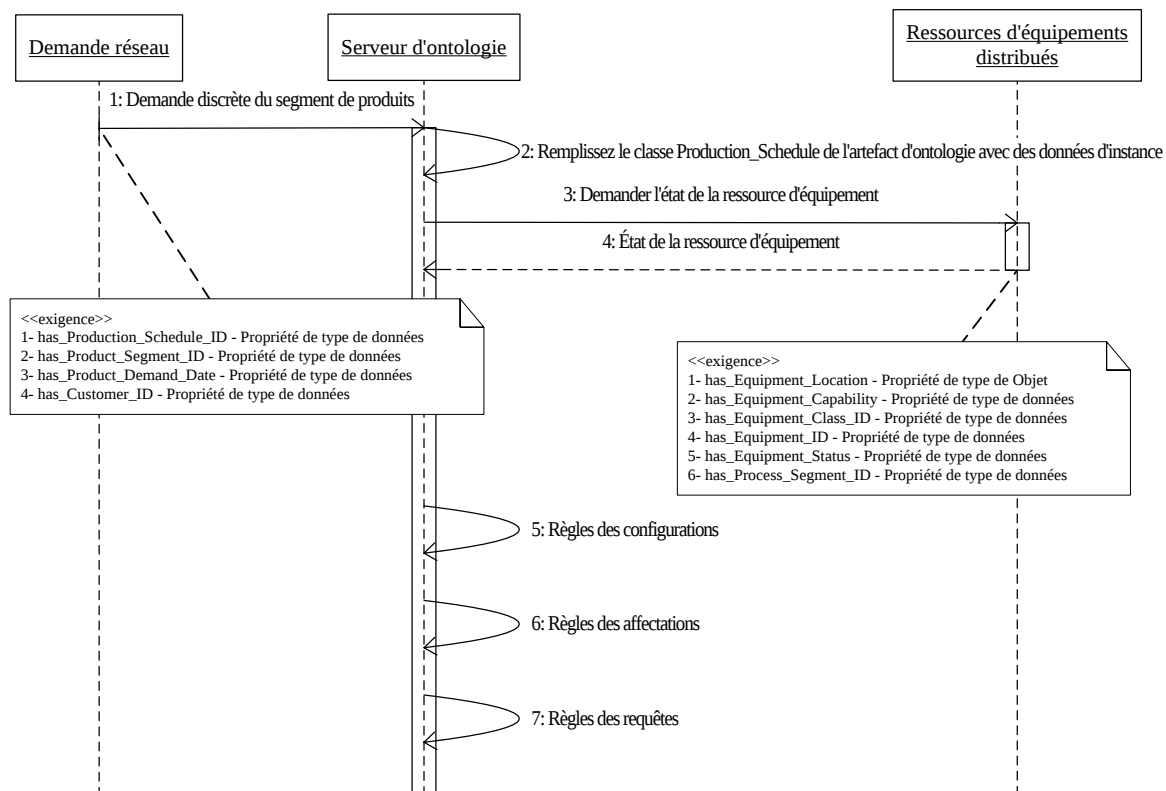


Figure 36 : Diagramme de séquence pour l'application des règles sur l'artefact ontologique pour l'intégration des ressources

- les règles de configuration, qui sont utilisées pour la configuration initiale de l'artefact ontologique ;
- les règles d'affectation, qui assignent les exigences matérielles du produit et les exigences du processus de produits basés sur les spécifications des équipements de produit. De même, ces règles d'affectation sont également utilisées pour l'attribution du matériel, de l'équipement, des processus et de leurs entités de localisation correspondant à chacun des plans de production individuels ;

¹⁴ <http://www.w3.org/Submission/SWRL/>

- les règles de requête, définies pour extraire la connaissance inférée de l'artefact ontologique.

Règle Set 1 : La règle 1.1 et la règle 1.2 sont utilisées pour la configuration initiale de l'artefact ontologique, pour les instances de la définition du matériel, en fonction de leurs propriétés des classes de matériel et d'emplacement, qui sont déjà modélisées dans l'artefact ontologique. La règle 1.2 utilise la propriété emplacement de l'objet associé à chaque instance de la définition du matériau et agrège ces instances au sein des instances de la classe d'emplacement, configurant ainsi l'artefact ontologique avec toutes les ressources matériau disponibles. De même, les règles 1.3 et 1.4 sont utilisées pour la configuration initiale de l'artefact ontologique pour les entités de l'équipement. La règle 1.3 identifie l'équipement disponible à intervalles discrets : si la propriété état de l'équipement individuel est « disponible », à l'intervalle de la demande discrète alors cette information sur la disponibilité est exprimée par l'identification de l'équipement.

1.1. **Material_Definition (?a) $\wedge$ has_Material_Class_ID (?a, ?b) $\wedge$ Material_Class (?c) $\wedge$ has_Material_Class_ID (?c, ?d) $\wedge$ swrlb:equal (?b, ?d) $\rightarrow$ has_Material_Definition (?c, ?a)**

1.2. **Material_Definition (?a) $\wedge$ has_Location (?a, ?b) $\wedge$ has_Material_Definition_ID (?a, ?c) $\wedge$ Location (?d) $\wedge$ sameAs (?b, ?d) $\rightarrow$ has_Material_Segment_Support (?d, ?c)**

1.3. **Equipment (?a) $\wedge$ has_Equipment_Location (?a, ?b) $\wedge$ has_Process_Segment_ID (?a, ?x) $\wedge$ has_Equipment_Status(?a, "Available") $\wedge$ Location (?c) $\wedge$ has_Location_ID (?c, ?d) $\wedge$ sameAs (?b, ?c) $\rightarrow$ has_Process_Segment_Support (?c, ?x)**

1.4. **Equipment (?a) $\wedge$ has_Equipment_Class_ID (?a, ?b) $\wedge$ Equipment_Class (?c) $\wedge$ has_Equipment_Class_ID (?c, ?d) $\wedge$ swrlb:equal (?b, ?d) $\rightarrow$ has_Equipment (?c, ?a).**

Règle Set 2: Ces règles de configuration déterminent l'identification du produit associé à la demande discrète de l'individu segment de produit et attribuent les exigences par lot de production des entités définition de matériel, sur la base des exigences de produits individuels de la classe de matériaux qui sont pré-modélisées dans la classe segment de produits au sein de l'artefact ontologique.

2.1. **Product_Segment** (?a) $\wedge$ has_Product_ID (?a, ?b) $\wedge$ has_Product_ManufacturingBill_MC (?a, ?c) $\wedge$ sqwrl:makeSet (?MatClass, ?c) $\wedge$ sqwrl:element (SC_A, ?MatClass) $\rightarrow$ has_Product_ManufacturingBill_MD_Quantity (?a, "1 = Material Class SC_A Lot Requirement")

2.2. **Product_Segment** (?a) $\wedge$ has_Product_ID (?a, ?b) $\wedge$ has_Product_ManufacturingBill_MC (?a, ?c) $\wedge$ sqwrl:makeSet (?MatClass, ?c) $\wedge$ sqwrl:element (SC_BA, ?MatClass) $\rightarrow$ has_Product_ManufacturingBill_MD_Quantity (?a, "2 = Material Class SC_BA Lot Requirement")

2.3. **Product_Segment** (?a) $\wedge$ has_Product_ID (?a, ?b) $\wedge$ has_Product_ManufacturingBill_MC (?a, ?c) $\wedge$ sqwrl:makeSet (?MatClass, ?c) $\wedge$ sqwrl:element (SC_BB, ?MatClass) $\rightarrow$ has_Product_ManufacturingBill_MD_Quantity (?a, "2 = Material Class SC_BB Lot Requirement").

Règle Set 3: Ces règles de configuration précisent les identifications individuelles du produit dans la demande discrète du segment de produit et permettent d'attribuer les exigences du processus du produit, basées sur la spécification de l'équipement du segment de produit qui est déjà modélisé dans l'artefact ontologique.

3.1. **Product_Segment** (?a) $\wedge$ has_Product_ID(?a, ?b) $\wedge$ has_Equipment_Class_Spec(?a, ?c) $\wedge$ sqwrl:makeSet(?ProdEq, ?c) $\wedge$ sqwrl:element(EClass_M, ?ProdEq) $\rightarrow$ has_Product_Process_Requirement(?a, 1)

3.2. **Product_Segment** (?a) $\wedge$ has_Product_ID (?a, ?b) $\wedge$ has_Equipment_Class_Spec (?a, ?c) $\wedge$ sqwrl:makeSet (?ProdEq, ?c) $\wedge$ sqwrl:element (EClass_A, ?ProdEq) $\rightarrow$ has_Product_Process_Requirement (?a, 2)

3.3. **Product_Segment** (?a) $\wedge$ has_Product_ID (?a, ?b) $\wedge$ has_Equipment_Class_Spec (?a, ?c) $\wedge$ sqwrl:makeSet (?ProdEq, ?c) $\wedge$ sqwrl:element (EClass_P, ?ProdEq) $\rightarrow$ has_Product_Process_Requirement (?a, 3).

Après les trois premières séries de règles de configuration, les séries suivantes sont appliquées pour gérer la configuration des ressources qui sont présentes dans l'entreprise virtuelle. Ceci est réalisé par l'identification des ressources d'équipement et de matériel pour chaque demande de production puis par l'identification des instances d'emplacement du matériau et des ressources équipement pour la demande de production. À la fin de l'application de la

règle, chaque demande individuelle de production contiendra l'ensemble des emplacements possibles pour ses fournisseurs en matières, son équipement de fabrication, l'assemblage, l'emballage et les distances inter ressources ainsi que les contraintes de temps.

Règle Set 4: Cette règle d'attribution attribue à chaque demande de production les exigences du segment de processus, en identifiant les segments de produits. Les exigences processus pour chaque segment de produit ont déjà été modélisées dans l'artefact ontologique au cours de la phase de modélisation.

4.1. **Production_Schedule (?a) $\wedge$ has_Product_Segment_ID (?a, ?b) $\wedge$ Product_Segment (?c) $\wedge$ has_Product_ID (?c, ?d) $\wedge$ has_Product_Process_Requirement (?c, ?e) $\wedge$ swrlb:equal (?b, ?d) $\rightarrow$ has_Process_Segment_Requirement_ID (?a, ?e).**

Règle Set 5: Ces règles d'affectation attribuent à chaque individu de la demande de production, les exigences correspondantes de matériau, en identifiant le segment des produits individuels et en assignant l'instance de la définition du matériau. Les exigences matérielles de chaque segment de produits ont déjà été modélisées dans l'artefact ontologique au cours de la phase de modélisation.

5.1. **Production_Schedule (?a) $\wedge$ has_Product_Segment_ID (?a, 1) $\wedge$ Product_Segment (?b) $\wedge$ has_Product_ID (?b, 1) $\wedge$ has_Product_ManufacturingBill_MD (?b, ?e) $\wedge$ sqwrl:makeSet (?ProdMD, ?e) $\wedge$ Material_Definition (?f) $\wedge$ has_Material_Definition_ID (?f, ?g) $\wedge$ sqwrl:makeSet (?SetMD, ?f) $\wedge$ sqwrl:intersection (?element, ?SetMD, ?ProdMD) $\wedge$ sqwrl:element (?h, ?element) $\wedge$ has_Material_Definition_ID (?h, ?z) $\rightarrow$ has_Material_Definition_Requirement (?a, ?z).**

5.2. **Production_Schedule (?a) $\wedge$ has_Product_Segment_ID (?a, 2) $\wedge$ Product_Segment (?b) $\wedge$ has_Product_ID (?b, 2) $\wedge$ has_Product_ManufacturingBill_MD (?b, ?e) $\wedge$ sqwrl:makeSet (?ProdMD, ?e) $\wedge$ Material_Definition (?f) $\wedge$ has_Material_Definition_ID (?f, ?g) $\wedge$ sqwrl:makeSet (?SetMD, ?f) $\wedge$ sqwrl:intersection (?element, ?SetMD, ?ProdMD) $\wedge$ sqwrl:element (?h, ?element) $\wedge$ has_Material_Definition_ID (?h, ?z) $\rightarrow$ has_Material_Definition_Requirement (?a, ?z).**

5.3. **Production_Schedule (?a) $\wedge$ has_Product_Segment_ID (?a, 3) $\wedge$ Product_Segment (?b) $\wedge$ has_Product_ID (?b, 3) $\wedge$ has_Product_ManufacturingBill_MD (?b, ?e) $\wedge$ sqwrl:makeSet (?ProdMD, ?e) $\wedge$ Material_Definition (?f) $\wedge$ has_Material_Definition_ID (?f,**

$?g) \wedge \text{sqwrl:makeSet} (?SetMD, ?f) \wedge \text{sqwrl:intersection} (?element, ?SetMD, ?ProdMD) \wedge$
 $\text{sqwrl:element} (?h, ?element) \wedge \text{has_Material_Definition_ID} (?h, ?z) \rightarrow$
 $\text{has_Material_Definition_Requirement} (?a, ?z).$

5.4. **Production_Schedule (?a)** $\wedge \text{has_Product_Segment_ID} (?a, 4) \wedge \text{Product_Segment} (?b)$
 $\wedge \text{has_Product_ID} (?b, 4) \wedge \text{has_Product_ManufacturingBill_MD} (?b, ?e) \wedge$
 $\text{sqwrl:makeSet} (?ProdMD, ?e) \wedge \text{Material_Definition} (?f) \wedge \text{has_Material_Definition_ID} (?f,$
 $?g) \wedge \text{sqwrl:makeSet} (?SetMD, ?f) \wedge \text{sqwrl:intersection} (?element, ?SetMD, ?ProdMD) \wedge$
 $\text{sqwrl:element} (?h, ?element) \wedge \text{has_Material_Definition_ID} (?h, ?z) \rightarrow$
 $\text{has_Material_Definition_Requirement} (?a, ?z).$

5.5. **Production_Schedule (?a)** $\wedge \text{has_Product_Segment_ID} (?a, 5) \wedge \text{Product_Segment} (?b)$
 $\wedge \text{has_Product_ID} (?b, 5) \wedge \text{has_Product_ManufacturingBill_MD} (?b, ?e) \wedge$
 $\text{sqwrl:makeSet} (?ProdMD, ?e) \wedge \text{Material_Definition} (?f) \wedge \text{has_Material_Definition_ID} (?f,$
 $?g) \wedge \text{sqwrl:makeSet} (?SetMD, ?f) \wedge \text{sqwrl:intersection} (?element, ?SetMD, ?ProdMD) \wedge$
 $\text{sqwrl:element} (?h, ?element) \wedge \text{has_Material_Definition_ID} (?h, ?z) \rightarrow$
 $\text{has_Material_Definition_Requirement} (?a, ?z).$

Règle Set 6: Ces règles d'affectation associent chaque demande individuelle de production avec l'emplacement du fournisseur de matériel et l'emplacement correspondant du processus de fabrication, pour chacun des segments de produit. Ces informations peuvent ensuite être récupérées à partir de l'artefact ontologique à travers l'opération de requête.

6.1. **Production_Schedule (?a)** $\wedge \text{has_Material_Definition_Requirement} (?a, ?b) \wedge$
 $\text{sqwrl:makeSet} (?PSMDid, ?b) \wedge \text{sqwrl:groupBy} (?PSMDid, ?a) \wedge \text{Location} (?x) \wedge$
 $\text{has_Location_ID} (?f, ?x1) \wedge \text{has_Material_Segment_Support} (?f, ?g) \wedge \text{sqwrl:makeSet}$
 $(?LocMD, ?g) \wedge \text{sqwrl:groupBy} (?LocMD, ?f) \wedge \text{sqwrl:intersection} (?MaterialDefID,$
 $?LocMD, ?PSMDid) \wedge \text{sqwrl:element} (?z1, ?MaterialDefID) \rightarrow$
 $\text{has_Material_Supplier_Location_ID} (?a, ?x1).$

6.2. **Production_Schedule (?a)** $\wedge \text{has_Production_Schedule_ID} (?a, ?b) \wedge$
 $\text{has_Process_Segment_Requirement_ID} (?a, 1) \wedge \text{Location} (?x) \wedge \text{has_Location_ID} (?x, ?y) \wedge$
 $\text{has_Process_Segment_Support} (?x, ?z) \wedge \text{sqwrl:makeSet} (?LocationInstanceProcessSupport,$
 $?z) \wedge \text{sqwrl:groupBy} (?LocationInstanceProcessSupport, ?y) \wedge \text{swrlb:equal} (?z, 1) \wedge$
 $\text{sqwrl:element} (?locPS, ?LocationInstanceProcessSupport) \rightarrow$
 $\text{has_Manufacturing_Process_Location_ID} (?a, ?y).$

6.3. **Production_Schedule (?a)** $\wedge$ has_Production_Schedule_ID (?a, ?b) $\wedge$ has_Process_Segment_Requirement_ID (?a, 2) $\wedge$ **Location (?x)** $\wedge$ has_Location_ID (?x, ?y) $\wedge$ has_Process_Segment_Support (?x, ?z) $\wedge$ sqwrl:makeSet (?LocationInstanceProcessSupport, ?z) $\wedge$ sqwrl:groupBy (?LocationInstanceProcessSupport, ?y) $\wedge$ swrlb:equal (?z, 2) $\wedge$ sqwrl:element (?locPS, ?LocationInstanceProcessSupport) $\rightarrow$ has_Assembly_Process_Location_ID (?a, ?y).

6.4. **Production_Schedule (?a)** $\wedge$ has_Production_Schedule_ID (?a, ?b) $\wedge$ has_Process_Segment_Requirement_ID (?a, 3) $\wedge$ **Location(?x)** $\wedge$ has_Location_ID (?x, ?y) $\wedge$ has_Process_Segment_Support (?x, ?z) $\wedge$ sqwrl:makeSet (?LocationInstanceProcessSupport, ?z) $\wedge$ sqwrl:groupBy (?LocationInstanceProcessSupport, ?y) $\wedge$ swrlb:equal(?z, 3) $\wedge$ sqwrl:element (?locPS, ?LocationInstanceProcessSupport) $\rightarrow$ has_Packaging_Process_Location_ID (?a, ?y).

6.5. **Location (?a)** $\wedge$ has_Location_ID (?a, ?b) $\wedge$ sqwrl:makeSet (?LocationID, ?b) $\wedge$ **Production_Schedule (?c)** $\wedge$ has_Material_Supplier_Location_ID (?c, ?d) $\wedge$ sqwrl:makeSet (?ProductionSchedule, ?d) $\wedge$ sqwrl:groupBy (?ProductionSchedule, ?c) $\wedge$ sqwrl:intersection (?element, ?LocationID, ?ProductionSchedule) $\wedge$ sqwrl:element (?e, ?element) $\wedge$ has_Location_ID (?a, ?e) $\rightarrow$ has_Material_Supplier (?c, ?a).

6.6. **Location (?a)** $\wedge$ has_Location_ID (?a, ?b) $\wedge$ sqwrl:makeSet (?LocationID, ?b) $\wedge$ **Production_Schedule (?c)** $\wedge$ has_Manufacturing_Process_Location_ID (?c, ?d) $\wedge$ sqwrl:makeSet (?ProductionSchedule, ?d) $\wedge$ sqwrl:groupBy (?ProductionSchedule, ?c) $\wedge$ sqwrl:intersection (?element, ?LocationID, ?ProductionSchedule) $\wedge$ sqwrl:element (?e, ?element) $\wedge$ has_Location_ID (?a, ?e) $\rightarrow$ has_Manufacturer_Location (?c, ?a).

6.7. **Location (?a)** $\wedge$ has_Location_ID (?a, ?b) $\wedge$ sqwrl:makeSet (?LocationID, ?b) $\wedge$ **Production_Schedule (?c)** $\wedge$ has_Assembly_Process_Location_ID (?c, ?d) $\wedge$ sqwrl:makeSet (?ProductionSchedule, ?d) $\wedge$ sqwrl:groupBy (?ProductionSchedule, ?c) $\wedge$ sqwrl:intersection (?element, ?LocationID, ?ProductionSchedule) $\wedge$ sqwrl:element (?e, ?element) $\wedge$ has_Location_ID (?a, ?e) $\rightarrow$ has_Assembly_Location (?c, ?a).

6.8. **Location (?a)** $\wedge$ has_Location_ID (?a, ?b) $\wedge$ sqwrl:makeSet (?LocationID, ?b) $\wedge$ **Production_Schedule (?c)** $\wedge$ has_Packaging_Process_Location_ID (?c, ?d) $\wedge$ sqwrl:makeSet (?ProductionSchedule, ?d) $\wedge$ sqwrl:groupBy (?ProductionSchedule, ?c) $\wedge$ sqwrl:intersection

$(?element, ?LocationID, ?ProductionSchedule) \wedge sqwrl:element (?e, ?element) \wedge has_Location_ID (?a, ?e) \rightarrow has_Package_Location (?c, ?a).$

Règles Set 7: Après l'application séquentielle des règles de configuration et des règles d'affectation dans SWRL avec l'aide de l'API Protégé-OWL, l'artefact ontologique résultant est interrogé via l'API SQWRL pour obtenir l'information pertinente déduite, cette information est ensuite utilisée pour remplir la feuille Excel en utilisant l'API JXL.

7.1. **Production_Schedule (?a) $\wedge$ has_Product_Segment_ID (?a,?x) $\wedge$ has_Customer_ID (?a,?y) $\wedge$ has_Material_Supplier (?a, ?b) $\wedge$ has_Customer_Distance (?b, ?c) $\rightarrow$ sqwrl:select (?a,?c,?x,?y) $\wedge$ sqwrl:orderBy (?a).**

7.2. **Production_Schedule (?a) $\wedge$ has_Product_Segment_ID (?a,?x) $\wedge$ has_Customer_ID (?a,?y) $\wedge$ has_Manufacturer_Location (?a, ?b) $\wedge$ has_Customer_Distance (?b,?c) $\rightarrow$ sqwrl:select (?a,?c,?x,?y) $\wedge$ sqwrl:orderBy (?a).**

7.3. **Production_Schedule (?a) $\wedge$ has_Product_Segment_ID (?a,?x) $\wedge$ has_Customer_ID (?a,?y) $\wedge$ has_Assembly_Location (?a, ?b) $\wedge$ has_Customer_Distance (?b,?c) $\rightarrow$ sqwrl:select (?a,?c,?x,?y) $\wedge$ sqwrl:orderBy (?a).**

7.4. **Production_Schedule(?a) $\wedge$ has_Product_Segment_ID (?a,?x) $\wedge$ has_Customer_ID (?a,?y) $\wedge$ has_Package_Location(?a, ?b) $\wedge$ has_Customer_Distance(?b,?c) $\rightarrow$ sqwrl:select(?a,?c,?x,?y) $\wedge$ sqwrl:orderBy (?a).**

7.5. **Production_Schedule(?a) $\wedge$ has_Product_Segment_ID(?a, ?b) $\wedge$ Product_Segment(?c) $\wedge$ has_Product_ID(?c, ?d) $\wedge$ swrlb:equal(?b, ?d) $\wedge$ has_Product_ManufacturingBill_MD_Quantity(?c, ?x) $\rightarrow$ sqwrl:selectDistinct(?a, ?x) $\wedge$ sqwrl:orderBy(?a)**

7.6. **Production_Schedule(?a) $\wedge$ has_Manufacturer_Location (?a, ?b) $\wedge$ sqwrl:makeSet (?ManufacturerMachineLocation, ?b) $\wedge$ sqwrl:groupBy (?ManufacturerMachineLocation, ?a) $\wedge$ Equipment (?c) $\wedge$ has_Equipment_Location (?c, ?e) $\wedge$ has_Equipment_Material_Usage_Efficiency (?c, ?d) $\wedge$ sqwrl:makeSet (?EquipmentLocation, ?e) $\wedge$ sqwrl:intersection (?eqLoc, ?ManufacturerMachineLocation, ?EquipmentLocation) $\wedge$ sqwrl:element (?e, ?eqLoc) $\wedge$ has_Location_ID (?e, ?ID) $\rightarrow$ sqwrl:select (?a, ?ID, ?d).**

7.7. **Production_Schedule** (?a) $\wedge$ has_Assembly_Location (?a, ?b) $\wedge$ sqwrl:makeSet (?AssemblyMachineLocation, ?b) $\wedge$ sqwrl:groupBy (?AssemblyMachineLocation, ?a) $\wedge$ **Equipment** (?c) $\wedge$ has_Equipment_Location (?c, ?e) $\wedge$ has_Equipment_Material_Usage_Efficiency (?c, ?d) $\wedge$ sqwrl:makeSet (?EquipmentLocation, ?e) $\wedge$ sqwrl:intersection (?eqLoc, ?AssemblyMachineLocation, ?EquipmentLocation) $\wedge$ sqwrl:element (?e, ?eqLoc) $\wedge$ has_Location_ID (?e, ?ID) $\rightarrow$ sqwrl:select (?a, ?ID, ?d).

7.8. **Production_Schedule** (?a) $\wedge$ has_Package_Location (?a, ?b) $\wedge$ sqwrl:makeSet (?PackageMachineLocation, ?b) $\wedge$ sqwrl:groupBy (?PackageMachineLocation, ?a) $\wedge$ **Equipment** (?c) $\wedge$ has_Equipment_Location (?c, ?e) $\wedge$ has_Equipment_Material_Usage_Efficiency (?c, ?d) $\wedge$ sqwrl:makeSet (?EquipmentLocation, ?e) $\wedge$ sqwrl:intersection (?eqLoc, ?PackageMachineLocation, ?EquipmentLocation) $\wedge$ sqwrl:element (?e, ?eqLoc) $\wedge$ has_Location_ID (?e, ?ID) $\rightarrow$ sqwrl:select (?a, ?ID, ?d).

7.9. **Production_Schedule**(?a) $\wedge$ has_Process_Segment_Requirement_ID (?a,?x) $\wedge$ **Process_Segment**(?b) $\wedge$ has_Process_Segment_ID (?b,?x) $\wedge$ has_Process_Segment_Duration (?b,?c) $\rightarrow$ sqwrl:select(?a,?b,?c,?x) $\wedge$ sqwrl:orderBy (?a).

7.10. **Production_Schedule**(?a) $\wedge$ has_Product_Demand_Date (?a,?x) $\rightarrow$ sqwrl:select(?a,?x) $\wedge$ sqwrl:orderBy (?a).

IV.6 Présentation et discussion des résultats

Pour l'utilité de l'artefact ontologique comme artefact de configuration, les principales sources d'information nécessaires sont d'abord les informations d'entrée qui identifient les entités de la chaîne logistique du réseau, puis l'information qui concerne l'état actuel des entités du réseau logistique, c'est-à-dire soit les entités fournisseurs, soit les entités équipements disponibles. Enfin, la demande du client correspondant qui est représentée par la demande de production et qui est utilisée pour le remplissage de l'artefact ontologique. Avec ces sources d'informations, l'artefact ontologique modélisé est rempli à l'aide des bibliothèques Java identifiées dans IV.4.2.1. Il convient de préciser ici que le modèle d'intégration des ressources d'entreprises distribuées avec l'artefact ontologique est également identifié dans la Figure 33.

Les ressources combinées des entités de la chaîne logistique sont modélisées dans l'artefact ontologique initial avec leurs informations correspondantes concernant leur soutien au

processus. Ce modèle d'information combine toutes les ressources de l'entreprise distribuée au sein d'un artefact ontologique central. Avec cette information modélisée, les distances dans le réseau de ressources sont également modélisées dans l'artefact ontologique. L'approche de modélisation des distances dans ce réseau de ressources est expliquée en détail dans la Figure 35. Sur la base de l'ensemble des règles d'affectation, des règles de configuration et des règles de requête définies précédemment, les informations déduites sur les ressources disponibles sont extraites de l'artefact ontologique et elles peuvent en outre être utilisées aux fins de la planification des ressources du réseau logistique.

Les deuxième et troisième sources d'informations sont respectivement la disponibilité des ressources matérielles matière et la demande du client. À chaque intervalle de temps, l'artefact ontologique est mis à jour avec la demande du client et avec l'état de disponibilité des ressources matérielles et équipements modélisées. Cette demande discrète et l'interaction correspondante des composants sont représentées au sein de la Figure 36.

Le livrable de notre approche est donc l'information nécessaire pour planifier l'utilisation des ressources distribuées au sein de l'entreprise virtuelle. Ce résultat se compose de l'identification du lieu, ainsi que du réseau de distances entre les ressources distribuées.

Les ensembles de règles 1, 2 et 3 configurent l'artefact ontologique. L'ensemble de règles 1 configure l'artefact ontologique avec les informations nécessaires sur les entités de localisations disponibles à partir desquelles les acquisitions de matériel peuvent être faites, ainsi que sur les entités d'emplacements correspondantes requises pour les procédés de fabrication.

L'ensemble de règles 2 identifie l'exigence par lot des entités individuelles de matériel qui sont nécessaires pour la production d'un lot unitaire du produit considéré. Cette identification des besoins de lot est basée sur les besoins matériels du segment produit exigé par le client.

L'ensemble de règles 3 identifie les besoins du procédé. Le modèle unifié des procédés fondamentaux pour la fabrication du produit est déjà identifié à l'intérieur de l'artefact ontologique.

Les règles 4, 5 et 6 affectent le produit demandé par le client avec l'ensemble des entités possibles d'emplacement disponibles, où les processus d'approvisionnement, de fabrication, d'assemblage et d'emballage peuvent être effectués. Le rendement fondamental de ce système décisionnel basé sur l'ontologie est l'ensemble des entités possibles d'emplacement à partir

desquelles les demandes clients pour les segments de produits peuvent être satisfaites. Il est important ici de préciser que cette approche ne fournit pas un algorithme de planification pour le réseau, mais une information pertinente sur laquelle un algorithme d'ordonnement de la chaîne logistique réseau est basé. Cet algorithme d'ordonnement est l'une des perspectives de cette thèse.

Les Figure 37 et Figure 38 montrent et expliquent la nature des informations extraites de l'artefact ontologique à l'aide de l'ensemble de règles de requête portant sur une demande individuelle de segments de produit sous les contraintes de processus, des fournisseurs disponibles au sein de l'entreprise virtuelle et du réseau de distances correspondant aux ressources matérielles disponibles ainsi que les ressources d'équipement nécessaires et disponibles pour l'activité de production individuelle.

L'information extraite identifie également la durée du procédé et l'efficacité du processus d'équipement individuel. Cette information sert de base pour les opérations supplémentaires de planification reposant sur des préférences d'efficacité de la manutention de matériaux ou de la distance de transport minimale. Par exemple, dans la Figure 37, la Production Request 1 exigeant un lot unitaire du Product Segment 3.

Les règles déterminent les lieux où se trouvent une ressource disponible à la première étape de fabrication et les distances entre les deux entités de ressources : le fournisseur de classe de matériel SC_BB et les fabricants de premier niveau. Les règles servent également à déterminer l'efficacité du processus dès la première phase de fabrication ainsi que les temps associés au processus de fabrication dès la première étape, cette information peut être utilisée sur celui-ci aux fins de planification dans l'entreprise virtuelle.

En outre, les règles extraient les informations suivantes, à l'aide de la propriété d'objet transitif « has_Customer »¹⁵ : l'instance emplacement de l'équipement disponible pour le processus d'assemblage, ainsi que les informations de distance entre le lieu de fabrication de la première étape et l'emplacement de la deuxième étape de montage.

¹⁵ Discutés en détail dans la phase de modélisation de l'artefact d'ontologie.

Production Request 1					
Product Demand Date & Time		05-mars-12	1100		
Product Segment ID	Customer Location ID				
3	19				
Material Class SC_BB Lot Quantity		1			
Material Class SC_BA Lot Quantity		2			
Material Class SC_A Lot Quantity		1			
Supplier Location	Manufacture Location	Resource Efficiency	Process Time	(ε) Distance S-M	
4	9	80	06:00:00	63	
4	10	76	06:00:00	66	
4	11	90	06:00:00	34	
4	12	68	06:00:00	98	
Manufacture Location	Assembly Location	(α) Distance M-A			
9	13	95			
9	14	73			
9	15	40			
10	13	47			
10	14	50			
10	15	68			
11	13	50			
11	14	56			
11	15	60			
12	13	64			
12	14	40			
12	15	50			
Supplier Location	Assembly Location	Resource Efficiency	Process Time	(ζ) Distance S-A	
6	13	80	04:00:00	120	
6	14	75	04:00:00	122	
6	15	68	04:00:00	45	
7	13	80	04:00:00	230	
7	14	75	04:00:00	200	
7	15	68	04:00:00	100	
Assembly Location	Packaging Location	(β) Distance A-P			
13	16	80			
13	17	100			
14	16	69			
14	17	50			
15	16	60			
15	17	80			
14	17	50			
15	16	60			
15	17	80			
Supplier Location	Package Location	Resource Efficiency	Process Time	(η) Distance S-P	
8	16	90	01:00:00	400	
8	17	95	01:00:00	70	
8	17	95	01:00:00	70	
Package Location	Customer Location	(γ) Distance P-C			
16	19	70			
17	19	60			

Figure 37 : Données de configuration extraites de l'artefact ontologique pour la demande de production

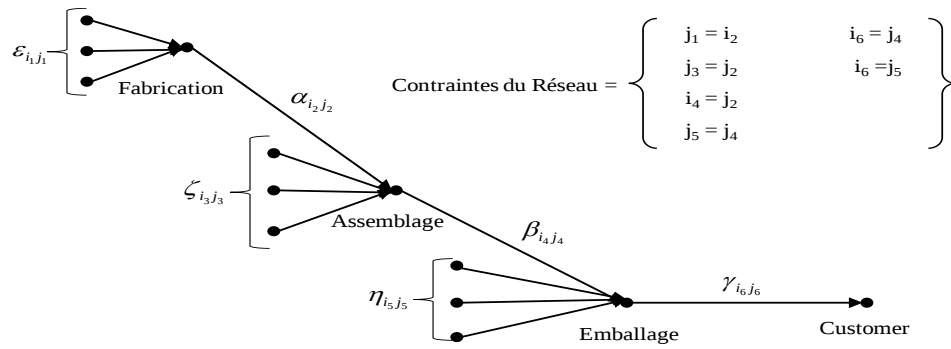


Figure 38 : Utilité de l'information extraite par l'artefact d'ontologie dans le réseau logistique

Cette propriété de l'objet transitif représente le flux physique unidirectionnel des marchandises, à partir de leur emplacement initial fournisseur vers leur lieu de fabrication ultérieure. Cette approche de l'identification des flux unidirectionnels entre les entités du réseau contribue à la représentation logique de l'association entre les ressources du réseau. La relation de transitivité entre les individus d'emplacement est représentée par une relation `has_Customer`, mais celle-ci représente la valeur de la distance entre ces entités individuelles d'emplacement et nécessite l'expression de l'entité d'emplacement initiale, l'entité d'emplacement finale et la distance entre les deux entités, qui sont modélisées dans l'artefact ontologique et qui est discutés en détail dans la Figure 35.

Comme nous l'avons mentionné précédemment dans la description de l'étude de cas (cf. IV.2), la deuxième étape de la réalisation du produit est l'assemblage du produit partiellement fini de la première étape. L'ensemble de règles extrait l'efficacité et le temps associé à l'assemblage et permet aussi d'identifier les instances d'emplacement des fournisseurs de matériel disponibles de `SC_BA`, avec leurs contraintes de distance. À ce point le flux physique de produits partiellement finis de la première étape du processus de fabrication et le flux physique de la classe de matériau `SC_BA`, provenant de leur fournisseur sont réunis à l'emplacement des ressources d'assemblage. Les informations pour la planification de cette opération doivent identifier les réseaux de distances et l'efficacité de manutention parmi les équipements distribués ainsi que les délais de traitement associés, cette information est inférée par l'artefact ontologique à l'aide des règles appliquées sur celui-ci.

De même, pour le processus d'emballage tiers, les marchandises assemblées de la deuxième étape sont unies avec les fournisseurs disponibles de la classe du matériel d'emballage `SC_A`, pour obtenir le lot final exigé de la demande de production, qui peut ensuite être transporté vers le client final.

Ainsi, dans chaque intervalle de temps discret de la demande client en ce qui concerne le produit (l'identification du produit), la localisation du client et la date de la demande sont utilisés pour remplir, avec les données d'instance, la classe définie *Production_Schedule* au sein de l'ontologie. À cet intervalle de temps, l'état des ressources du réseau logistique est également mis à jour au sein de l'artefact ontologique. Avec l'aide de l'ensemble des règles définies précédemment, la localisation des fournisseurs, des ressources disponibles et des matériels sont identifiés pour chacune des instances de la classe définie *Production_Schedule*.

Les informations extraites de l'une des instances de la classe définie *Production_Schedule* sont représentées dans la Figure 37. Pour utiliser ces informations, nous proposons un algorithme représenté sur la Figure 39. Cet algorithme identifie la configuration du réseau à partir des informations extraites de l'artefact ontologique au sein de chaque intervalle de temps. Ainsi, l'artefact ontologique est peuplé avec la demande du segment de produit, ainsi qu'avec l'identification de la localisation du client. Les résultats de cet algorithme sont montrés en Figure 40 après l'application des contraintes du réseau, qui sont analysés aux Figure 37 et Figure 38. Dans le cas où les moyens de production ne sont pas disponibles pour cet intervalle de temps discret, la demande client est retardée à l'intervalle de temps suivant, comme indiqué dans l'algorithme.

Les résultats discutés ci-dessus, l'artefact ontologique modélisé et les règles de gestion appliquées sont utilisés comme base pour la décision de configuration au sein du réseau d'approvisionnement. Comme nous l'avons vu, la faiblesse actuelle du système de gestion des ressources d'entreprises modernes est leur incapacité à s'adapter en cas de changements relativement fréquents de la configuration du réseau. En effet, comme les petites et moyennes entreprises se concentrent sur leurs compétences de base, il leur devient difficile d'entrer dans un accord contractuel à long terme au sein d'une chaîne logistique conventionnelle. Ce que notre approche permet, c'est que ce type d'entreprises puisse facilement former une entreprise virtuelle et gérer efficacement l'intégration de l'information ainsi que l'intégration de ressources dans une stratégie efficiente, efficace, flexible et robuste aux changements dans la structure du réseau des ressources.

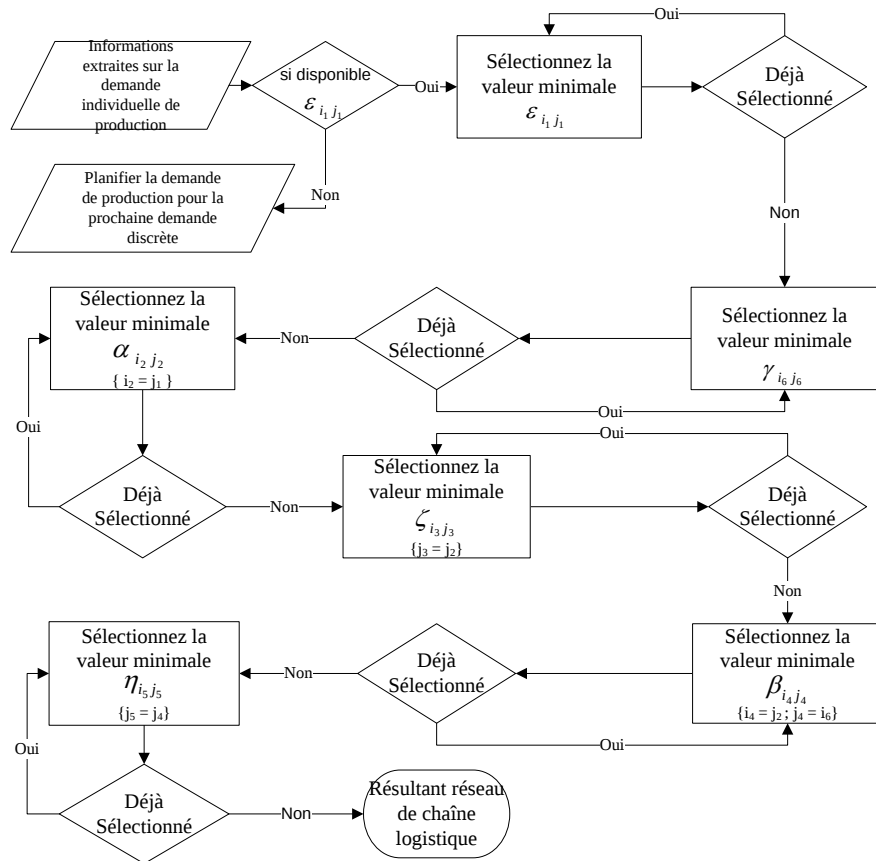


Figure 39 : Algorithme d'extraction de l'information de configuration du réseau logistique

Production Request 1				
Product Demand Date & Time		05-mars-12	1100	
Product Segment ID	Customer Location ID			
3	19			
Supplier Location (i_1)	Manufacture Location (j_1)	Resource Efficiency	Process Time	Distance $\epsilon_{i_1 j_1}$
4	11	90	06:00:00	34
Manufacture Location (i_2)	Assembly Location (j_2)	Distance $\alpha_{i_2 j_2}$		
11	13	50		
Supplier Location (i_3)	Assembly Location (j_3)	Resource Efficiency	Process Time	Distance $\zeta_{i_3 j_3}$
6	13	80	04:00:00	120
Assembly Location (i_4)	Packaging Location (j_4)	Distance $\beta_{i_4 j_4}$		
13	16	80		
Supplier Location (i_5)	Package Location (j_5)	Resource Efficiency	Process Time	Distance $\eta_{i_5 j_5}$
8	16	90	01:00:00	400
Package Location (i_6)	Customer Location (j_6)	Distance $\gamma_{i_6 j_6}$		
16	19	70		

Figure 40 : Résultats de l'algorithme de configuration du réseau logistique

IV.7 Conclusion:

Dans ce chapitre, nous avons montré l'utilité d'un artefact ontologique pour l'intégration des ressources distribuées de l'entreprise virtuelle. Ce résultat est obtenu grâce à une approche pour la modélisation de la structure du réseau logistique au sein de l'entreprise distribuée discutée en III.4. Nous fournissons ensuite une hiérarchie des ressources d'entreprise virtuelle dans la Figure 26. Le modèle ainsi obtenu identifie les ressources collectives présentes dans l'entreprise virtuelle établie. Par la suite, ce modèle est transformé en un artefact ontologique et finalement des ensembles de règles sont appliqués à cet artefact pour la configuration du réseau. Au final, une information utile à l'ordonnancement est extraite qui contient des ressources disponibles à l'instant de la demande et qui peuvent donc être utilisées pour la satisfaire. Les principales caractéristiques de l'approche proposée pour l'intégration des ressources distribuées sont un temps de mise en œuvre court et une flexibilité de la structure. Il s'agit de la proposition principale de cette thèse pour l'utilisation d'ontologies web sémantiques pour la configuration de processus et d'intégration des ressources distribuées. Les exigences fondamentales de la mise en œuvre de cette approche, basée sur l'ontologie, sont discutées et les difficultés correspondantes identifiées.

La procédure de planification pour chaque demande de production dépend entre autres des contraintes d'efficacité ou de temps de l'entreprise virtuelle, mais l'objectif principal de l'utilisation de l'artefact ontologique web sémantique pour configurer les ressources distribuées est montré. Il est important de noter ici que cette approche de l'intégration des ressources distribuées est réalisée au moyen de techniques basées sur le web pour minimiser les difficultés d'accessibilité. Il est également important de noter que le cycle de mise en œuvre de cette approche est rapide et que cela répond bien à la reconfiguration des ressources du réseau. En conséquence de nouvelles ressources peuvent être facilement ajoutées ou supprimées comme le montre la Figure 41 où « Assemblage 2 » a été inséré entre « Assemblage 1 » et « Customer » (voir Figure 38) avec, bien sur, ses propres contraintes de configuration.

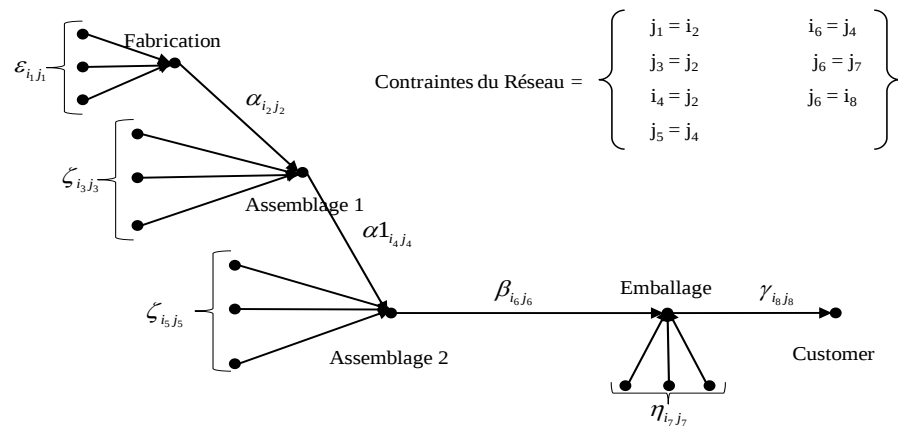


Figure 41 : Modèle de la structure d'un réseau complexe avec ses contraintes de configuration

Chapitre V

Conclusions et perspectives

Chapitre 5 : Conclusions et perspectives

V.1 Conclusion

Dans cette thèse, nous avons présenté un examen complet des technologies du web sémantique et leurs utilités correspondantes dans le contexte actuel pour les petites et moyennes entreprises. Les approches traditionnelles sur l'intégration des entreprises ont déjà fourni de bons résultats, mais en même temps les entreprises sont toujours sous « pression » pour pouvoir garder un niveau de compétitivité accru. L'entreprise distribuée, dans un tel environnement, a un nombre limité de choix, que ce soit de prendre des décisions stratégiques et entrer dans un accord à long terme avec leurs partenaires et de limiter leur flexibilité aux changements du marché ou d'utiliser les possibilités offertes par le web sémantique pour intégrer leurs ressources autour des produits qu'ils créent en collaboration. Par conséquent, elles se doivent de garder leurs compétences individuelles ainsi que leur flexibilité aux changements du marché.

Les technologies du web sémantique ont déjà prouvé leurs utilités dans l'intégration des artefacts d'informations et beaucoup de recherches ont été orientées vers l'exploitation de cette intégration des capacités fournies par les technologies du web sémantique comme en bio-informatique ou pour l'intégration des informations dans le domaine des soins de santé. Il y a longtemps que le secteur industriel utilise aussi ce pouvoir d'intégration de l'ontologie de web sémantique. La revue de la littérature fournie dans cette thèse présente les derniers développements dans le domaine de l'ingénierie industrielle, on peut citer la traçabilité de l'information du produit, la représentation de modèle d'information pour les composants du produit, etc.

Dans cette thèse, nous avons travaillé sur les questions fondamentales liées à l'ontologie qui se posent lorsque l'application des approches basées sur l'ontologie est utilisée dans la perspective de la chaîne logistique. Le premier problème présenté dans cette thèse est la délimitation des différents processus impliqués dans la gestion d'ontologies, une distinction claire fournie dans le Chapitre II est l'identification des différents termes utilisés dans les opérations de gestion d'ontologies, qui sont présentés dans la littérature distribuée dans les différents domaines. Le deuxième problème délimité dans la revue de la littérature est l'utilisation de l'ontologie centrale ou des artefacts distribués et modulaires d'ontologie. Grâce à la revue de la littérature, il est évident, dans le contexte de l'applicabilité de l'artefact

d'ontologie au sein de l'entreprise, qu'un artefact ontologique central est la seule option, car les capacités de raisonnement distribués ne sont pas encore suffisamment matures et ne prennent pas en charge l'application d'un artefact modulaire et distribué. Le troisième problème discuté en détail dans le Chapitre II est le processus actuel de modélisation d'artefacts ontologiques.

Aux vues de cette revue de la littérature, il est évident que la procédure de modélisation d'ontologie n'est pas une action individuelle, mais qu'elle exige la collaboration d'experts de domaines multiples. L'approche automatisée actuelle appliquée pour l'intégration des ontologies multiples nécessite toujours une intervention humaine pour l'évaluation définitive et la validation de l'artefact d'ontologie, donc un processus collaboratif pour la modélisation d'ontologie est important dans un environnement distribué.

De la discussion qui précède, il est clair que les ontologies sont utilisées comme un artefact d'intégration de base. Les capacités de raisonnement parmi les ontologies modulaires multiples sont limitées et la recherche concernant leur utilité dans l'entreprise distribuée collaborative est inexistante, mais elle va se développer lorsque des raisonnements plus acceptables seront mis en place. Dans l'ensemble, les utilisations pratiques des ontologies OWL-DL sont des artefacts centraux, car les services de raisonnement d'ontologie comme test de consistance ou de la satisfiabilité de concept ne sont fournis que dans une telle configuration centrale. Dans le Chapitre III, nous avons fourni le thème central de l'intégration de l'information à travers l'artefact d'ontologie dans le contexte de l'entreprise. Dans l'entreprise virtuelle, des partenaires multiples, non seulement, exigent l'intégration de leur matériel distribué et de leurs ressources matérielles, mais aussi l'intégration de leurs artefacts d'information qui sont créés au sein des entités collaborant. Pour répondre à cette modélisation d'ontologie, et ultérieurement à l'intégration des informations dans le cadre de l'entreprise virtuelle, nous avons :

- Discuté d'abord d'un processus conceptuel pour la modélisation collaborative de l'artefact ontologique pour l'entreprise virtuelle ;
- Après ce processus, nous avons identifié un modèle d'information au sein de l'entreprise virtuelle ;
- Puis, nous avons défini une architecture pour le système de gestion d'ontologies et de contenu, ce qui permet de gérer les opérations d'intégration du contenu nouvellement créés avec l'ontologie modélisée, au sein de l'entreprise virtuelle ;

- Dans l'identification de ce système de gestion de contenu et d'ontologie, les différents composants sont définis avec leurs responsabilités fonctionnelles correspondantes, ainsi que la façon dont ils interagissent les uns avec les autres pour atteindre l'objectif final de l'intégration du contenu créé au sein de l'entreprise virtuelle avec l'artefact d'ontologie modélisée.

La méthodologie de recherche pour identifier les composants de l'architecture discutée ci-dessus est effectuée à partir du processus de la modélisation de l'ontologie courante ainsi que des techniques actuelles qui sont utilisés pour l'intégration de l'artefact d'information avec l'artefact ontologique.

Les dernières parties du Chapitre III, traitent de l'utilisation des artefacts ontologiques comme des artefacts de configuration de l'équipement distribué et des ressources matérielles distribuées de l'entreprise virtuelle. Pour cet objectif, les systèmes à base de règles sont également inclus dans la phase de mise en œuvre de la recherche.

Pour l'utilisation d'un artefact d'ontologie central, les exigences de base sont, premièrement, l'identification d'un modèle pour représenter les ressources physiques au sein de l'artefact d'ontologie. Dans ce but le modèle d'ISA-95 est utilisé, qui identifie les ressources de l'entreprise de base et leur hiérarchie correspondante. L'utilité de la norme industrielle, à cet égard, est d'atténuer la différence de la modélisation des ressources au sein de l'artefact d'ontologie pour l'entreprise distribuée. Deuxièmement, il faut identifier une architecture de mise en œuvre, cette architecture est discutée dans le Chapitre III.

Dans ce même Chapitre III, nous avons discuté de l'approche de modélisation pour le réseau des entités de la chaîne logistique, ce qui est la suite de l'objectif ci-dessus en utilisant des d'ontologies pour l'intégration de l'équipement des entreprises et des ressources matérielles. Nous avons également fourni un modèle pour la représentation d'ontologie. Pour cela, l'approche basée sur les processus est une méthode dominante dans la modélisation du réseau logistique. Cette approche permet la décomposition d'un processus complexe en processus plus simples, avec des unités logiques appropriées permettant d'expliquer leurs opérations concernées, leurs attributs, leurs entrées, leurs sorties et leurs interrelations avec les autres processus adjacents. Cette approche identifie également au sein de chaque processus, les ressources nécessaires à l'exécution des processus ainsi que des contraintes potentielles pertinentes pour les processus définis. Dans les réseaux logistiques, les processus sont

fortement couplés, par conséquent, la modélisation des processus est une base essentielle de l'approche dans la modélisation des réseaux logistiques.

Nous fournissons également dans ce chapitre la méthodologie d'intégration des ressources. Par la suite, cette méthode est validée par une étude de cas, décrites dans le Chapitre IV. Notre démarche porte sur l'ontologie sémantique web pour représenter l'utilité des ressources physiques au sein d'un réseau distribué d'entreprises et pour fournir un cadre pour l'intégration, basée sur l'ontologie, de la fabrication et des ressources d'approvisionnement. Pour cela, nous délimitons la conversion du modèle standard pour l'intégration d'entreprise (ISA-95) dans une ontologie basée sur la Logique Descriptive. Par la suite, nous appliquons des règles métiers afin de fournir à la base de connaissances, qui agit comme un système d'aide à la décision pour le processus d'approvisionnement du matériel en termes de fournisseur et d'exigences matérielles de produit, des résultats de cette connaissance dans un artefact central, capables d'intégrer les interactions de l'entreprise distribuée. Les règles de gestion développées ont comme portée la limitation des produits, des procédés, de la logistique, du matériel, mais nous pouvons, également, développer plus la règle portant sur la logique d'affaires.

L'idée principale de la flexibilité du réseau concerné dans cette thèse est la flexibilité de l'utilité des ressources physiques et des ressources matérielles. Cet accent mis sur la flexibilité du réseau constituera une incitation pour les PME à intégrer leurs ressources distribuées et, en définitive, à améliorer la flexibilité de livraison du réseau logistique. Ce qui permettra, entre autre, de réduire l'écart parmi le réseau d'entreprises distribuées.

Dans cette thèse, nous nous sommes directement concentrés sur l'extrémité amont de la chaîne logistique ainsi que sur les entités de transformation du réseau logistique, tels que le réseau d'approvisionnement et le réseau des équipements de transformation de matériel. D'autres entités réseau doivent être intégrées dans les approches basées sur les ontologies, comme dans les réseaux logistiques modernes. Ainsi, les deux concepts du cycle de vie produit et du cycle de vie final (recyclage) du produit sont pertinents dans l'introduction de nouveaux produits sur le marché. Les principaux obstacles sont la caractérisation et la standardisation du modèle de données correspondant à ces concepts ainsi que l'aide à la décision concernant l'intégration du produit à ces différentes étapes afin de proposer un meilleur ordonnancement et un meilleur recyclage. Les approches basées sur l'ontologie peuvent aider à une meilleure compréhension de ces processus et de leurs interrelations logiques. La méthodologie proposée

dans cette thèse aura une influence directe en outre, sur la mise en œuvre pratique dans les systèmes de gestion du cycle de vie des produits, ainsi que dans le domaine de la chaîne logistique verte dans laquelle la représentation logique du réseau inverse, dans le sens du recyclage du produit, peut également être représentée.

Avec l'intégration croissante des entreprises, les obstacles rencontrés proviennent essentiellement de l'information et des limites du partage de cette information.

V.2 Perspectives

Les perspectives de cette thèse concernant l'application d'ontologie sémantique web seraient de passer de domaines d'application spécifiques à des domaines plus généraux. Comme nous l'avons mentionné précédemment, l'approche présentée pour l'intégration de ressources distribuées d'entreprise peut également être utilisée pour l'intégration des ressources impliquées dans la logistique inverse. Pour cela, l'approche actuelle, permettant d'identifier les normes de représentation des ressources au sein de la logistique inverse, leur modélisation au sein de l'artefact central de l'ontologie sémantique web et l'utilisation de cet artefact central ontologique pour la décision de la configuration, peut être utilisée. Dans cette thèse, nous avons pris en compte le point de vue de l'entreprise virtuelle, mais cette approche peut aussi être projetée vers le bas, au niveau de l'entreprise individuelle, dans la gestion de ces ressources locales au niveau de l'atelier.

Une autre perspective est dans le processus de modélisation d'ontologie, car les processus actuels pour la modélisation d'ontologie, se concentrent essentiellement sur un coordinateur central. La modélisation d'ontologie a besoin d'être un processus de collaboration afin de capturer les sémantiques combinées du domaine de la conceptualisation. Nous avons proposé un tel processus collaboratif dans le Chapitre III. Comme la modélisation collaborative de l'ontologie nécessite la compréhension des changements qui se sont produits au sein de l'artefact d'ontologie, l'obstacle principal à sa mise en œuvre pratique est l'absence de consentement dans les capacités de représentation de l'éventuel langage d'ontologie. Comme les capacités des langages de représentation des d'ontologie évoluent, cela nécessite plus de recherche dans le domaine de la gestion du changement d'ontologie discuté dans le Chapitre II. Voici quelques-unes des perspectives immédiates qui peuvent être appliquées à la suite de nos recherches. Certaines plus générales sur l'utilisation d'ontologie sont discutées par la suite. Le langage d'ontologie du web (OWL) évolue dans le même temps et depuis le début de cette thèse, OWL-2 a été introduit permettant un plus grand niveau d'expressivité logique.

Nous pouvons noter par exemple, comme perspectives directes de nos travaux, que :

1. Les ontologies peuvent aussi être utilisées dans l'analyse des risques dans des processus de fabrication différents utilisant de hautes technologies, par exemple, qui reposent fortement sur la fiabilité et la performance de l'historique des équipements de fabrication. Les ontologies peuvent être aussi utilisées pour intégrer l'historique des données de performances pour prévoir la performance future de l'équipement. Cette approche permet aux opérateurs de minimiser leurs pertes de fabrication par l'utilisation de meilleurs outils d'aide à la décision. L'approche proposée de l'intégration du contenu et de l'information ontologique peut être utile à cet égard.
2. Le processus de conception technique est une tâche qui nécessite beaucoup de connaissances et qui exige que les équipes de concepteurs travaillent simultanément. Les approches basées sur l'ontologie peuvent être utilisées pour intégrer le processus de conception technique du produit avec ses artefacts d'information correspondants au cours de la phase de conception (Brandt S.C. *et al.*, 2008).
3. La définition formelle de l'ontologie et la conceptualisation du domaine d'intérêt traitent dans certaines circonstances de concepts flous. Cette conceptualisation floue est importante dans de nombreux domaines d'intérêt, comme la réponse à une requête ou dans l'analyse des risques. La recherche sur la modélisation du comportement floue dans l'ontologie peut être aussi une perspective de recherche intéressante.
4. La formalisation des connaissances, dans les ontologies, joue un rôle important dans l'alignement stratégique des différentes entités commerciales, ainsi que dans les processus d'affaires. Cet alignement doit être à un niveau sémantique de telle sorte qu'une meilleure compréhension des entités commerciales et des processus d'affaires peut aider à avoir une réponse plus efficace et flexible.
5. L'intégration des sources hétérogènes entre les différents partenaires peut être résolue avec les propriétés d'intégration des ontologies. Les sources d'information hétérogènes peuvent être classées en termes de l'hétérogénéité syntaxique ou l'hétérogénéité sémantique. Dans les deux cas, les ontologies peuvent être utilisées pour mieux réduire l'écart sémantique au sein des partenaires.

Concernant les perspectives plus générales, nous pouvons citer, entre autres, que :

1. L'utilisation d'ontologie comporte également la représentation de l'innovation au sein de l'entreprise virtuelle. L'innovation concerne généralement l'innovation des processus dans l'environnement virtuel. Les partenaires sont en permanence confrontés à la prise de décision locale qui influe sur la production générale de l'entreprise virtuelle et la conceptualisation du domaine de cette innovation locale est importante dans le domaine de recherche. L'approche actuelle des innovations suivies par SAP : (1) inventer, (2) définir, (3) développer, (4) déployer, (5) optimiser ne saisit pas tous les détails complexes que représente le processus d'innovation. Ainsi, pour une meilleure compréhension de l'innovation d'un processus, cette approche doit être centrée autour de l'acquisition, la compréhension, l'organisation, l'intégration, l'enrichissement, la distribution, la validation d'une grande quantité de connaissances qui sont généralement organisées autour d'un ensemble prédéfini de documents.
2. L'application d'ontologies réside aussi dans l'intégration de l'information publiée par les instances gouvernementales. Ces données publiées sont généralement au format XML et sont distribuées à travers de multiples entités gouvernementales. Les procédures décisionnelles pour la compréhension du problème sont basées sur ces ensembles de données distribuées. La même logique s'applique à l'intégration de la gestion de documents et de contenus à l'intérieur ou à travers de multiples entités gouvernementales. Grâce à une meilleure intégration sémantique de la procédure et du contenu associé, les entités gouvernementales peuvent rationaliser leurs opérations de gestion interne. Par exemple dans la gestion du commerce transfrontalier, la mise en œuvre de la limitation gouvernementale sur le partenaire commercial ou l'importation et l'exportation des marchandises, les autorités exigent une information complète des composants et doivent gérer cette information au travers d'entités transfrontalières multiples. L'approche basée sur l'ontologie peut être utilisée pour modéliser le produit. Cela peut fournir une information complète du produit et de sa ressource fournisseur, permettant ainsi une gestion du risque au niveau d'un seul pays ou au niveau d'une fédération.
3. Comme pour des données gouvernementales, les ontologies peuvent également être utilisées dans l'analyse des risques au sein des différentes entités. Par exemple, les processus de fabrication nécessitant de hautes technologies comptent fortement sur la fiabilité et la performance historique des équipements de fabrication. Les ontologies peuvent être utilisées pour intégrer les historiques de données de performance pour prévoir la performance future de l'équipement. Cette approche permet aux opérateurs

de meilleurs outils de soutien décisionnels en minimisant leur perte de fabrication. Notre approche propose un contenu et des informations ontologiques qui peuvent être utiles dans ce domaine.

4. Les approches basées sur l'ontologie se révèlent également utiles dans la modélisation de systèmes multi-agents. Dans la plupart des méthodes de simulation, des agents multiples nécessitent une description du monde en constante évolution. Par exemple, on peut citer la description des situations, la description du logiciel, la description des objets informationnels, la description des plans, etc. Cette description peut être efficacement modélisée dans des modèles ontologiques que les multiples agents peuvent alors utiliser efficacement dans leurs processus décisionnels.

Bibliographie

Bibliographie

- Adler, A., Nash, J.C. & Noël, S., (2006). Evaluating and Implementing a Collaborative Office Document System. In *Interacting with Computers*, 18(4): 665-682.
- Akkermans, H.A. *et al.*, (2003). The impact of ERP on supply chain management: Exploratory findings from a European Delphi study. In *European Journal of Operational Research*, 146(2), 284-301.
- Amjad F., Roy D., Anciaux D., (2009). Authoring tool architecture and Workflow for collaborative development of semantic documents and learning objects. *6th International Conference on Knowledge Management*, Honk Kong, China.
- Amjad F., Roy D. and Anciaux D., (2010). Framework for knowledge management in collaborative distributed enterprise. *MOSIM'10 : 8ème Conférence Internationale de Modélisation et Simulation*, Hammamet, Tunisie.
- Amjad F., Roy D. and Anciaux D., (2011). Ontology maturation workflow and authoring tool architecture for the collaborative creation of semantic document in a distributed environment. *IESM'11: Industrial Engineering and Systems Management*, Metz France.
- Baader, F. *et al.*, (2010). *The Description Logic Handbook: Theory, Implementation and Applications* 2 éd., New York, NY, USA: Cambridge University Press.
- Baader, F. & Hollunder, B., (1991). KRIS: Knowledge Representation and Inference System. In *SIGART Bull.*, 2: 8-14.
- Baader, F. & Sattler, U., (2001). An Overview of Tableau Algorithms for Description Logics. In *Studia Logica*, 69: 5-40.
- Bock, C. *et al.*, (2010). Ontological product modeling for collaborative design. In *Advanced Engineering Informatics*, 24(4): 510-524
- Brachman, R.J. & Schmolze, J.G., (1985). An overview of the KL-ONE Knowledge Representation System. In *Cognitive Science*, 9(2): 171-216.
- Brandt, S.C. *et al.*, (2008). An ontology-based approach to knowledge management in design processes. In *Computers & Chemical Engineering*, 32(1–2): 320-342.
- Braun, S., Schmidt, A. & Walter, A., (2007). Ontology maturing: a collaborative web 2.0 approach to ontology engineering. *16th International World Wide Web Conference WWW2007 Banff, Canada*, 273.
- Buffa, M. *et al.*, (2008). SweetWiki: A semantic wiki. In *Web Semantics: Science, Services and Agents on the World Wide Web*, 6(1): 84-97.
- Calvanese, D., De Giacomo, G. & Lenzerini, M., (2002). A framework for ontology integration. In *The Emerging Semantic Web—Selected Papers from the First Semantic Web Working Symposium*, 201-214.
- Chen, Y., Chen, Y. & Chu, H., (2009). Development of a mechanism for ontology-based product lifecycle knowledge integration. In *Expert Systems with Applications*, 36(2, Part 2): 2759-

2779.

- Chi, Y.L., (2010). Rule-based ontological knowmedge base for monitoring partners across supply networks. In *Expert Systems with Applications*, 37(2): 1400-1407.
- Choi, N., Song, I. & Han, H., (2006). A survey on ontology mapping. In *SIGMOD Rec.*, 35: 34-41.
- Chu, H.C., Chen, M.Y. & Chen, Y.M., (2009). A semantic-based approach to content abstraction and annotation for content management. In *Expert Systems with Applications*, 36(2P1): 2360-2376.
- Dassisti, M., Panetto, H. & Tursi, A., (2006). Product-Driven Enterprise Interoperability for Manufacturing Systems Integration. In *Business Process Management Workshops*. Lecture Notes in Computer Science. Springer Berlin / Heidelberg,
- Davis, A.H., Sun, C. & Lu, J., (2002). Generalizing operational transformation to the standard general markup language. *CSCW '02 Proceedings of the 2002 ACM conference on Computer supported cooperative work*, 58-67.
- Decouchant, D. *et al.*, (2009). Contextual awareness based communication and coauthoring proximity. In the internet. In *Expert Systems with Applications*, 36(4): 8391-8406.
- Dotsika, F., (2009). Uniting formal and informal descriptive power: Reconciling ontologies with folksonomies. In *International Journal of Information Management*.
- Eriksson, H., (2007). The semantic-document approach to combining documents and ontologies. In *International journal of human-computer studies*, 65(7): 624-639.
- Flouris, G. *et al.*, (2006). Inconsistencies, negations and changes in ontologies. In *proceedings of the 21st national conference on Artificial intelligence*, 2. 1295-1300.
- Flouris, G. *et al.*, (2008). Ontology change: classification and survey. In *The Knowledge Engineering Review*, 23(02): 117-152.
- Friedman, E., (2003). *Jess in action: rule-based systems in java*. Manning Publications Co. ISBN: 1930110898
- De la Fuente, M., Ros, L. & Ortiz, A., (2010). Enterprise modeling methodology for forward and reverse supply chain flows integration. In *Computers in Industry*, 61(7): 702-710.
- Gaeta, M., Orciuoli, F. & Ritrovato, P., (2009). Advanced ontology management system for personalized e-Learning. In *Knowledge-Based Systems*, 22(4): 292-301.
- Giménez, D.M. *et al.*, (2008). PProduct ONTOlogy: Defining product-related concepts for logistics planning activities. In *Computers in Industry*, 59(2-3): 231-241.
- Gómez-Pérez, A., Fernández-López, M. & Corcho, O., (2004). The Most Outstanding Ontologies. In *Ontological Engineering*. Advanced Information and Knowledge Processing. Springer Berlin Heidelberg, 47-106.
- Gordon, L. Brown, (2004). The Ontology of Learning Environments. IACR Ontology of Learning Environments.

- Grau, B.C., Parsia, B. & Sirin, E., (2006). Combining OWL ontologies using-Connections. In *Web Semantics: Science, Services and Agents on the World Wide Web*, 4(1): 40-59.
- Grau, B.C. et al., (2008). OWL 2: The next step for OWL. In *Web Semantics: Science, Services and Agents on the World Wide Web*, 6(4): 309-322.
- Grubic, T. & Fan, I.S., (2010). Supply chain ontology: Review, analysis and synthesis. In *Computers in Industry*, 61(8): 776-786.
- Haarslev, V. & Müller, R., (2001). RACER System Description. In R. Goré, A. Leitsch, & T. Nipkow, éd. *Automated Reasoning*. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 701-705.
- Hardwick, M. & Bolton R., (1997). The Industrial. Virtual Enterprise, Communications of the ACM, 40: 59-60.
- Harland, C.M., (1996). Supply Chain Management: Relationships, Chains and Networks. In *British Journal of Management*, 7, S63-S80.
- Horrocks, I., (1998). The FaCT System. In. *Automated Reasoning with Analytic Tableaux and Related Methods*. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 307-312.
- Horrocks, I., Sattler, U. & Tobies, S., (2000). Practical reasoning for very expressive description logics. In *Logic Journal of IGPL*, 8(3): 239.
- Horrocks, I., Patel-Schneider, P.F. & van Harmelen, F., (2003). From SHIQ and RDF to OWL: the making of a Web Ontology Language. In *Web Semantics: Science, Services and Agents on the World Wide Web*, 1(1): 7-26.
- Hustadt, U., Motik, B. & Sattler, U., (2005). Data complexity of reasoning in very expressive description logics. In *Proceedings of the 19th international joint conference on Artificial intelligence*. IJCAI'05. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.,
- Jason J., J., (2009). Semantic business process integration based on ontology alignment. In *Expert Systems with Applications*, 36(8): 11013-11020.
- John, Zachman, (1987). Concepts of the framework for enterprise architecture. In *IBM Systems Journal*, 26(3).
- Juhnyoung Lee, Goodwin Richard & Akkiraju Rama, (2006). Ontology Management for Large-Scale Enterprise Systems. In *Web Semantics & Ontology*, ISBN13: 9781591409052, 91-114.
- Kahan, J. et al., (2002). Annotea: an open RDF infrastructure for shared Web annotations. In *Computer Networks*, 39(5): 589-608.
- Kalfoglou, Y. & Schorlemmer, M., (2003). Ontology mapping: the state of the art. In *The knowledge engineering review*, 18(01): 1-31.
- Kang, D. et al., (2010). An ontology-based Enterprise Architecture. In *Expert Systems with Applications*, 37(2): 1456-1464.

- Khan, W.A., Raouf, A. & Cheng, K., (2011). *Virtual Manufacturing*, Springer. ISBN: 978-0-85729-185-1
- Kim, K.Y., Manley, D.G. & Yang, H., (2006). Ontology-based assembly design and information sharing for collaborative product development. In *Computer-Aided Design*, 38(12): 1233-1250.
- Kim, M. & Compton, P., (2006). The perceived utility of standard ontologies in document management for specialized domains. In *International Journal of Human-Computer Studies*, 64(1): 15-26.
- Kiyavitskaya, N. *et al.*, (2009). Cerno: Light-weight tool support for semantic annotation of textual documents. In *Data & Knowledge Engineering*, 68(12): 1470-1492.
- Klein, M. *et al.*, (2004). *Change Management for Distributed Ontologies*. ISBN 90-9018400-7.
- Ku, K.C., Wensley, A. & Kao, H.P., (2008). Ontology-based knowledge management for joint venture projects. In *Expert Systems with Applications*, 35(1-2): 187-197.
- Lin, H. & Harding, J., (2007). A manufacturing system engineering ontology model on the semantic web for inter-enterprise collaboration. In *Computers in Industry*, 58(5): 428-437.
- Little, Eric G. & Rogova, Galina L., (2009). Designing ontologies for higher level fusion. In *Information Fusion*, 10(1): 70-82.
- Márque, C., (2010). Dynamic Modelling for Supply Chain Management. *Dealing with Front-end, Back-end and Integration Issues* 1er éd., Springer.
- Matsokis, A. & Kiritsis, D., (2010). An ontology-based approach for Product Lifecycle Management. In *Computers in Industry*, 61(8): 787-797.
- McGuinness, D.L. & van Harmelen, F., (2004). Owl web ontology language overview, w3c recommendation 10 february 2004. *World Wide Web Consortium*.
- McGuinness, D.L., Van Harmelen, F. & others, (2004). OWL web ontology language overview. *W3C recommendation*, 10, 2004-03.
- Meseguer, J. & Qian, X., (1993). A logical semantics for object-oriented databases. In *ACM SIGMOD Record*, 22(2): 89-98.
- Meyer, T., Lee, K. & Booth, R., (2005). Knowledge integration for description logics. In *Proceedings of the 20th national conference on Artificial intelligence - Volume 2*. AAAI Press, 645-650..
- Mikroyannidis, A. & Theodoulidis, B., (2010). Ontology management and evolution for business intelligence. In *International Journal of Information Management*, 30(6): 559-566.
- Monteiro, T., Roy, D. & Anciaux, D., (2007). Multi-site coordination using a multi-agent system. In *Computers in Industry*, 58(4): 367-377.
- Nešić, S., (2009). Semantic Document Model to Enhance Data and Knowledge Interoperability. Dans V. Devedžić & D. Gašević, éd. *Web 2.0 & Semantic Web*. Annals of Information Systems. Springer US, 135-160.
- Noy, N. *et al.*, (2006). A Framework for Ontology Evolution in Collaborative Environments. Dans I.

- Cruz et al., éd. *The Semantic Web - ISWC 2006*. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 544-558.
- Noy, N.F. & Musen, M.A., (2004). Ontology Versioning in an Ontology Management Framework. In *IEEE Intelligent Systems*, 19: 6-13.
- Patel-Schneider, P. & Horrocks, I., (1999). DLP and FaCT. Dans N. Murray, éd. *Automated Reasoning with Analytic Tableaux and Related Methods*. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 660-660.
- Pirro, G., Mastroianni, C. & Talia, D., (2010). A framework for distributed knowledge management: Design and implementation. In *Future Generation Computer Systems*, 26(1): 38-49.
- Plessers, P., (2006). *An approach to web-based ontology evolution*. PhD thesis, Vrije Universiteit Brussel.
- Plessers, P., De Troyer, O. & Casteleyn, S., (2007). Understanding ontology evolution: A change detection approach. In *Web Semantics: Science, Services and Agents on the World Wide Web*, 5(1): 39-49.
- Plessers, P. & De Troyer, O., (2005). Ontology Change Detection Using a Version Log. In *The Semantic Web – ISWC 2005*. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 578-592.
- Poggi, A., Lembo, D., Calvanese, D., De Giacomo, G., Lenzerini, M., et Rosati, R., "Linking data to ontologies," *Journal on data semantics X*, 2008, p. 133–173.
- Price, S.L. et al., (2009). Using semantic components to search for domain-specific documents: An evaluation from the system perspective and the user perspective. In *Information Systems*, 34(8): 724-752.
- Puigjaner, Luis & Laínez, José Miguel, (2008). Capturing dynamics in integrated supply chain management. In *Computers & Chemical Engineering*, 32(11): 2582-2605.
- R, T. & Gruber, (1993). A translation approach to portable ontology specifications. In *Knowledge Acquisition*, 5(2): 199-220.
- Rajsiri, V. et al., (2010). Knowledge-based system for collaborative process specification. In *Computers in Industry*, 61(2): 161-175.
- Ressel, M., Nitsche-Ruhland, D. & Gunzenhäuser, R., (1996). An integrating, transformation-oriented approach to concurrency control and undo in group editors, 288-297.
- Santoro, F.M., Borges, M.R. & Rezende, E.A., (2006). Collaboration and knowledge sharing in network organizations. In *Expert Systems with Applications*, 31(4): 715-727.
- Schmidt-Schauß, M. & Smolka, G., (1991). Attributive concept descriptions with complements. In *Artificial Intelligence*, 48(1): 1-26.
- Sebastian, A. et al., (2008). A Generic Ontology for Collaborative Ontology-Development Workflows, 318-328.
- Serafini, L. & Tamin, A., Drago (2005) Distributed reasoning architecture for the semantic web. In

- The Semantic Web: Research and Applications*, 361-376.
- Simatupang, T.M., Wright, A.C. & Sridharan, R., (2002). The knowledge of coordination for supply chain integration. In *Business Process Management Journal*, 8(3): 289-308.
- Sirin, E. *et al.*, (2007). Pellet: A practical OWL-DL reasoner. In *Web Semantics: Science, Services and Agents on the World Wide Web*, 5(2): 51-53.
- Stefanovic, D., Stefanovic, N. & Radenkovic, B., (2009). Supply network modelling and simulation methodology. In *Simulation Modelling Practice and Theory*, 17(4): 743-766.
- Sun, C. *et al.*, (1998). Achieving convergence, causality preservation, and intention preservation in real-time cooperative editing systems. In *ACM Transactions on Computer-Human Interaction (TOCHI)*, 5(1): 63-108.
- Sun, C., (2002). Optional and responsive fine-grain locking in Internet-based collaborative systems. In *IEEE Transactions on Parallel and Distributed Systems*, 13(9): 994-1008.
- Sun C. *et al.*, (2006). Transparent Adaptation of Single-User Applications for Multi-User Real-Time Collaboration. In *ACM Transactions on Computer-Human Interaction*, 13(4): 531-582.
- Tudorache, T. *et al.*, (2008). Supporting Collaborative Ontology Development in Protégé In *ISWC '08 Proceedings of the 7th International Conference on The Semantic Web Springer-Verlag Berlin, Heidelberg*, 32.
- Tursi, A. *et al.*, (2009). Ontological approach for products-centric information system interoperability in networked manufacturing enterprises. In *Annual Reviews in Control*, 33(2): 238-245.
- Urbán, Miguel Ángel Sicilia & Barriocanal, Elena García, (2003). Fuzzy Group Models for Adaptation in Cooperative Information Retrieval Contexts. In *PsychNology Journal*, 1(3): 242-255
- Uren, V. *et al.*, (2006). Semantic annotation for knowledge management: Requirements and a survey of the state of the art. In *Web Semantics: Science, Services and Agents on the World Wide Web*, 4(1): 14-28.
- Verdouw, C.N. *et al.*, (2010). Towards dynamic reference information models: Readiness for ICT mass customization. In *Computers in Industry*, 61(9): 833-844
- Wadhwa , S., Rao, K.S. & Chan, F.T.S., (2012). Flexibility-enabled lead-time reduction in flexible systems. In *International Journal of Production Research*, 43(15): 3131-3162.
- Adler, A., Nash, J. C., & Noël, S. (2006). Evaluating and implementing a collaborative office document system. *Interacting with Computers*, 18(4), 665–682.
- Aversano, L., De Lucia, A., Gaeta, M., Ritrovato, P., Stefanucci, S., & Luisa Villani, M. (2004). Managing coordination and cooperation in distributed software processes: the GENESIS environment. *Software Process: Improvement and Practice*, 9(4).
- Braun, S., Schmidt, A., Walter, A., Nagypal, G., & Zacharias, V. (2007). Ontology maturing: a collaborative web 2.0 approach to ontology engineering. Dans *Proceedings of the Workshop on Social and Collaborative Construction of Structured Knowledge at the 16th International World Wide Web Conference (WWW 07)*, Banff, Canada.

- C Sun et al. (2006). Transparent Adaptation of Single-User Applications for Multi-User Real-Time Collaboration. *ACM Transactions on Computer-Human Interaction*, 13(4), 531-582.
- Chen, Y., Chen, Y., & Chu, H. (2009). Development of a mechanism for ontology-based product lifecycle knowledge integration. *Expert Systems with Applications*, 36(2, Part 2), 2759-2779.
- Chi, Y. Ontology-Based Curriculum Content Sequencing System with Semantic Rules. *Expert Systems with Applications*, In Press, Accepted Manuscript. Can be found at http://www.sciencedirect.com/science?_ob=ArticleURL&_udi=B6V03-4V34D59-C&_user=8661858&_rdoc=1&_fmt=&_orig=search&_sort=d&_docanchor=&view=c&_searchStrId=1106012375&_rerunOrigin=scholar.google&_acct=C000026658&_version=1&_urlVersion=0&_userid=8661858&md5=01e552b9752bca81b031bde4bd5fe60a
- Davis, A. H., Sun, C., & Lu, J. (2002). Generalizing operational transformation to the standard general markup language. Dans *Proceedings of the 2002 ACM conference on Computer supported cooperative work* (pp. 58-67).
- De Nicola, A., Missikoff, M., & Navigli, R. (2009). A software engineering approach to ontology building. *Information Systems*, 34(2), 258-275.
- Decouchant, D., Escalada-Imaz, G., Martinez Enriquez, A. M., Mendoza, S., & Muhammad, A. (2009). Contextual awareness based communication and coauthoring proximity in the internet. *Expert Systems with Applications*, 36(4), 8391-8406.
- Eriksson, H. (2007). The semantic-document approach to combining documents and ontologies. *International Journal of Human-Computer Studies*, 65(7), 624-639.
- Gaeta, M., Orciuoli, F., & Ritrovato, P. . Advanced ontology management system for personalised e-Learning. *Knowledge-Based Systems*, In Press, Corrected Proof. Can be found at http://www.sciencedirect.com/science?_ob=ArticleURL&_udi=B6V0P-4VGPS8Y-4&_user=8661858&_rdoc=1&_fmt=&_orig=search&_sort=d&_docanchor=&view=c&_searchStrId=1106019407&_rerunOrigin=scholar.google&_acct=C000026658&_version=1&_urlVersion=0&_userid=8661858&md5=3408d122d5bf7c89cf220cfa5c46d815
- Guzmán-Arenas, A., & Cuevas, A; Knowledge accumulation through automatic merging of ontologies. *Expert Systems with Applications*, In Press, Corrected Proof. Can be found at http://www.sciencedirect.com/science?_ob=ArticleURL&_udi=B6V03-4WP47MJ-11&_user=8661858&_rdoc=1&_fmt=&_orig=search&_sort=d&_docanchor=&view=c&_searchStrId=1106020321&_rerunOrigin=scholar.google&_acct=C000026658&_version=1&_urlVersion=0&_userid=8661858&md5=912d44713391e6d8ce8d18738cf9424a
- Held, M., & Blochinger, W. (2009). Structured collaborative workflow design. *Future Generation Computer Systems*, 25(6), 638-653.
- Jean-Mary, Y. R., Shironoshita, E. P., & Kabuka, M. R.. Ontology matching with semantic verification. *Web Semantics: Science, Services and Agents on the World Wide Web*, In Press, Corrected Proof. Can be found on http://www.sciencedirect.com/science?_ob=ArticleURL&_udi=B758F-4W6Y5M7-1&_user=8661858&_rdoc=1&_fmt=&_orig=search&_sort=d&_docanchor=&view=c&_searchStrId=1105984076&_rerunOrigin=scholar.google&_acct=C000026658&_version=1&_urlVersion=0&_userid=8661858&md5=681f2012cca6c0b72aa975d3c2aef6f6

- Kaza, S., & Chen, H. (2008). Evaluating ontology mapping techniques: An experiment in public safety information sharing. *Decision Support Systems*, 45(4), 714-728.
- Klein, M. C. (2004). Change management for distributed ontologies. Thesis report 2004. Can be found at http://books.google.fr/books?hl=en&lr=&id=6lrgJ_7YeR4C&oi=fnd&pg=PA1&ots=_kXVkvjQnt&sig=ID7aQrCZ9s6bikNBpRUTtsm8zDk#v=onepage&q=&f=false
- Lee, J., & Goodwin, R. (2006). Ontology management for large-scale enterprise systems. *Electronic Commerce Research and Applications*, 5(1), 2-15.
- Lin, H., & Harding, J. (2007). A manufacturing system engineering ontology model on the semantic web for inter-enterprise collaboration. *Computers in Industry*, 58(5), 428-437.
- Nesic, S. (2008). Semantic Document Model to Enhance Data and Knowledge Interoperability. Can be found at <http://www.inf.usi.ch/phd/nesic/papers/SasaNesic.pdf>
- Nilsson, M., Palmer, M., Brase, J., & others. (2003). The LOM RDF binding-principles and implementation. Dans 3rd. Annual Ariadne Conference.
- Noy, N., Chugh, A., Liu, W., & Musen, M. (2006). A Framework for Ontology Evolution in Collaborative Environments. *The Semantic Web - ISWC 2006*.
- Stuckenschmidt, H., & Klein, M. (2007). Reasoning and change management in modular ontologies. *Data & Knowledge Engineering*, 63(2), 200-223.
- Tee, K., Greenberg, S., & Gutwin, C. (2006). Providing artifact awareness to a distributed group through screen sharing. Dans *Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work* (pp. 99–108).
- Tudorache, T., Noy, N. F., Tu, S., & Musen, M. A. (2008). Supporting Collaborative Ontology Development in Protégé. Dans *Proceedings of the 7th International Conference on The Semantic Web* (p. 32).
- Uren, V., Cimiano, P., Iria, J., Handschuh, S., Vargas-Vera, M., Motta, E., et al. (2006). Semantic annotation for knowledge management: Requirements and a survey of the state of the art. *Web Semantics: Science, Services and Agents on the World Wide Web*, 4(1), 14-28.
- Weal, M. J., Alani, H., Kim, S., Lewis, P. H., Millard, D. E., Sinclair, P. A., et al. (2007). Ontologies as facilitators for repurposing web documents. *International Journal of Human-Computer Studies*, 65(6), 537-562.
- Weibel, S., (1999). The state of the Dublin Core metadata initiative. In *Bulletin of the American Society for Information Science*, 25(5): 18-22
- Yan, J. et al., (2010). Ontology of collaborative manufacturing: Alignment of service-oriented framework with service-dominant logic. In *Expert Systems with Applications*, 37(3): 2222-2231.
- Yang, D., Dong, M. & Miao, R., (2008). Development of a product configuration system with an ontology-based approach. In *Computer-Aided Design*, 40(8): 863-878.

Yang, D. *et al.*, (2009). Product configuration knowledge modeling using ontology web language. In *Expert Systems with Applications*, 36(3, Part 1): 4399-4411.

Zurawski, R., (2006). *Integration Technologies for Industrial Automated Systems (Industrial Information Technology)*, CRC Press, Inc. ISBN: 0849392624

Annexes

Annexes

Annexe - 1 Explication sur les services de raisonnement

Les propositions suivantes donnent les explications sur les services de raisonnement utiles pour la maintenance de la consistance du modèle de base de connaissances

- **Proposition 1** (La réduction des services de raisonnement dans la subsumption) pour les concepts C et D que nous avons :
 1. C est non satisfiable $\Leftrightarrow C$ est subsumé par $\perp$.
 2. C et D sont équivalents $\Leftrightarrow C$ est subsumé par D et D est subsumé par C .
 3. C et D sont disjoints $\Leftrightarrow C \sqcap D$ est subsumé par $\perp$.

Ces déclarations sont également valables pour TBox.

- **Proposition 2** (La réduction des services de raisonnement dans la non satisfiabilité) pour les concepts C et D que nous avons :
 1. C est subsumé par $D \Leftrightarrow C \sqcap \neg D$ est non satisfiable.
 2. C et D sont équivalentes $\Leftrightarrow$ tous les deux $(C \sqcap \neg D)$ et $(C \neg \sqcap D)$ sont non satisfiables.
 3. C et D sont disjoints $\Leftrightarrow C \sqcap D$ est non satisfiable.

Ces déclarations sont également valables pour TBox.

- **Proposition 3** (La réduction des services de raisonnement dans la non satisfiabilité) pour le concept C nous avons :
 1. C est non satisfiable.
 2. C est subsumé par $\perp$.
 3. C et $\perp$ sont équivalents.
 4. C et $\top$ sont disjoints.

Ces déclarations sont également valables pour TBox.

- **Proposition 4** (La réduction de la subsumption) pour les concepts C et D nous avons :
 1. C est subsumé par D ($C \sqsubseteq D$) $\Leftrightarrow C \setminus D = \emptyset$.

Annexe - 2 Technique d'optimisation des algorithmes tableau par expansion des arbres.

Un ensemble de règles d'extension pour la Logique Descriptive $\mathcal{ALC}$ (Langage Attributif pour les Concepts) est donné dans le Tableau 1 où C et D sont des concepts et R est un rôle.

Une des techniques théoriques d'optimisation pour l'expansion de l'arbre est la technique de trace qui est largement utilisée dans la mise en œuvre pratique pour optimiser la quantité d'espace utilisé pour stocker l'expansion. Cette technique impose des limitations sur l'application des règles d'expansion de sorte que l'expansion sans conflit des conjonctions et des disjonctions en utilisant la règle $\sqcap$ et la règle $\sqcup$ soient terminées avant que de nouveaux nœuds soient créés en utilisant la règle $\exists$. Ces algorithmes qui utilisent la technique de trace, incorporent les règles $\forall$ et $\exists$ et utilisent une seule règle $\forall\exists$. Toute nouvelle expansion de l'arbre en utilisant les règles $\forall$ et $\exists$ peut être traitée comme un sous problème indépendant qui renvoie la satisfiabilité ou non. Cette technique de trace se base sur le fait que les nœuds ne sont pas affectés par leurs successeurs (Tableau 1).

Tableau 1 : Règles d'extension pour algorithme tableau de $\mathcal{AL}$ (Baader F. et al., 2010)

$\sqcap$ -règle	si 1. $(C \sqcap D) \in \mathcal{L}(x)$ 2. $\{C, D\} \not\subseteq \mathcal{L}(x)$ alors $\mathcal{L}(x) \rightarrow \mathcal{L}(x) \cup \{C, D\}$
$\sqcup$ -règle	si 1. $(C \sqcup D) \in \mathcal{L}(x)$ 2. $\{C, D\} \cap \mathcal{L}(x) = \emptyset$ alors soit $\mathcal{L}(x) \rightarrow \mathcal{L}(x) \cup \{C\}$ soit $\mathcal{L}(x) \rightarrow \mathcal{L}(x) \cup \{D\}$
$\exists$ -règle	si 1. $\exists R.C \in \mathcal{L}(x)$ 2. il n'existe pas y de telle sorte que. $\mathcal{L}(\langle x, y \rangle) = R$ et $C \in \mathcal{L}(y)$ alors créer un nouveau nœud y et un bord $\langle x, y \rangle$ avec $\mathcal{L}(y) = \{C\}$ et $\mathcal{L}(\langle x, y \rangle) = R$
$\forall$ -règle	si 1. $\forall R.C \in \mathcal{L}(x)$ 2. il existe une certaine y de telle sorte que. $\mathcal{L}(\langle x, y \rangle) = R$ et $C \notin \mathcal{L}(y)$ alors $\mathcal{L}(y) \rightarrow \mathcal{L}(y) \cup \{C\}$
$\exists\forall$ -règle	si 1. $\exists R.C \in \mathcal{L}(x)$ 2. il n'existe pas y de telle sorte que. $\mathcal{L}(\langle x, y \rangle) = R$ et $C \in \mathcal{L}(y)$ 3. ni $\sqcap$ -règle ni the $\sqcup$ -règle est appliquée à $\mathcal{L}(x)$ alors créer un nouveau nœud y et un bord $\langle x, y \rangle$ avec $\mathcal{L}(y) = \{C\} \cup \{D \mid \forall R.D \in \mathcal{L}(x)\}$ et $\mathcal{L}(\langle x, y \rangle) = R$

Une autre technique qui aide à l'expansion de l'arbre est le « lazy unfolding » qui signifie qu'un concept n'est déployé que s'il correspond à un nœud. Si la TBox ($\mathcal{T}$) contient un axiome $A \equiv C$ et qu'une $\sqcap$ -règle est appliquée à un concept $(A \sqcap D) \in \mathcal{L}(x)$ tels que A et D

sont ajoutées à $\mathcal{L}(x)$, alors A peut être déployé en le remplaçant par C . Les règles correspondantes pour l'expansion de l'arbre par le « lazy unfolding » sont dans le Tableau 2.

Tableau 2 : Règles d'expansion pour le Lazy Unfolding de l'algorithme tableau (Baader F. et al., 2010)

U ₁ -règle	si 1. $A \in \mathcal{L}(x)$ et $(A \equiv C) \in \mathcal{T}$ 2. $C \notin \mathcal{L}(x)$ alors $\mathcal{L}(x) \rightarrow \mathcal{L}(x) \cup \{C\}$
U ₂ -règle	si 1. $\neg A \in \mathcal{L}(x)$ and $(A \equiv C) \in \mathcal{T}$ 2. $\neg C \notin \mathcal{L}(x)$ alors $\mathcal{L}(x) \rightarrow \mathcal{L}(x) \cup \{\neg C\}$
U ₃ -règle	si 1. $A \in \mathcal{L}(x)$ and $(A \sqsubseteq C) \in \mathcal{T}$ 2. $C \notin \mathcal{L}(x)$ alors $\mathcal{L}(x) \rightarrow \mathcal{L}(x) \cup \{C\}$

Annexe - 3 Gestion de l'évolution des ontologies

A3.1 Correspondance d'ontologies

Le terme de correspondance d'ontologie fait référence à la tâche qui consiste à relier le vocabulaire de deux ontologies qui partagent le même domaine. Cette opération doit être telle que la structure mathématique de la signature des ontologies et leurs interprétation soit respectées, tel que spécifié par leurs axiomes ontologiques (Kalfoglou Y. and Schorlemmer M., 2003). Une correspondance basique d'ontologies consiste généralement en une collection de fonctions qui assignent les symboles utilisés dans l'un des vocabulaires aux symboles de l'autre. La correspondance d'ontologie peut aussi fournir un niveau commun duquel plusieurs ontologies peuvent être accédées et par conséquent pourraient échanger des informations de manière sémantique.

Les scénarios de correspondance d'ontologie peuvent être considérés soit :

- comme une correspondance d'une ontologie globale vers une ontologie locale ;
- comme une correspondance d'une ontologie locale vers une ontologie locale correspondante ;
- soit comme l'alignement et la fusion d'ontologies locales.

Dans OIS (Ontology Integration Systems), la correspondance est exprimée comme des requêtes sur une ontologie intégrée globale et une ontologie locale. Les deux approches pour la correspondance sont : premièrement en faisant correspondre les concepts globaux à des requêtes sur l'ontologie locale (approche centrée globale), deuxièmement en faisant correspondre les concepts de l'ontologie locale à des requêtes sur l'ontologie globale (approche centrée locale). D'une manière générale, il est possible de classer les méthodes de correspondance selon les capacités des traducteurs utilisés dans les phases antérieures de la

correspondance pour une comparaison taxinomique et selon les limites algorithmiques (Kalfoglou Y. and Schorlemmer M., 2003).

A3.2 Alignement, traduction et articulation des ontologies

Comme la relation binaire entre le vocabulaire de deux ontologies peut elle-même être décomposée en couples de fonctions à partir d'une source intermédiaire commune, nous pouvons donc décrire l'alignement des deux ontologies O_1 et O_2 , au moyen d'une paire de correspondances à partir d'une ontologie intermédiaire source O_0 . Cette ontologie intermédiaire avec ses correspondances vers les deux ontologies ou articulation de deux ontologies, est présentée Figure 42.

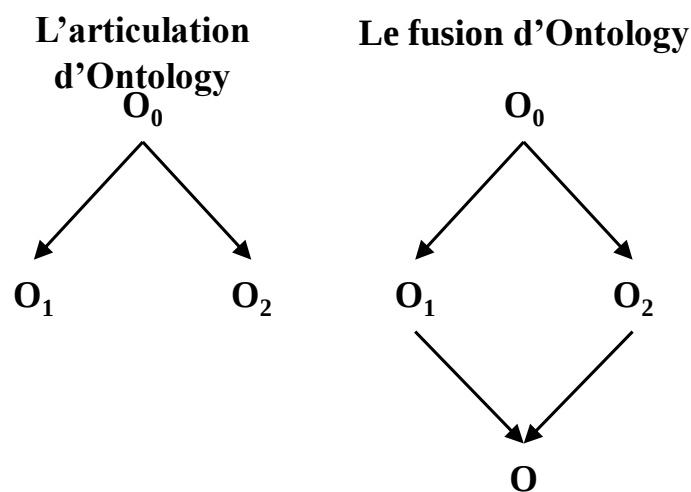


Figure 42 : Représentation de l'articulation d'ontologies et de la fusion d'ontologies (Flouris G. et al., 2008)

La différence entre correspondance et traduction est que la correspondance exige le processus de définition d'une collection de fonctions qui spécifient quels concepts et quelles relations correspondent aux concepts et relations de l'autre ontologie, alors que la traduction est une application directe de fonctions pour traduire une ontologie dans une autre. Le processus de traduction d'ontologie présuppose que les ontologies partagent le domaine dans lequel leur vocabulaire respectif est interprété. Un autre contexte pour la traduction d'ontologie est de la considérer en termes de traduction syntaxique comme, par exemple, traduire des ontologies OWL en RDF.

A3.3 Modification/Édition d'ontologies

Les modifications dans la conceptualisation d'un domaine sont inévitables, comme nous l'avons vu précédemment. Dans ce paragraphe, nous allons nous concentrer sur le problème de l'évolution de l'ontologie qui, suivant la classification présentée dans (Flouris G. et al., 2008), peut être considéré comme de l'édition d'ontologies. Dans la littérature concernant l'utilisation des services d'inférence et le raisonnement d'ontologie, par exemple, identifier la subsomption de concepts, est appliquée sur une ontologie centrale ; les services de raisonnement distribués ne sont pas encore assez matures pour assurer l'exhaustivité de raisonnement ainsi que le maintien de la complexité d'un calcul fiable. Avec ce paradigme de

l'ontologie centrale, l'effort de recherche pour la modélisation efficace et la gestion des modifications dans l'ontologie devient important.

A3.4 Évolution des ontologies

(Plessers P. *et al.*, 2007) soutient que la diffusion de toutes les informations de modification à tous les participants impliqués dans une création collaborative d'ontologie peut créer une confusion, donc les mécanismes qui fourniraient une assistance à la compréhension de l'évolution d'une ontologie peuvent se révéler utiles. Par conséquent (Klein M. *et al.*, 2004) fournit les mécanismes qui assurent la compréhension de plus haut niveau élémentaire et des opérations de changements composites¹⁶. Deuxièmement, les différents utilisateurs peuvent avoir des vues différentes (en termes de granularité) et différents besoins. Il serait donc souhaitable que les utilisateurs puissent définir explicitement les changements qui les intéressent ainsi que le niveau d'abstraction approprié pour eux. Finalement, ce mécanisme permet l'implication dans les opérations de modification, c'est-à-dire l'identification des conséquences liées à une opération de modification (Plessers P. and De Troyer O., 2005).

L'approche basique pour gérer les évolutions d'ontologies est de garder un historique des versions des modifications dans les concepts. Les modifications des concepts sont sauvegardées à des intervalles de temps discrets et linéairement ordonnés $T \subseteq \mathbb{N}$, où $\mathbb{N}$ est l'ensemble des nombre naturels. À chaque intervalle de temps T , la version v du concept (pour un concept $c \in C$ donné) est le quadruplet (σ, A, t_s, t_e) . Ici, $t_s, t_e \in T$ sont les temps de début et fin de l'intervalle de temps correspondant. $\sigma \in \mathbb{N}$ est le nom du concept et A est l'axiome de concept associé tel que $C = (c, A)$. La relation de précédence entre les versions v des concepts est une relation antisymétrique transitive notée $<_v$, telle que $v_1 <_v v_2$ si $t_{e1} < t_{s2}$. L'évolution du concept (E_c) est ainsi définie telle que $\forall v \in E_c$. L'historique des versions V est défini comme un ensemble d'évolution de concepts telles que $\forall c \in C (E_c \in V) \wedge \forall E_c \in V$ (Plessers P. *et al.*, 2007).

A3.5 Débogage d'ontologies

Le débogage d'ontologies est un autre champ de recherche étroitement lié aux évolutions d'ontologies. La différence entre les deux tient dans le fait que dans l'évolution d'ontologies les modifications dans la caractérisation des domaines sont prises en charge et que des tentatives constantes de correction des contradictions logiques sont faites. C'est par exemple le cas lors de l'introduction de nouveaux concepts.

Les raisonnements standards sont très utiles pour les besoins de débogage ontologie, parce que même si ils peuvent identifier l'existence d'une contradiction, ils fournissent peu de soutien pour le résoudre et l'éliminer (Meyer T. *et al.*, 2005). Le type de contradiction logique qui existe dans les ontologies selon (Flouris G. *et al.*, 2006) est une incohérence dans l'ontologie. Une ontologie est incohérente si elle contient un concept insatisfaisant.

¹⁶ L'opération de modification est considérée comme composite, lorsque cette opération de modification englobe plusieurs opérations élémentaires du changement.

A3.6 Versionnage des ontologies

Le versionnage doit fournir aux éléments (données, services, applications et autres ontologies) un accès transparent à chaque version en fonction de leurs besoins. Les algorithmes de versionnage doivent prendre des décisions sur les mises à jours syntaxiques ou sémantiques dans l'ontologie ce qui n'est pas une tâche triviale. De fait, comme d'autres systèmes de la gestion de l'évolution des ontologies cela requiert une intervention humaine dans le déroulement du processus (Noy N.F. *et al.*, 2004).

A3.7 Fusion d'ontologies

Dans la classification des opérations ontologiques, la fusion est la dernière. Dans le cas de l'union, l'ontologie résultat contient l'union des entités des ontologies initiales. Bien sur, dans ce cas se pose le souci de la résolution des différences de représentation d'un même concept. Pour l'intersection, par contre, le résultat ne comporte que les parties communes des composants initiaux. Dans les deux cas, une première approche par recherche de similarités (syntaxique, sémantique, structurelle, d'instance) est souvent un bon angle d'attaque.

On peut noter enfin que les tâches impliquées dans la résolution des hétérogénéités sont toujours les premières tâches du processus d'intégration et/ou de fusion.

Au final, la fusion est souvent considérée comme une intégration particulière dont les différences principales tiennent aux processus de sortie (Little and Rogova, 2009) (Tableau 3).

Tableau 3 : Résumé des sous-zones du changement d'ontologie (Flouris G. *et al.*, 2008)

Résolution d'Hétérogénéité	La cartographie d'ontologie	Objectif: L'interopérabilité des ontologies. Entrée: Deux ontologies (hétérogènes). Sortie: Un mappage entre les vocabulaires d'ontologies. Propriétés: La sortie identifie les entités vocabulaire liées.
	Le morphisme d'ontologie	Objectif: L'interopérabilité des ontologies. Entrée: Deux ontologies (hétérogènes). Sortie: Correspondance entre les vocabulaires et les axiomes des ontologies. Propriétés: La sortie identifie les entités lies du vocabulaire et des axiomes.
	La correspondance d'ontologie (la sortie est appelé l'alignement d'ontologies)	Objectif: L'interopérabilité des ontologies. Entrée: Deux ontologies (hétérogènes). Sortie: Une relation entre des vocabulaires des ontologies. Propriétés: La sortie identifie les entités des vocabulaires liées.

	L'articulation d'ontologie	Objectif: L'interopérabilité des ontologies. Entrée: Deux ontologies (hétérogènes). Sortie: Une ontologie intermédiaire et la correspondance entre des vocabulaires de l'ontologie intermédiaire et chacune des ontologies source. Propriétés: La sortie identifie des entités liées vocabulaire à l'aide d'une ontologie intermédiaire.
	la traduction d'ontologie	Objectif: la traduction au langage d'ontologie différente. Entrée: Une ontologie et un langage d'ontologie cible. Sortie: Une ontologie exprimée dans le langage d'ontologie cible. Propriétés: Doit produit une ontologie équivalente.
	la traduction d'ontologie	Objectif: Mise en œuvre de la cartographie du vocabulaire. Entrée: Une ontologie et une cartographie. Sortie: Une ontologie Propriétés: Mettre en œuvre un changement de vocabulaire pour d'ontologies source tel que spécifié par la cartographie d'entrée.
Modification d'ontologie	L'évolution d'ontologie	Objectif: réagir à un changement dans le domaine ou sa conceptualisation. Entrée: Une ontologie et un ensemble d'opérations de changement. Sortie: Une ontologie Propriétés: Met en place un ensemble de changements à l'ontologie source.
	le débogage d'ontologie	Objectif: Restaure une cohérence ontologique. Entrée: Une ontologie incohérente. Sortie: Une ontologie cohérente. Propriétés: Rendre l'ontologie cohérente.
Gestion d'ontologies	Le versionnage d'ontologie	Objectif: l'accès transparent aux différentes versions d'une ontologie. Entrée: Différentes versions d'une ontologie. Sortie: Un système de versionnage. Propriétés: utilise des identifiants de version pour identifier les versions; fournit un accès transparent à la version correcte; détermine la compatibilité.
Fusion	L'intégration d'ontologie	But: Fusionner la connaissance des

d'ontologie		ontologies couvrant les domaines similaires. Entrée: Deux ontologies (couvrant le domaine similaire). Sortie: Une ontologie. Propriétés: Fusionner les connaissances nécessaires pour couvrir un domaine plus large.
	La fusion d'ontologie	Objectif: Fusionner les connaissances des ontologies couvrant un domaine identique. Entrée: Deux ontologies (couvrant le domaine identiques). Sortie: Une ontologie. Propriétés: Fusionner la connaissance pour décrire le domaine avec plus de précision.

Annexe - 4 Un Langage Ontologique pour Web sémantique basé sur la Logique Descriptive

Le web sémantique a été envisagé comme une évolution du web existant à partir d'une base de documents liés entre eux dans une plateforme d'application ce qui permet de délivrer l'information avec un sens bien mieux défini et ainsi améliorer les interactions homme/machine. Ceci fut réalisé en augmentant les contenus existant avec des annotations sémantiques. La Logique Descriptive propose une logique formelle basée sur la sémantique et fut principalement développée avec des procédures de décision compatibles avec des services de raisonnement.

L'autre priorité majeure dans le développement d'OWL fut de maintenir la compatibilité avec les langages web existant et en particulier avec RDF (Ressource Description Framework). RDF inclus déjà les fonctions de class et de propriété, comme OWL. Le niveau de verbosité de la représentation de la syntaxe RDF / XML n'est pas en tant que telle une difficulté majeure, mais comme RDF est un formalisme basée sur les graphes, avec des graphes exprimés comme un ensemble de triplets sujet-prédicat-objet. Ceci a pour résultat une vraie difficulté d'analyse syntaxique par rapport aux constructeurs d'OWL.

Pour répondre aux préoccupations ci-dessus qui vont du niveau d'expressivité au degré de compatibilité avec RDF, OWL définit trois sous-langages : OWL-Lite, OWL-DL et OWL-Full. OWL-lite est un sous ensemble syntaxique d'OWL-DL avec des restrictions sur certains constructeurs d'expression d'OWL-DL. À noter qu'OWL-FULL ne correspond pas syntaxiquement ou sémantiquement à la Logique Descriptive et maintient la compatibilité avec RDF.

La syntaxe et la sémantique de la Logique Descriptive sur laquelle OWL est basée est présentée dans le Tableau 4. Il s'agit de la Logique Descriptive utilisée dans *ALLC*, vu précédemment, qui inclus donc les rôles primitifs transitifs, les axiomes d'inclusion de rôles

($\mathcal{H}$), la valeur nominale ($\mathcal{O}$), les rôles inverses ($\mathcal{I}$) et la restriction de nombre qualifié possible ($\mathcal{Q}$ si qualifié, $\mathcal{N}$ sinon).

Tableau 4 : Syntaxe et Sémantique de la famille $\mathcal{S}$ de Logique Descriptive

Nom du Constructeur	Syntaxe	Sémantique	Représentation Symbolique
concept atomique	A	$A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$	$\mathcal{S}$
rôle atomique	R	$R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$	
rôle transitive	$R \in R_+$	$R^{\mathcal{I}} = (R^{\mathcal{I}})^+$	
conjonction	$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$	
disjonction	$C \sqcup D$	$C^{\mathcal{I}} \cup D^{\mathcal{I}}$	
négation	$\neg C$	$\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$	
restriction existentielle	$\exists R.C$	$\{x \mid \exists y. (x, y) \in R^{\mathcal{I}} \text{ et } y \in C^{\mathcal{I}}\}$	
restriction de la valeur	$\forall R.C$	$\{x \mid \forall y. (x, y) \in R^{\mathcal{I}} \text{ implique } y \in C^{\mathcal{I}}\}$	
hiérarchie des rôles	$R \sqsubseteq S$	$R^{\mathcal{I}} \subseteq S^{\mathcal{I}}$	$\mathcal{H}$
nominal	$\{o\}$	$\{o^{\mathcal{I}}\}$	$\mathcal{O}$
rôle inverse	R^-	$\{(x, y) \mid (y, x) \in R^{\mathcal{I}}\}$	$\mathcal{I}$
restrictions nombre	$\geq n P$ $\leq n P$	$\{x \mid \# \{y. (x, y) \in P^{\mathcal{I}}\} \geq n\}$ $\{x \mid \# \{y. (x, y) \in P^{\mathcal{I}}\} \leq n\}$	$\mathcal{N}$
restriction de numéro qualifiée	$\geq n P.C$ $\leq n P.C$	$\{x \mid \# \{y. (x, y) \in P^{\mathcal{I}} \text{ and } y \in C^{\mathcal{I}}\} \geq n\}$ $\{x \mid \# \{y. (x, y) \in P^{\mathcal{I}} \text{ and } y \in C^{\mathcal{I}}\} \leq n\}$	
			$\mathcal{Q}$

Dans le Tableau 4 (Horrocks I. *et al.*, 2003) :

- A est un nom de concept ;
- C et D sont des concepts ;
- R et S sont des rôles ;
- R^+ est l'ensemble des rôles transitifs ;
- o est un nom d'individu ;
- P est un rôle simple (c'est-à-dire non transitif et ayant des sous-rôles non transitifs) ;
- n est un entier non-négatif.

Ces logiques sont étendues avec une forme simple de domaine concret : les types de données, désignés par apposition d'un «(D)» aux noms des logiques, par exemple, $\mathcal{S} \mathcal{H} \mathcal{O} \mathcal{I} \mathcal{N} (D)$. Pour éviter l'indécidabilité, DL sépare l'interprétation des types de données de celle des classes et des individus. Pour cela un domaine d'interprétation supplémentaire est ajouté pour les valeurs de données : $\Delta^{\mathcal{I}}D$, séparé du domaine des individus $\Delta^{\mathcal{I}}$. De plus, les propriétés sont séparées en deux ensembles disjoints de propriétés : abstraites et type de données. Les propriétés abstraites sont interprétées comme des relations binaires entre $\Delta^{\mathcal{I}}$ et $\Delta^{\mathcal{I}}D$ (sous ensemble de $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}D$).

Pour résumer, les caractéristiques d'OWL sont :

1. Les ressources sont indiquées par les URI qui sont des noms ou des adresses. OWL construit les références URI de la même manière que RDF. Il est de plus courant que OWL utilise des noms qualifiés en guise de raccourci pour les références URI.
2. OWL recueille l'information dans les ontologies qui sont généralement stockées sous la forme de document web en RDF/XML. Les ontologies peuvent être importées dans d'autres ontologies, accumulant ainsi la connaissance.
3. OWL permet d'utiliser les propriétés d'annotation de RDF pour attacher une information aux classes, propriété ou ontologies. Ces annotations sont des triplets RDF, mais dans OWL-DL elles sont conservées séparément en raison des problèmes expliqués ci-dessus.
4. OWL utilise les possibilités des types de données RDF ainsi que des schémas de données XML pour fournir les types et valeurs de données.
5. OWL-DL et OWL-Lite ont un cadre syntaxique similaire alors que RDF/XML est la syntaxe officielle d'échange pour tous les sous-langages OWL.

Dans cette section, nous avons présenté la théorie de la Logique Descriptive ainsi que ses relations avec le langage ontologique web (OWL). Dans la suite nous nous intéresserons aux services de raisonnement correspondant ainsi qu'à la classification des différentes opérations de gestion du changement des ontologies.

Annexe - 5 Moteur de règles

Nous avons vu quelles sont les capacités représentatives d'OWL et identifié les différents objets et propriétés qu'OWL est capable de représenter. Les relations hiérarchiques entre les propriétés et les classes dans l'artéfact ontologique fournissent une intégration informationnelle et des possibilités d'intégration de processus dans un artéfact central. L'artéfact ontologique peut représenter à un niveau syntaxique les capacités d'OWL, mais pour une aide à la décision utile, un moteur de règle tel que JESS (Friedmann E., 2003) est appliqué pour inférer l'information à partir de l'artéfact ontologique. Les Composant de base du moteur de règles sont un moteur d'inférence, une base de règles et une mémoire de travail (Figure 43). Composant que nous allons rapidement présenter ci-dessous :

A5.1 Mémoire de Travail

Le moteur de règles opère sur une mémoire de travail, aussi appelée la « base de faits », qui contient toutes les informations nécessaires au fonctionnement du moteur de règles. Cette mémoire de travail peut contenir tant les prémisses que les conclusions des différentes règles. Typiquement, le moteur de règles maintient un ou plusieurs index similaires à ceux utilisés dans les bases de données relationnelles, afin que la recherche dans cette mémoire de travail soit une opération très rapide.

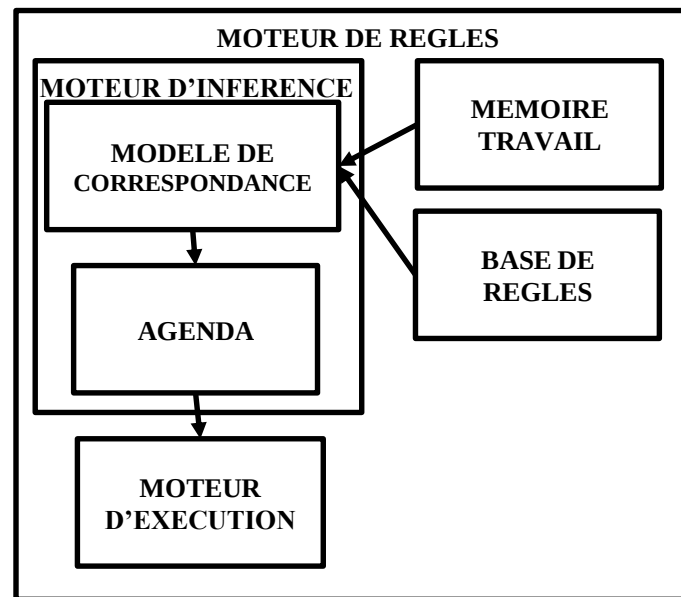


Figure 43: Composants du moteur de règles

À cette fin, la mémoire de travail de JESS est contenue dans une structure de données complexe avec de multiples index. Cette structure est donc similaire à une base de données relationnelle et les faits individuels sont comparables à des lignes dans une table de base de données. Cela implique que les faits, comme les lignes des tables, doivent avoir une structure spécifique pour être indexés. JESS apporte trois façons différentes pour représenter les faits, chacune ayant sa propre structure et des manières différentes d'indexation.

La structure des faits non ordonnés doit être définie *a priori* avant tout travail sur eux. Les faits ordonnés, quant à eux, constitue une liste courte à une dimension et sont simple et pratiques à utiliser pour représenter une information homogène. Ils ne disposent pas de la notion d'identité. Enfin, les faits cachés sont des faits non ordonnés qui sont liés à des objets java dans le monde réel. Ils fournissent une manière de raisonner sur les événements qui sont à l'extérieur de JESS.

A5.2 Base de règles

La base de règles contient toutes les règles que le système connaît. Les règles peuvent simplement être stockées sous la forme de chaînes de texte, mais la plupart du temps elles sont traitées dans une forme qui permet au moteur d'inférence de travailler avec elles plus efficacement. Certains compilateurs de règles réarrangent les prémisses d'une règle afin de la rendre plus efficace ou pour clarifier sa signification pour une exécution automatique.

A5.3 Moteur d'inférence

La première fonction du moteur d'inférence est d'appliquer les règles sur la mémoire de travail. Cela positionne le moteur d'inférence à une place centrale du moteur de règles car c'est lui qui contrôle le processus complet d'inférence des règles afin d'obtenir les sorties du système. Généralement, le moteur d'inférence fonctionne suivant des cycles discrets :

1. Toutes les règles sont confrontées à la mémoire de travail (en utilisant le modèle de correspondance de stratégie : l'algorithme RETE) pour décider laquelle doit être activée sur un cycle. Cette liste non ordonnée de règles activées ainsi que toutes celles activées durant le cycle précédent est nommé ensemble de conflits.
2. L'ensemble de conflits est ordonné pour former un agenda. Le processus pour ordonner l'agenda est appelé : résolution de conflits. La stratégie de résolution de conflits diffère d'un moteur de règles à l'autre.
3. Pour terminer le cycle, la première règle dans le moteur de règle est appliquée (ce qui peut changer la mémoire de travail) et le processus complet est répété.

A5.3.1 Modèle de correspondance – Algorithme RETE

Le principal problème résolu par JESS est celui de la correspondance des règles de la base de règles avec les faits de la mémoire de travail. JESS doit donc réaliser les étapes suivantes en une boucle infinie :

1. Trouver toutes les règles qui sont satisfaites par un ensemble de faits ;
2. Former les enregistrements d'activation à partir de ces associations règles/faits ;
3. Choisir l'enregistrement d'activation à exécuter.

Un modèle de correspondance garde une liste des règles et en vérifie simplement chacune à son tour la partie gauche (prédicats) par rapport à la mémoire de travail pour former un ensemble d'enregistrement d'activation. Après avoir choisi une règle et l'avoir exécutée, il l'efface de l'ensemble des enregistrements d'activation et recommence le processus. Cela est appelé l'approche « rules finding facts ». Une base de règle a généralement un ensemble de règles plus ou moins fixe, alors que la mémoire de travail change continuellement. Cependant, c'est un fait empirique que dans la plupart des systèmes basé sur des règles, une bonne partie de la mémoire de travail est aussi fixée au cours du temps, bien que de nouveaux faits arrivent et que d'anciens disparaissent durant l'exécution du système. De fait, si l'algorithme « rules finding facts » peut enregistrer des résultats de modèles de correspondance antérieurs entre les cycles, ajuster la correspondance seulement pour les faits qui viennent de changer sera plus efficace (Friedman E., 2003).

L'algorithme RETE (Figure 44) est la technique utilisée par JESS pour réaliser une modèle de correspondance rapide. Seuls les faits nouveaux ou effacés sont testés face aux règles à chaque étape. L'algorithme RETE crée un réseau de nœuds interconnectés où chaque nœud représente la partie gauche (prédicats) des règles. Les nœuds entrés sont en haut alors que les nœuds sortis sont en bas. Les faits ajoutés ou enlevés sont traités dans ce réseau de nœud. Au bas du réseau sont les nœuds qui représentent les règles individuelles. L'ensemble de faits traverse le réseau et les faits qui passent tous les tests jusqu'au bas du réseau deviennent de nouveaux résultats d'activation ou un ordre d'annulation d'un enregistrement d'activation précédent (Friedman E., 2003).

Une représentation de l'algorithme de RETE est montrée Figure 44. Les nœud du haut sont les têtes des règles et agissent comme des contraintes pour les évaluations postérieures dans l'algorithme.

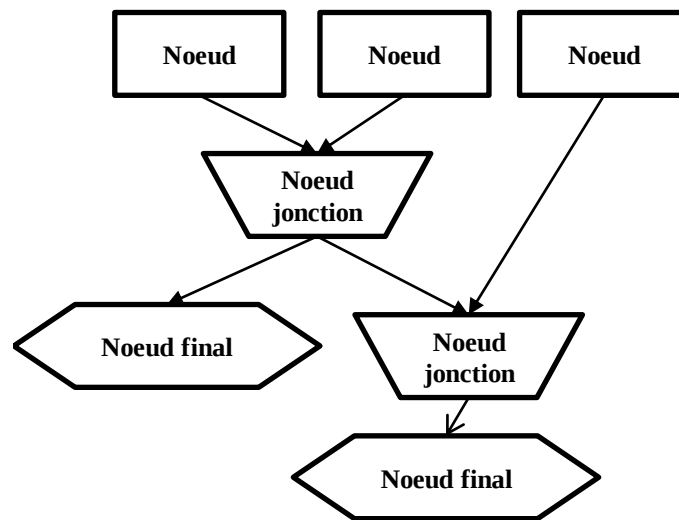


Figure 44: Algorithme Rete

Les faits qui passent le test sont envoyés vers les nœuds « sorties », les autres sont ignorés. Les nœuds jonctions amalgament les résultats de correspondance des premiers $n-1$ modèles (provenant de haut gauche dans la figure) avec le $n^{\text{ième}}$ (en haut droit). Les nœuds jonction enregistrent tous les faits ou groupe de faits qui arrivent sur l'une de leurs deux entrées. Le réseau est construit de telle manière que l'entrée de gauche peut recevoir des groupes d'un ou plusieurs faits alors celle de droite ne reçoit qu'un seul et unique fait. Chaque nœud jonction produit un groupe de deux ou plus faits en sortie. Les nœuds finaux, au bas du réseau, représentent les règles individuelles. Ils ont une seule entrée et pas de sortie. Lorsqu'ils reçoivent une entrée, ils construisent un enregistrement d'activation à partir de l'objet d'entrée et de la règle qu'ils représentent et le placent dans l'agenda (Friedman E., 2003).

Lorsque le moteur d'inférence décide qu'elle règle doivent être tirées, il doit encore décider laquelle appliquer en premier. La liste des règles qui peuvent potentiellement être appliquées est stockée dans l'agenda. Celui-ci utilise la stratégie de résolution de conflits pour décider quelle règle appliquer puis celle qui ont la priorité la plus grande. Après que le moteur de règle ait décidé quelle règle utiliser, il a à exécuter la partie action de la règle (droite). Le moteur d'exécution est le composant du moteur de règles qui applique les règles. JESS offre un langage de programmation complet qui peut être utilisé pour définir ce qu'il advient quand une règle donnée est appliquée.